Estimation of angular velocity and acceleration from shaft encoder measurements

Pierre R Belanger

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1436

Programme 4
Robotique, Image et Vision

ESTIMATION OF ANGULAR VELOCITY AND ACCELERATION FROM SHAFT ENCODER MEASUREMENTS

Pierre R. BÉLANGER

Mai 1991

RR-1436

ESTIMATION OF ANGULAR VELOCITY AND ACCELERATION FROM SHAFT ENCODER MEASUREMENTS

Pierre R. Bélanger

TR-CIM-91-1

Computer Vision and Robotics Laboratory
McGill Research Centre for Intelligent Machines
McGill University
Montréal, Québec, Canada

Postal Address: 3480 University Street, Montréal, Québec, Canada H3A 2A7
Telephone: (514) 398 6319 Telex: 05 268510 FAX: (514) 398 7348
Network Address: pierre@engl.lan.mcgill.ca

This report is being published simultaneously by the McGill Research Center for Intelligent Machines, McGill University, Montreal, Canada, and the Institut National de Recherche en Informatique et Automatique, Sophia-Antipolis, France.

Ce rapport est publié simultanément par le McGill Research Center for Intelligent Machines, Montréal, Canada, et l'Institut National de Recherche en Informatique et Automatique, Sophia-Antipolis, France.

Contents

Abstract

The output of a shaft encoder is basically a quantized measurement of the shaft angle. One method of angular velocity and acceleration measurement is by finite difference of angle measurements (net pulse counts) at regular time samples. It is shown that these estimates become significantly degraded as sampling rates increase

Kalman filtering of the angle measurements, based on some model of signal generation, is proposed. Asymptotic analysis at constant sampling rate as the intersample time tends to zero shows that the error variances tend to zero, but at decreasing rates for the angle, velocity and acceleration errors. The analysis also suggests the use of multiple-integrator systems to model the angle signal.

A triple-integrator model is used if the acceleration is to be estimated, or double-integrator model if not. It is possible to include an externally-generated model of the third derivative or acceleration signal. Example show that the angle estimates are only slightly better than the raw measurements, that the velocity estimates improve by a factor of 2 to 4, with an order of magnitude improvement for the acceleration. The improvements are greater at shorter sampling periods.

A second classical method of estimation of angular velocity and acceleration uses the pulse arrival times. It is shown that this method also becomes degraded at high sampling rates, especially the acceleration estimates.

A Kalman filtering solution is possible here as well, but for asynchronous data. The asymptotic analysis shows that the time constants of the Riccati error covariance equation are short, so that quasi steady state conditions hold.

Because of the computational burden imposed by the real-time propagation of the Riccati equation, a suboptimal solution is proposed. This is essentially a weighted least-squares solution within each sampling interval, taking into account the error covariance update at

the end of the interval. A low-speed modification is introduced, to include the information that an encoder pulse was not produced during a given time interval. The proposed algorithm may incorporate an external signal model, or may use such an estimate to adjust the variance of the plant driving noise in the model. The effect of time measurement errors is studied and examples are given. The estimation errors are generally smaller than in the case of the Kalman filter at constant sampling rate, but the error transients that stem from rapid step-like acceleration changes are attenuated only by a factor of about two.

Keywords: Robotics, control, angular velocity, angular acceleration, estimation

Résumé

La sortie d'un codeur est une mesure quantifiée de l'angle. Une méthode bien connue d'estimation de la vitesse et de l'accélération angulaires est celle des différences finies des mesures d'angle (comptes nets des impulsions) à intervales réguliers. On montre que ces estimés se détériorent rapidement avec l'augmentation du taux d'échantillonage.

On propose le filtrage de Kalman des mesures d'angle, moyennant certains modèles linéaires stochastiques du signal. Etant donné un taux d'échantillonage constant, une analyse asymptotique selon laquelle la durée entre les échantillons tend vers zéro montre que les variances des erreurs d'estimation tendent vers zéro. L'analyse appuie aussi l'usage de modèles à intégrations multiples.

On se sert d'un modèle avec trois intégrations pour estimer l'accélération, et d'un modèle avec deux intégrations si seulement la vitesse doit être estimée. On peut aussi ajouter comme entrée un signal externe modélisant la troisième dérivée on l'accélération. Quelques exemples démontrent que les estimés de l'angle ne représentent qu'une faible amélioration sur la sortie brute du codeur, que les estimés de la vitesse sont de 2 à 4 fois meilleurs que ceux de la méthode des différences finies et que l'amélioration des estimés de l'accélération est de l'ordre de 10. L'amélioration est plus prononcée si le taux d'échantillonage est élevé.

Une seconde méthode classique d'estimer la vitesse et l'accélération angulaires est basée sur les temps d'arrivée des impulsions du codeur. On montre que cette méthode donne aussi des résultats qui se détériorent avec l'augmentation du taux d'échantillonage surtout pource qui est de l'accélération.

Une filtre de Kalman est possible ici aussi, mais pour donnée asynchrones. L'analyse montre que les constantes de temps de l'équation de Riccati sont courtes, de sorte qu'il est permis de supposer un régime quasi permanent.

Le fardeau numérique amené par l'équation de Riccati nous amené à proposer une

méthode sous-optimale. Il s'agit essentiellement d'une méthode des moindres carrés pondérés appliquée à chaque intervale d'échantillonage, tenant compte de la covariance de l'erreur au début de l'intervale et mettant à jour la covariance avant le début de l'intervale suivant. On présente une modification pour les situations où la vitesse est basse, se servant de l'information contenue dans le fait que le codeur n'a pas produit d'impulsion pendant un certain intervale de temps. Le nouvel algorithme peut incorporer un modèle externe d'un signal, on encore peut, suivant ce signal, ajuster la variance du bruit d'entrée du modèle. On étudie aussi l'effet des erreurs de mesures temporelles, et on donne quelques exemples. Les erreurs d'estimation sont généralement plus faibles que celles du filtre de Kalman à taux constant, mais les transitoires d'erreur dues à des changements rapides de l'accélération ne sont atténuées que d'un facteur de deux.

Mots clés: Robotique, commande, vitesse angulaire, accélération angulaire, estimation

ACKNOWLEDGEMENTS

More than half of the work reported here was done at INRIA, Sophia-Antopolis, France, during a sabbatical leave, from January to June 1990.

I wish to thank INRIA for their hospitality. I wish also to recognize the support of the PRISME project team, in particular that of Claude Samson, whose collaboration was most useful to me.

REMERCIEMENTS

Plus de la moitié du travail rapporté in a été fait à l'INRIA, Sophia-Antipolis, lors d'un congé sabbatique de février à juin 1990. Je tiens à remercier l'INRIA de m'avoir acceuilli.

Je reconnais ici l'appui de l'équipe du project PRISME, en particulier œlui de Claude Samson, dont la collaboration m'a été précieuse.

Chapter 1 INTRODUCTION

1.1 INTRODUCTION

An optical shaft encoder is basically a disk with two sets of regularly-spaced slots se along concentric circles. The passage of a slot in front of a light beam produces a pulse the net number of pulses in a given direction, multiplied by the (constant) angle betwee slots, gives the angular displacement in that direction during the counting period. The tw sets are in quadrature, so that it is possible to deduce direction of motion by knowing whic pulse train leads the other.

Low-cost encoders (such as one finds on a PUMA) have a basic resolution of about 20 pulses/rev. Encoders of 5000 to 10000 pulses/rev are now available at costs of the orde of $1000. It is possible to use so-called *multiplicative decoding*, by triggering on the leadin and trailing edge of each pulse, thereby gaining a factor of 4. Of course, in Robotics, it i also common to place the encoder on the motor shaft to gain a factor of 50 to 100 on the measurement of the joint angle.

In commercial robots, as well as in many research units, joint velocity feedback is used either for PD control or for more complex state feedback. Tachometers are not always used in robots; the angular velocity is often derived from the joint angle measurements obtained by an optical shaft encoder. Joint acceleration is sometimes desired, for purposes

of identification [1] or control [2,3].

It is possible to use accelerometers to measure joint acceleration, but such instruments would measure the total rotary motion of a joint: the acceleration of the shaft could be isolated only by a kinematic calculation. As is the case with velocities, acceleration is also inferred from angular position.

There are some results on angular velocity estimation, fewer on angular acceleration. The velocity results fall in two categories: finite-difference and inverse-time. Finite-difference schemes count the net number of pulses during a fixed interval of time, multiply by the angle corresponding to successive pulses and divide by the duration of the interval. References [4, 5, 6] represent several physical implementations of the finite-difference scheme, with very few variations in the basic principle. The inverse-time method deduces velocity as the interpulse angle divided by the time between successive pulses [7, 8]. Here again, the focus in the literature is on the physical implementation rather than on the method.

In [9], the velocity estimation problem is posed in terms of Kalman filtering. A linear, noise-driven, time-invariant system is assumed to represent the angular motion to be measured. An exact position output is given at those time instants when a pulse is generated, but with a random error as to the time of occurrence. The random time error can be transformed to a position error (through multiplication by the velocity), and the problem then appears as a state estimation problem for a LTI system with non-uniform observation times and state (i.e. velocity)-dependent measurement noise covariance. The examples given in the paper lead one to believe that the results are applicable with relatively large interpulse intervals, so as to allow time to compute the complete time-varying Kalman filter equations.

A different scheme using a phase-lock loop [10] came to light recently. No comparisons were made with that particular methods, but it would appear to be suitable to speed control applications, where the velocity is reasonably high and varies relatively slowly.

To describe the nature of the problem, consider an encoder system with an interpulse

angle θ_m. Let T_s be the sampling interval of the control system, and let N_p be the number of pulses measured during the interval T_s. The angular velocity estimate by finite difference is $N_p\theta_m/T_s$; this relation shows that the estimate is quantitized, with a quantitization step θ_m/T_s. To illustrate, let $\theta_m = 0.003$ deg (encoder resolution of 300 pulses/rev., $\times 4$ decoding, 100 : 1 gear ratio) and $T_s = 0.01s$. Then $\theta_m/T_s = 0.3$ deg/s; that resolution may be appropriate at high angular velocity, but represents only 10% resolution at 3 deg/s, a figure that is not uncommonly low for a robot approaching steady state. Of course, at $1kHz$ sampling frequency, the resolution becomes a rather unacceptable 3 deg/s.

The acceleration estimates are worse. The application of the usual finite difference formula shows that the quantum is now θ_m/T^2, or 30 deg/s^2 in the above example.

The finite-difference schemes are limited by the encoder resolution; in the case of inverse-time methods, it is the measurement errors that are the dominant effect. Due to manufacturing errors and other such causes, successive encoder pulses do not correspond exactly to angular differences of θ_m. Errors are thought to be as much as 25% of θ_m. Assuming exact measurement of the time between successive pulses, the velocity estimate would be off by the same percentage, the acceleration by as much as twice that percentage.

Finite-difference methods operate at constant sampling rates; inverse-time schemes work asynchronously, according to the occurrence of pulses. This report will consider both constant-rate and asynchronous methods. The constant-rate algorithm turns out to be a Kalman filter for a system with uniform sampling rate. That is due to the fact that, under suitable assumptions, the knowledge that the output lies between two known levels is approximately equivalent to observing the output corrupted with Gaussian white noise. Since sampling periods are normally small compared to system time constants, some effort is devoted to the asymptotic behavior of the Kalman filter, as the sampling period tends to zero. It will be found that the Kalman gain for an nth-order, all-pole system tends to that of the nth- order in integration: that justifies the rather common usage of an all-integrator model,

which is followed here. A triple integrator models the process when acceleration estimates are wanted, a double integrator is used if only velocity estimates are required. Simulation runs are made, with an underdamped second-order system as the test system; even though the theory is stochastic in nature, the response of a robot is better modeled by a deterministic system. To conclude, the use of an a priori model to improve the estimation is considered. It is found that the degree of improvement, if any, depends on the modeling accuracy of the third derivative, for acceleration estimation, or of the acceleration, for velocity estimates.

The generalization of the inverse-time approach is, like [9], a Kalman-based method. In order to simplify the computation, the plant noise is ignored within a sampling period T_s, so that the Kalman filter becomes the solution of a (simpler) weighted least-squares problem. This allows all pulse arrival times within a system sampling period to be processed "in batch", a useful outcome since estimates are only desired at the sampling instants of the control system. The effect of plant noise is taken into account through a modification of the error covariance at the end of each sampling period. A novel aspect of the algorithm is that information is also obtained, at low velocities, from the fact that a pulse has *not* occurred, i.e. that the output has not crossed a new level (triggered a pulse) for over a certain period of time. This greatly improves the performance at low velocities. Simulations are also run, as in the case of the constant-rate method.

Chapter 2 ESTIMATION AT CONSTANT SAMPLING RATE

2.1 The Finite Difference Method

The finite-difference method is reviewed in this section, for an encoder whose operation is illustrated in Fig. 1. The dotted lines represent the nominal quantization levels of the encoder; they are equally spaced, with a a spacing θ_m. The solid lines show the true levels, distinct from the nominal levels by errors $\epsilon_{n-1}, \epsilon_n, \epsilon_{n+1}$. Pulses are emitted whenever the angular output $\theta(t)$ crosses the true levels, at times t_1, t_2, t_3. The ϵ_i are assumed to be independent, zero-mean, identically-distributed random variables, with a distribution such that $\mid \epsilon_i \mid < \theta_m/2$ (so as to ensure that level i cannot go above level $i+1$). The ϵ_i corresponding to level i is constant in time, and is therefore the same at every crossing of that level by the output, an assumption that was respected in the simulation work.

With reference to Fig. 1, if the counter is sampled at time $t_s, t_2 < t_s < t_3$, and if the sign of the velocity is known to be positive, then $\theta(t_s)$ is known to lie between levels n and $n+1$.

If the sign of the velocity is not known, then $\theta(t_s)$ can only be pinned down to be between levels $n-1$ and $n+1$. Assume the latter case, for the moment. Then, Prob. $(\theta(t_s)|$ last level $= n, \epsilon_{n-1}, \epsilon_{n+1})$ = Uniform distribution, $(n-1)\theta_m + \epsilon_{n-1}$ to $(n+1)\theta_m + \epsilon_{n+1}$.

It is clear that

$$\mathrm{E}[\theta(t_s)|\text{ last level } = n, \epsilon_{n-1}, \epsilon_{n+1}] = n\theta_m + \frac{\epsilon_{n-1} + \epsilon_{n+1}}{2}$$

and that the estimate $\widehat{\theta}(t_s)$ is

$$\widehat{\theta}(t_s) = \mathrm{E}[\theta(t_s)|\text{ last level } = n] = n\theta_m. \tag{1}$$

To calculate the variance, write $\widetilde{\theta} = \theta - \widehat{\theta}$ and

$$\begin{aligned}\mathrm{E}[\widetilde{\theta}^2|\text{ last level } &= n, \epsilon_{n-1}, \epsilon_{n+1}] \\ &= \frac{1}{2\theta_m + \epsilon_{n+1} - \epsilon_{n-1}} \int_{(n-1)\theta_m + \epsilon_{n-1}}^{(n+1)\theta_m + \epsilon_{n+1}} (\theta - n\theta_m)^2 \, d\theta \\ &= \frac{1}{3}(\theta_m^2 + \theta_m\, \epsilon_{n+1} - \theta_m\, \epsilon_{n-1} + \epsilon_{n+1}^2 + \epsilon_{n-1}^2 + \epsilon_{n+1}\epsilon_{n-1})\end{aligned}$$

and

$$\text{var } (\widetilde{\theta}) = \mathrm{E}[\widetilde{\theta}^2|\text{ last level } = n] = \frac{\theta_m^2 + 2r}{3} \tag{2}$$

where $r = \mathrm{E}[\epsilon_i^2]$.

If the sign of the velocity is known, then Equation (1) is replaced by

$$\widehat{\theta}(t_s) = (n \pm \frac{1}{2})\theta_m \tag{3}$$

where the + applies for positive velocity, the − for negative velocity. The variance in that case is

$$\text{var } (\widetilde{\theta}) = \frac{\theta_m^2/4 + 2r}{3}. \tag{4}$$

In order to write the finite-difference estimate, let t_{s-1} and t_s represent consecutive sampling times, with $t_s - t_{s-1} = T$, and let $n(t_s), n(t_{s-1})$ represent the encoder readings at t_s and t_{s-1}, respectively. In the absence of knowledge concerning the sign of the velocity, the angular velocity estimate is

$$\widehat{v}(t_s) = \frac{\widehat{\theta}(t_s) - \widehat{\theta}(t_{s-1})}{T}. \tag{5}$$

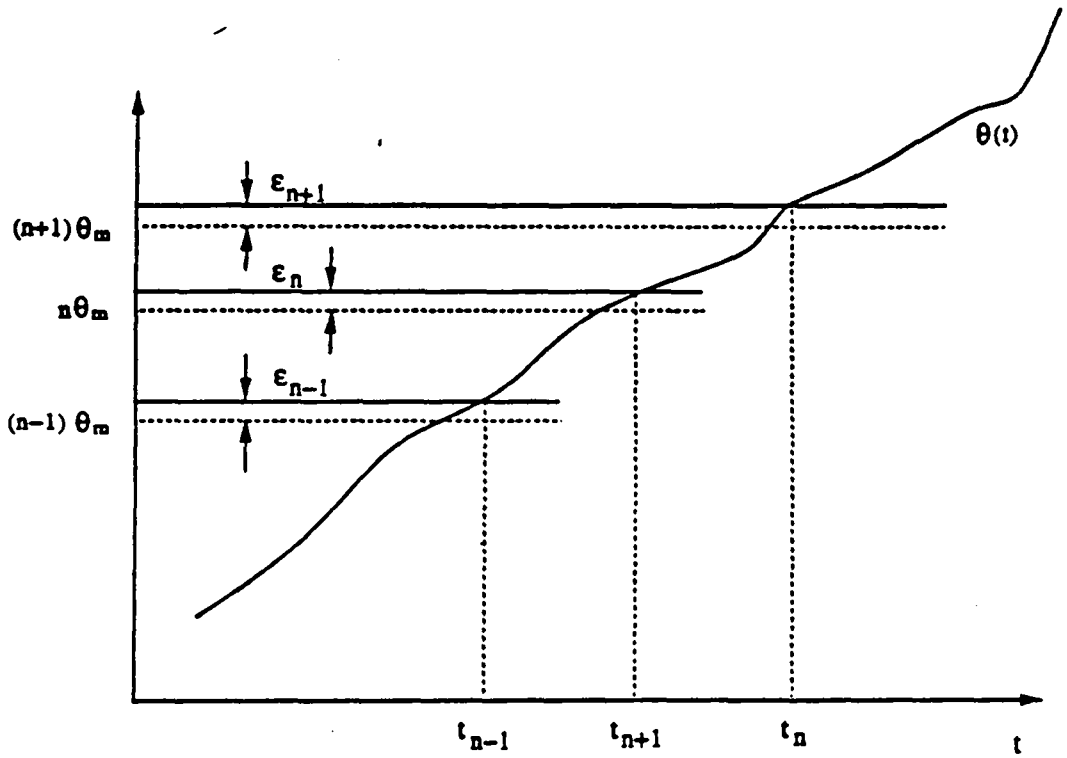


Figure 1 Encoder input-output characteristics.

By Taylor's theorem,

$$\theta(t_{s-1}) = \theta(t_s) - v(t_s)T + \frac{1}{2}a(t_s - \alpha T)T^2, \quad 0 \leq \alpha \leq 1 \tag{6}$$

where a is the angular acceleration.

Then,

$$v(t_s) = \frac{\theta(t_s) - \theta(t_{s-1})}{T} + \frac{1}{2}a(t_s - \alpha T)T \tag{7}$$

and

$$\tilde{v}(t_s) = v(t_s) - \hat{v}(t_s) = \frac{\tilde{\theta}(t_s) - \tilde{\theta}(t_{s-1})}{T} + \frac{1}{2}a(t_s - \alpha T)T. \tag{8}$$

The estimation error $\tilde{v}(t_s)$ has a deterministic component and a stochastic component, with

$$|\tilde{v}_{\text{det}}| \leq \frac{1}{2}|a|_{\max}T \tag{9}$$

where $|a|_{\max} = \max_{t\in[t_s-T,T]} |a(t)|$.

The stochastic component has a variance

$$\text{var}\,[\tilde{v}_{\text{stoc}}] = \frac{2}{T^2}\ \text{var}\,[\tilde{\theta}]. \tag{10}$$

Equation (10) assumes that $\tilde{\theta}(t_s)$ and $\tilde{\theta}(t_{s-1})$ are independent, which in turn requires that $n(t_s)$ and $n(t_{s-1})$ differ by at least 2; thus, Equation (10) is not valid at very low velocity. In the case where the sign of the velocity is not known, Equation (2) is used in Equation (10) to yield

$$\text{var}\,[\tilde{v}_{\text{stoc}}] = \frac{2}{3}\frac{\theta_m^2 + 2r}{T^2}. \tag{11}$$

Comparing Equations (9) and (11), the deterministic error is proportional to T, while the standard deviation of the stochastic part goes as $1/T$. This suggests that there is an optimal T for the finite-difference estimate of velocity.

The acceleration estimate is

$$\hat{a}(t_s) = \frac{\hat{\theta}(t_s) - 2\hat{\theta}(t_{s-1}) + \hat{\theta}(t_{s-2})}{T^2}. \tag{12}$$

Straightforward application of Taylor's Theorem yields

$$a(t_s) = \frac{\theta(t_s) - 2\theta(t_{s-1}) + \theta(t_{s-2})}{T^2} - \frac{1}{3}\,\theta'''(t_s - \alpha T)T + \frac{4}{3}\,\theta'''(t_s - 2\beta T)T \tag{13}$$

for some α, β between 0 and 1, and

$$\begin{aligned}\tilde{a}(t_s) &= a(t_s) - \hat{a}(t_s) \\ &= \frac{\tilde{\theta}(t_s) - 2\tilde{\theta}(t_{s-2}) + \tilde{\theta}(t_{s-2})}{T^2} - \frac{1}{3}\,\theta'''(t_s - \alpha^T)T + \frac{4}{3}\,\theta'''(t_s - 2\beta T)T.\end{aligned} \tag{14}$$

Therefore

$$|\tilde{a}_{\text{det}}| \leq \frac{5}{3}|\,\theta'''|_{\max} T \tag{15}$$

with

$$|\,\theta'''|_{\max} = \max_{t\epsilon[t_s, t_s - 2T]} |\,\theta'''(t)|.$$

The stochastic component has a variance

$$\text{var}\,[\tilde{a}_{\text{stoc}}] = \frac{4}{T^4}\,\text{var}\,[\tilde{\theta}] \tag{16}$$

where it is assumed that $n(t_s), n(t_{s-1}), n((t_{s-2})$ are all different by at least 2.

For the case where the sign of the velocity is not known, Equation (2) yields

$$\text{var}\,[\tilde{a}_{\text{stoc}}] = \frac{4}{3}\frac{\theta_m^2 + 2r}{T^4}. \tag{17}$$

Here again, comparing Equations (15) and (17), there is an optimal T for the finite-difference estimator of acceleration.

Example 1

Let ϵ_i follow a triangular distribution with $-\epsilon_m \leq \epsilon \leq \epsilon_m$. It can be shown that, for such a distribution, the variance $r = \epsilon_m^2/6$. Use $\theta_m = 0.003°, \epsilon_m = \theta_m/4, T = 0.01s$.

From Equations (11) and (17),

$$\begin{aligned}\text{Std. dev}\,(\tilde{v}_{\text{stoc}}) &= (\frac{2}{3})^{1/2}\frac{(.003)(1 + 1/48)^{1/2}}{0.01} \\ &= 0.247\ \text{deg}/s \\ \text{Std. dev}\,(\tilde{a}_{\text{stoc}}) &= (\frac{4}{3})^{1/2}\frac{(.003(1 + 1/48)^{1/2}}{.0001} \\ &= 35.0\ \text{deg}/s^2.\end{aligned}$$

Note that the encoder error has little influence on the result; for reasonable tolerances, $r \ll \theta_m^2$.

2.2 Filtering Solution

Obviously, it is necessary to use filtering in order to reduce the variances; that is especially so at high sampling rates. It is useful to postulate a linear, stochastic model to represent $\theta(t)$:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + \Gamma w(t)$$
$$y(t) = \theta(t) = C\mathbf{x}(t) + e(t). \tag{18}$$

In this model, the states x_1, x_2, x_3 are taken to be the angle, angular velocity and angular acceleration, respectively; if the acceleration is not of interest, only x_1 and x_2 need be so defined. Thus,

$$C = [1 \;\; 0 \;\; 0 \cdots 0].$$

In Equation (18), $w(t)$ is the usual white Gaussian noise, zero mean with covariance q. It is recognized that robotic motion is not well characterized by such a stationary random process, and that q may be looked upon simply as a filter parameter to be adjusted.

The scalar output $y(t)$ is the measurement, equal to the latest encoder count times the resolution θ_m. The error $e(t)$ is the quantitization error, assumed white, zero mean and of variance R given by Equations (2) or (4), depending on whether or not the sign of the velocity is known.

The optimal filter is, of course, the Kalman filter. Because the observations are equally spaced in time, the discrete steady-state version is used; this is in contrast to [9], where the Kalman filter is time-varying because of the asynchronous observations. A few remarks are in order:

1. Because $e(t)$ is not Gaussian, the Kalman filter is suboptimal. It is, of course, the best linear filter in the least-squares sense.
2. One could use the raw estimates of velocity and acceleration (Equations (5) and (12)) as filter inputs. There is no advantage in this, if the model is valid: the result will yield exactly the same filter transmission y to $\hat{x}$. There may be something to be gained, however, if the model is poorly defined: some structures can be more robust than the Kalman filter.

2.3 Asymptotic Analysis

Since modern robots operate at high sampling rates, it is useful to examine the performance of the Kalman filter as the sampling time tends to zero. The equations of interest are

$$\begin{aligned} P(t+1/t) &= A_d P(t/t) A_d^T + Q_d \\ P(t/t) &= [I - KC]P(t/t)[I - KC]^T + KRK^T \\ K &= P(t/t-1)C^T[CP(t/t-1)C^T + R]^{-1} \end{aligned} \tag{19}$$

All matrices are constant; A_d and Q_d are the A and Q matrices for the discrete-time model that corresponds to the system of Equation (18) at the given sampling rate. Note that R does *not* vary with the sampling rate: that is a crucial feature of this analysis.

For small sampling period T,

$$\begin{aligned} A_d &= e^{AT} \approx I + AT \\ Q_d &= q\int_0^T e^{A(t-\tau)}\Gamma\Gamma^T e^{A^T(t-\tau)}d\tau \approx q\Gamma\Gamma^T. \end{aligned} \tag{20}$$

Equation (20) is used in Equation (19), which is manipulated to yield

$$P = P + APT + PA^TT - PC^T(CPC^T + R)^{-1}CP + q\Gamma\Gamma^TT$$

$$- APC^T(CPC^T + R)^{-1}CPT - PC^T(CPC^T + R)^{-1}CPA^TT$$
$$+ APA^TT^2 - APC^T(CPC^T + R)^{-1}CPA^TT^2 \qquad (21)$$

where $P = P(t + 1/t)$.

It is assumed (to be justified later) that P is of the order of T to some positive power. If only terms of order up to T^2 are retained, there results

$$AP + PA^T - PC^T(RT)^{-1}CP + q\Gamma\Gamma^T = 0. \qquad (22)$$

This is the continuous-time Kalman filter equation for the system of Equation (18), but with the measurement noise variance equal to RT. It is also the dual of the LQ control problem defined by

$$\dot{\mathbf{x}} = A^T\mathbf{x} + C^T u$$
$$J = \int_0^\infty [q\,\mathbf{x}^T\,\Gamma\,\Gamma^T\,\mathbf{x} + T\,R\,u^2]dt$$

which is seen to be the so-called *cheap control problem.*

Basic results on this problem may be found in [11], and are summarized as follows:

1. $\lim_{T\to 0} P = P_0$
2. $\lim_{T\to 0} \int_0^\infty q\mathbf{x}^T\Gamma\Gamma^T\mathbf{x}dt = \mathbf{x}^T(0)P_0\mathbf{x}(0)$
3. If dim $(\Gamma^T\mathbf{x})$ > dim $(\mathbf{u})$, then $P_0 \neq 0$
4. If dim $(\Gamma^T\mathbf{x})$ = dim $(\mathbf{u})$ and
   $H(s) = \Gamma^T(sI - A^T)C^T$ has all LHP zeros, then $P_0 = 0$.

Here, C^T is a column vector, and u is a scalar. Furthermore, there is no loss of generality with respect to second-order statistics if the model of Equation (18) is chosen to be minimum-phase. Therefore, Condition 4 is satisfied and $P_0 = 0$.

There remains to solve for P_0 as a function of T. Two authors [12, 13] present solutions, under the condition expressed in this case as

$$C\Gamma\Gamma^TC^T > 0$$

corresponding to $B^TQB > 0$ in the control problem The condition is reasonably practical in the control case; it states, roughly, that at least one of the state variables that are driven directly by u appear in the performance index. The condition is much less practical in filtering, since it requires the output to contain at least one variable driven directly by the plant noise. Here, since $C = [1\ 0\ 0\ \cdots\ 0]$, the condition $C\Gamma \neq 0$ requires that the model contain a white noise velocity component, which is not realistic. No other results could be found, as it appears that work on the cheap control problem virtually stopped in the mid seventies.

The problem is solved here for the all-pole case, which is adequate to present purposes. The model is illustrated in Fig. 2, with an all-pole $G(s)$ of order $m > 0$; this structure entails some loss of generality, in that neither a nor v may have white noise components. The fact that $G(s)$ is all-pole does take away some generality, but rather little since a given spectrum can be achieved arbitrarily closely by a high-order $G(s)$.

The state space model, in companion form, is

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots\cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots\cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots\cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots\cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots\ 0 & 1 \\ 0 & 0 & -a_0 & -a_1 & \cdots\cdots & -a_{m-1} \end{bmatrix}$$

$$\Gamma^T = [0\ \ 0\ \ \cdots\ 0\ \ 1]$$

$$C = [1\ \ 0\ \ \cdots\cdots\ \ 0]$$

The solution is worked out for the dual control problem. The following result is proved in Appendix A:

Lemma 1: The solution of the Riccati equation (22) is of the form $P_{ij} = P^*_{ij} T^{\frac{2n-i-j+1}{2n}}$, where P^*_{ij} is a constant and $n = m + 2$ is the order of the system. This leads to

Theorem 1 The solution of the Riccati equation (22) for small T is $P^*_{ij} T^{\frac{2n-i-j+1}{2n}}$, where P^*_{ij} is the solution of the same Riccati equation, but with $T = 1$ and $a_0 = a_1 = a_{m-1} = 0$.

The Kalman gain is that related to this modified Riccati equation, divided by T.

Proof: Write Equation (22) element by element, with

$$P_{ij} = P^*_{ij}\mu^{2n-i-j+1}, \quad \mu = (T)^{1/2n}.$$

For $i, j \neq n$,

$$-\frac{P^*_{ij}P^*_{ij}}{R}\mu^{4n-i-j} + \mu^{2n}(P^*_{j,i+1}\mu^{2n-i-j} + P^*_{i,j+1}\mu^{2n-i-j}) = 0$$

or

$$-\frac{P^*_{1i}P^*_{1j}}{R} + P^*_{j,i+1} + P^*_{i,j+1} = 0 \tag{24}$$

For $j = n, i \neq n$,

$$-\frac{P^*_{1i}P^*_{1n}}{R}\mu^{3n-i} + \mu^{2n}(P^*_{i_1,n}\mu^{n-i} - a_0 P^*_{i3}\mu^{2n-i-2}$$
$$- A_1 P^*_{i4}\mu^{2n-i-3} - \cdots - a_{m-1}P^*_{in}\mu^{n-i+1}) = 0$$

Leaving out all terms except those in μ^{3n-i} yields

$$-P^*_{1i}P^*_{1n} + P^*_{i+1,n} = 0 \tag{25}$$

Finally, for $i = j = n$,

$$-\frac{P^{*2}_{1n}}{R}\mu^{2n} + \mu^{2n}q + 2\mu^{2n}(-a_0 P^*_{3n}\mu^{n-2} - A_1 P^*_{4m}\mu^{n-3}\cdots$$
$$- a_{m-1}P^*_{nn}\mu) = 0$$

Since $n > 0$ (because $n = m + 2, m > 0$) the leading terms are

$$-\frac{P^{*2}_{1n}}{R} + q = 0. \tag{26}$$

It is easy to verify that Equations (24), (25) and (26) are precisely the Riccati equation of Equation (22), for the case $a_0 = a_1 = \cdots = a_{m-1} = 0$ and $T = 1$.

The Kalman gain is

$$K = PC^T/R$$

It is written out as

$$K = \frac{1}{R}\begin{bmatrix} P_{11}^{\bullet} & \mu^{2n-1} \\ P_{12}^{\bullet} & \mu^{2n-2} \\ & \vdots \\ P_{1n}^{\bullet} & \mu^{n} \end{bmatrix}$$

There now remains to justify the assumption made to go from Equation (21) to Equation (22). The discrete Riccati equation is

$$P = (I + AT)[P - PC^T(CPC^T + R)^{-1}CP](I + AT)^T + q\Gamma\Gamma^T T \tag{27}$$

The following result is demonstrated in Appendix A:

Theorem 2: The solution of Equation (22) is also the solution of Equation (27), to the lowest powers of T.

Remarks:

1. The diagonal elements of the covariance matrix are $P_{11} = P_{11}^{\bullet} T^{\frac{2n-1}{2n}}, P_{22} = P_{22}^{\bullet} T^{\frac{2n-3}{2n}}$, $\cdots P_{ii} = P_{ii}^{\bullet} T^{\frac{2n-2i+1}{2n}} \cdots P_{nn} = P_{nn}^{\bullet} T^{\frac{1}{2n}}$; the term in T gets progressively larger as the index i grows. Ignoring the relative magnitudes of the $P_{ii}^{\bullet}$, this means that the covariance becomes larger with larger i. For instance, with $n = 3$ and $T = 0.01, T^{5/6} = 0.0215, T^{3/6} = 0.1, T^{1/6} = 0.464$.

2. The fact that the asymptotic gain and covariance do not depend on the coefficients $a_0, a_1, \cdots a_{m-1}$ is significant, in that it supports the choice of an n-integrator model if the sampling period is small.

3. The effect of varying T can be readily assessed, since it is necessary to solve only one continuous-time Riccati equation and multiply the elements of the solution by powers of μ.

2.4 Models to be used

If angular acceleration is to be estimated, the model

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w$$
$$y = [1 \quad 0 \quad 0]\mathbf{x} + e \tag{28}$$

is postulated.

The corresponding discrete-time system is

$$\mathbf{x}(kT+T) = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}(kT) + \mathbf{w}(kT)$$

$$Q = \mathrm{E}[\mathbf{w}(kT)\mathbf{w}^T(kT)] = q \begin{bmatrix} \frac{1}{20}T^5 & \frac{1}{8}T^4 & \frac{1}{6}T^3 \\ \frac{1}{8}T^4 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{6}T^3 & \frac{1}{2}T^2 & T \end{bmatrix}$$

In Robotics, the behavior is perhaps better described by a deterministic system, which can be written as

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dddot{\theta}(t) \tag{29}$$

The white plant noise of Equation (28) is a surrogate for the third derivative. Clearly, the more wide-band $\dddot{\theta}$ is, the better the correspondence will be between the two models.

Equation (29) suggests that it may be possible to use a model $\dddot{\theta}_{\mathrm{mod}}(t)$ of the third derivative. In that case,

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dddot{\theta}_{\mathrm{mod}} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} (\dddot{\theta} - \dddot{\theta}_{\mathrm{mod}}) \tag{30}$$

The corresponding discrete-time system is

$$\mathbf{x}(kT+T) = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}(kT) + \begin{bmatrix} \frac{1}{6}T^3 \\ \frac{1}{2}T^2 \\ T \end{bmatrix} \dddot{\theta}_m(kT) + \mathbf{w}(kT)$$

with the Q matrix as before.

Here, the random input w models the difference $\dddot{\theta} - \dddot{\theta}_{\text{mod}}$. If there is to be an improvement over the case where a model of $\dddot{\theta}$ is not used, the intensity of the "noise" $\dddot{\theta} - \dddot{\theta}_{\text{mod}}$ must be less than that of $\dddot{\theta}$. Roughly speaking, the error in modeling the third derivative must be smaller than the third derivative itself.

For velocity estimation only, the model is

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x} + e \end{aligned} \tag{31}$$

Corresponding to Equation (29), one writes

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a(t) \tag{32}$$

and, rather than Equation (30),

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a_{\text{mod}} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (a - a_{\text{mod}}). \tag{33}$$

The corresponding discrete-time model is

$$\mathbf{x}(kT + T) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}(kT) + \begin{bmatrix} \frac{1}{2}T^2 \\ T \end{bmatrix} a_{\text{mod}}(kT) + \mathbf{w}(kT)$$

with

$$Q = \mathrm{E}(\mathbf{w}(kT)\mathbf{w}^T(kT)] = q \begin{bmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^2 & T \end{bmatrix}$$

The high-frequency content of $a(t)$ is not as high as that of $\dddot{\theta}$, so that modeling $a(t)$ by white noise has less justification. On the other hand, it is easier to model $a(t)$, so one would expect to do better using $a_{\text{mod}}(t)$.

2.5 Examples

Figure 3 shows a bare-essentials model of a robot joint. The transfer function is

$$\frac{y}{y_d} = \frac{(1/J)(k_p + k_D s)}{s^2 + \frac{k_D}{J} s + \frac{k_p}{J}}. \tag{34}$$

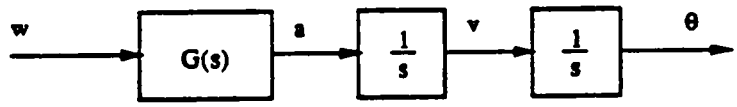


Figure 2 Angular signal model.

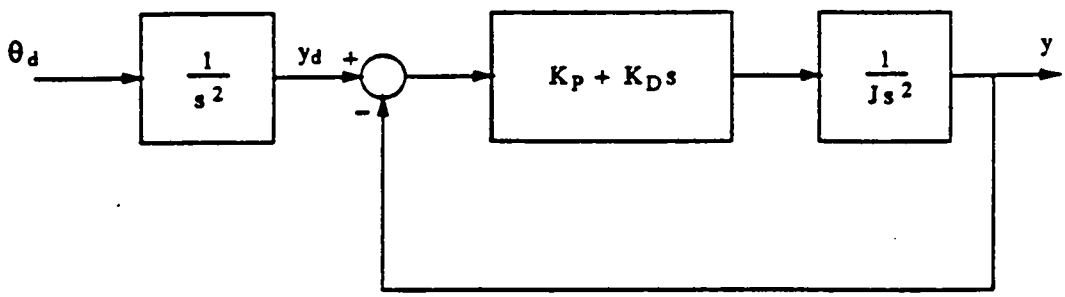


Figure 3 Robot joint closed-loop model.

The nominal case is for $J = 1, k_D = 6, k_p = 100$ so that

$$\frac{y}{y_d} = \frac{6s + 100}{s^2 + 6s + 100}. \tag{35}$$

The pole pair has an ω_0 of 10 rad/s and a damping factor ζ of 0.3, leading to somewhat underdamped responses; that was thought to create a greater challenge for estimation than would have been provided by overdamped signals.

The desired acceleration $a_d(t)$ is shown in Fig. 4a, for an amplitude $A = 1$. The values $A = 1$ and $A = 10$ were used, corresponding to slow and fast trajectories. The responses $y(t), \dot{y}(t)$ and $\ddot{y}(t)$ are shown in Fig. 4b, c and d for $A = 1$.

There was no attempt to include the estimation in the control loop: the intent was only to generate reasonably realistic signals to test the estimator.

In certain runs, a model is assumed to be available, to generate $\dddot{\theta}_{\text{mod}}$ or a_{mod}. That model is taken to be of the same structure as the main one, with $J = 2, k_D = 6, k_p = 100$.

Unless otherwise stated, the shaft encoder resolution is taken to be $\theta_m = 0.003$ deg. and ϵ_i assumed to have a triangular distribution with $\epsilon_m = 0.00075$ deg. The sampling period is $T = 0.01s$.

Example 2

This is to assess the degree to which the asymptotic solution (Theorem 1) approximates the true solution, for small T.

Since the result holds only for the all-pole case, we use

$$\frac{y}{a_d} = \frac{100}{s^2(s^2 + 6s + 100)} \tag{36}$$

and model a_d by white noise, $Q = 1.0$ and $R = \theta_m^2/12$ with $\theta_m = 0.003°$. The controllable canonical form is used.

The Kalman gains, for the fully discretized system, is given in Fig. 5 for a range of values of the sampling period T (dotted curves). The gain as calculated via the asymptotic formulae is also given (full curves).

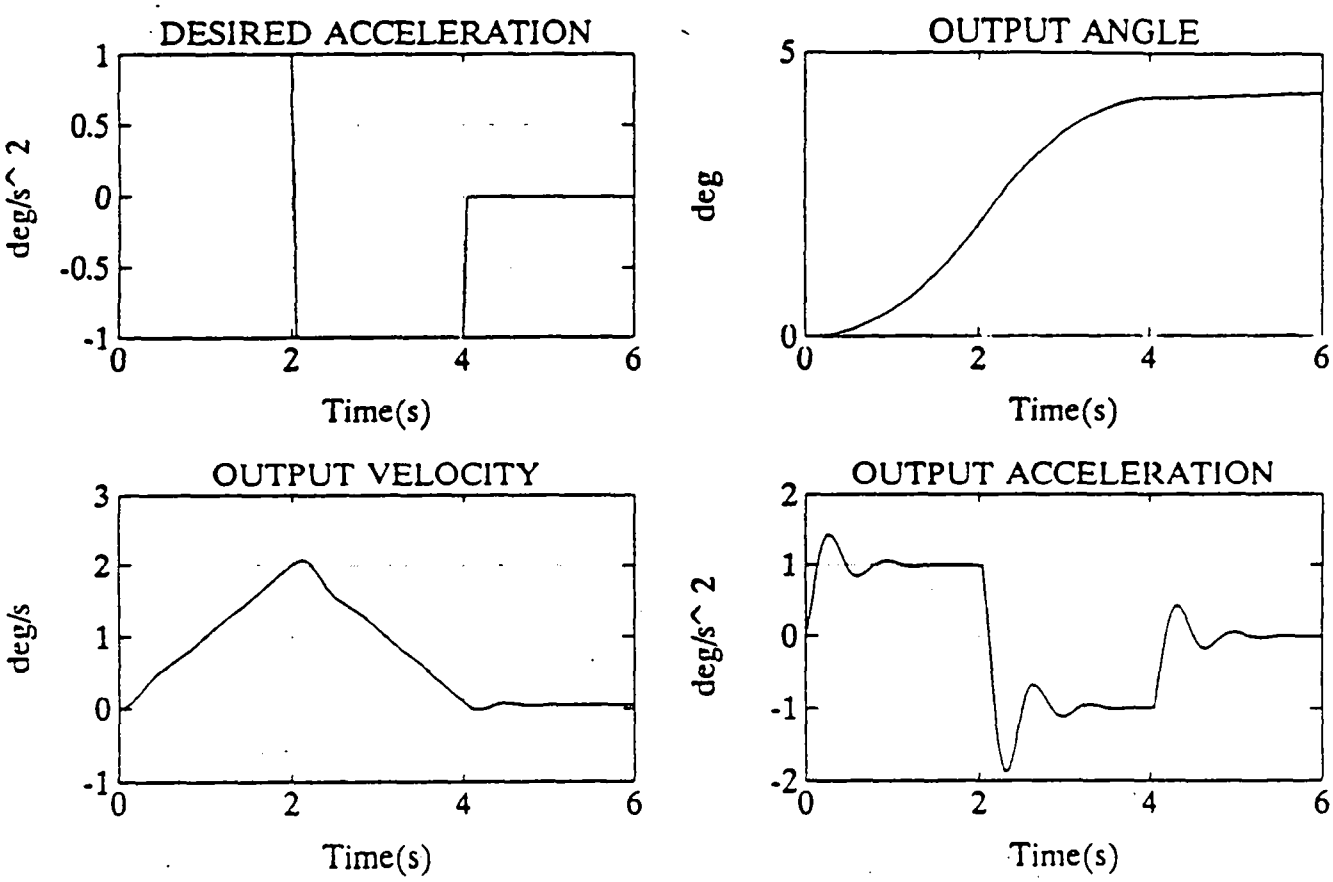


Figure 4 Angle, velocity and acceleration signals, $A = 1$.

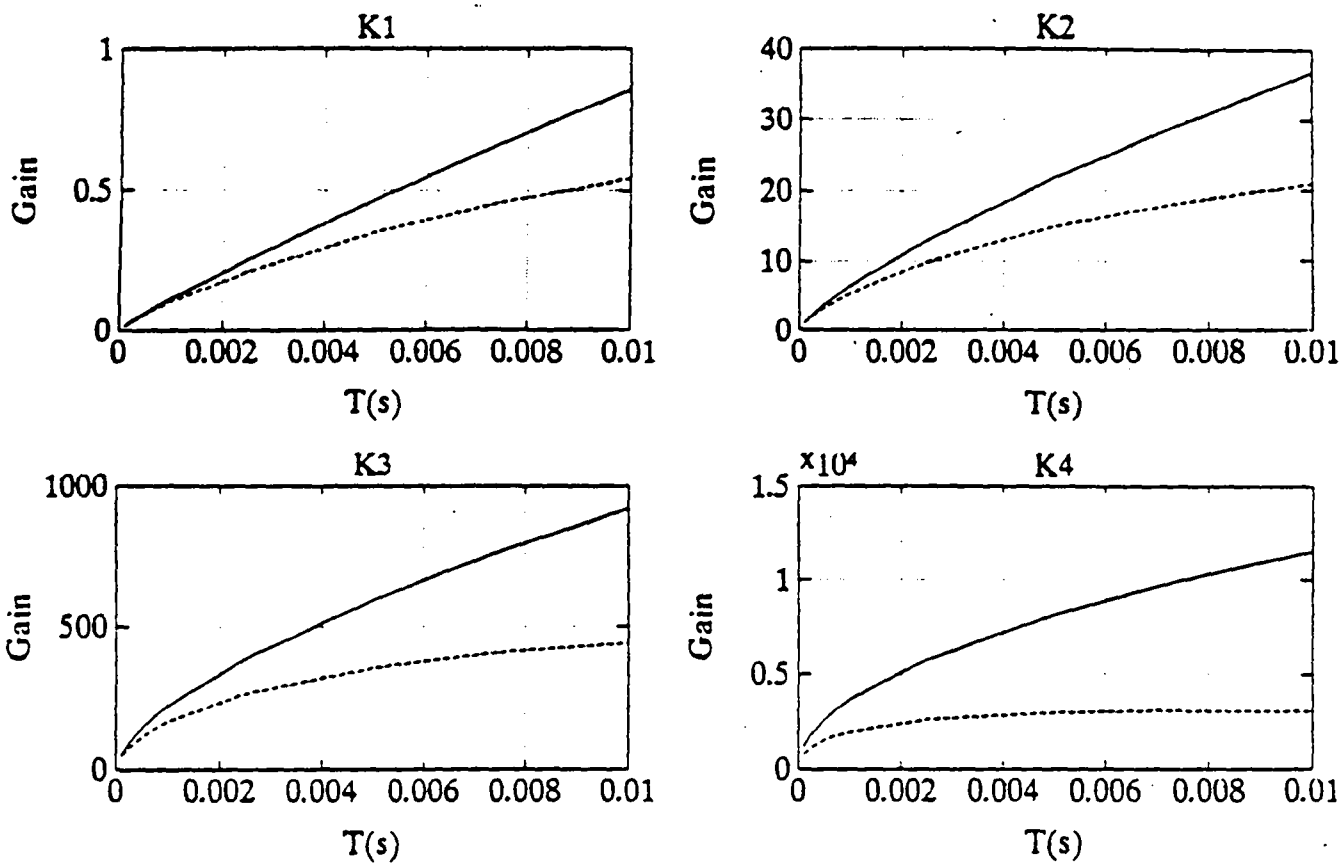


Figure 5 Kalman gains: true values (dotted), asymptotic approximation (full) vs sampling period.

Example 3

This is to explore the effect of replacing the "true" model by a simpler triple or double integrator. The Kalman gain is computed for the system of Equation (34), in the state form

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -100 & -6 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 6 \\ 64 \end{bmatrix} w$$
$$y = [1 \quad 0 \quad 0 \quad 0]\mathbf{x} + v. \tag{37}$$

Let the discretized system be of the form

$$\mathbf{x}(t+1) = A_d\mathbf{x}(t) + \mathbf{w}_d(t)$$
$$y(t) = C\mathbf{x}(t) + v(t)$$

with $\mathrm{E}[\mathbf{w}_d\mathbf{w}_d^T] = Q$.

The Kalman filter is

$$\hat{\mathbf{x}}(t+1|t) = A_d(I - KC)\hat{\mathbf{x}}(t|t-1) + A_dKy(t)$$

where K is the Kalman gain.

Similar equations are written for the simple models of Equations (29) and (31)). Since these models are not of the same order, it is appropriate to make the comparison on the basis of input-output behavior. Specifically, discrete bode plots of the transfer functions $\hat{x}_2/y$ and $\hat{x}_3/y$ (for the triple integrator of Equation (29), only the first for Equation (31)) are calculated and displayed in Fig. 6.

There is a problem in choosing the value of q for the two simple models, since the plant noise drives those systems differently from that of Equation (37). The q's were adjusted until a good fit was obtained (by eye); $q = 1.0$ was used for Equation (37), $q = 40$ for Equation (29) and $q = 0.2$ for Equation (31). For the model of Equation (37), the variance of the third derivative turns out to be 677. The sampling period was $T = 0.01s$ in each case. This

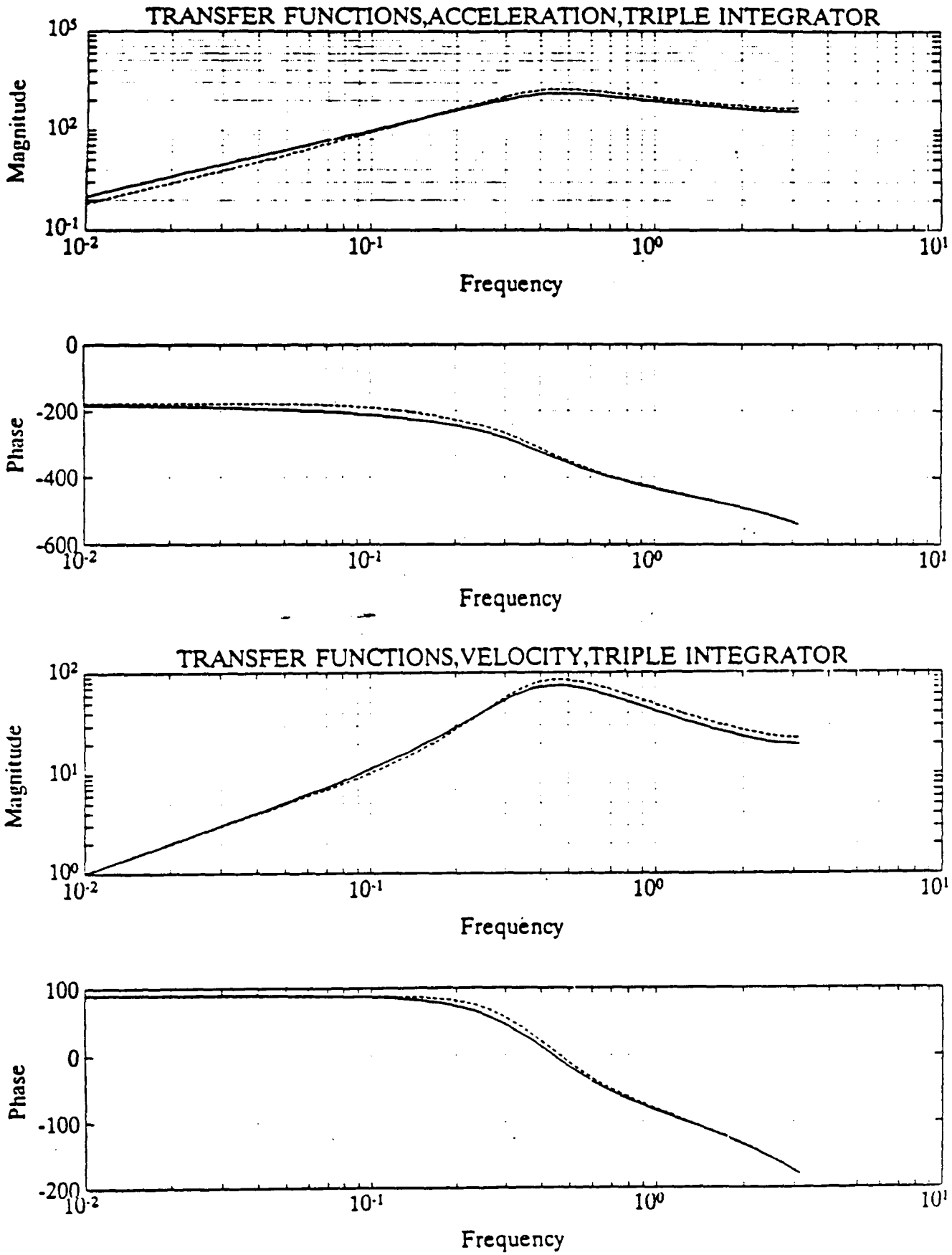


Figure 6 Measurement-to-estimate transfer functions (discrete time): true Kalman filter (dotted), based on triple integrator (full).

shows that it is possible to use the simpler models to generate near-optimal input-output characteristics.

Example 4

This is a simulation, using the system of Fig. 3 with the input of Fig. 4, with $A = 10$ and a sampling period of 0.01s. Figure 7 shows the angle error due to the quantization of the shaft encoder, and the angular velocity error of the finite difference estimate. (The angular acceleration estimate is a multiple of 30 deg/s^2, and is totally useless). Kalman filter estimates using the triple-integrator model, are shown in Fig. 8, for $q = 200$; that value corresponds roughly to the square of the maximum slope of the angular acceleration. Estimates were also computed for $q = 20$, but are not shown. All estimates are a priori estimates, i.e. $\hat{\mathbf{x}}(t+1|t)$.

Table I summarizes the means and standard deviations of several numerical experiments including this one. The results for $q = 200$ and $q = 20$ are similar, which seems to indicate that the choice of q is not crucial. An explanation for this may be gleaned from Fig. 9, where the magnitudes of the transfer function from plant noise input to angular velocity error and angular acceleration error are displayed. It can be verified that the plant noise input, in this case the third derivative, has most of its energy between 0.8 and 30 rad/s. Since $T = .01$, that corresponds to the range of ωT between .008 and .3 in Fig. 9. The magnitude is slightly higher for $q = 20$ than for $q = 200$ in the low frequency range; that is offset to some degree by the fact that the error variance due to observation noise is a bit larger for $q = 200$ than for $q = 20$.

The Kalman filter estimates of the angle are marginally worse than the raw estimates; that is possible because the plant noise is not white, so that the Kalman filter need not be optimal. The Kalman filter estimates of angular velocity are better than the finite difference estimates by a factor of 2 in the standard deviation, the acceleration estimates are improved by an order of magnitude.

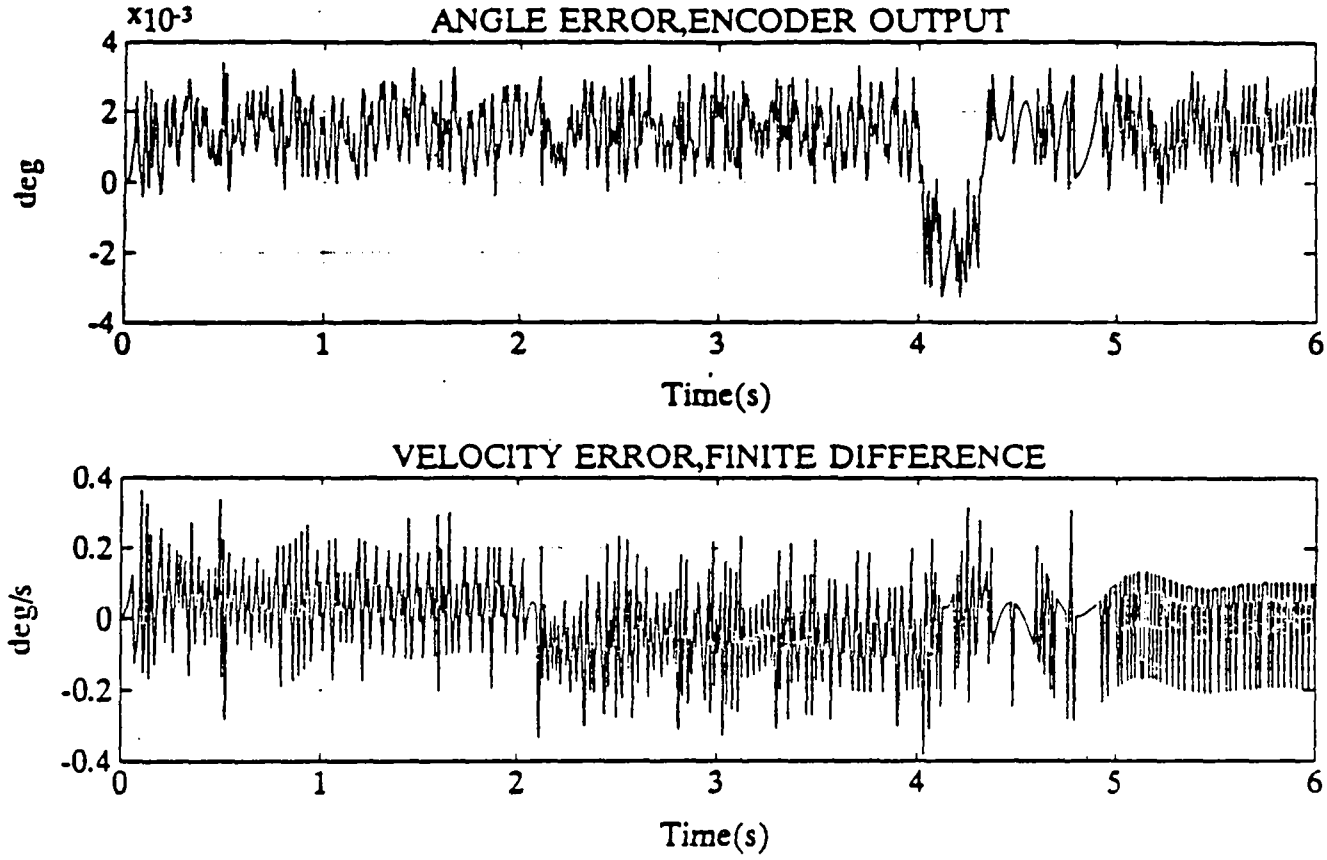


Figure 7 Fast trajectory ($A = 10$), encoder output error and finite-difference velocity error.

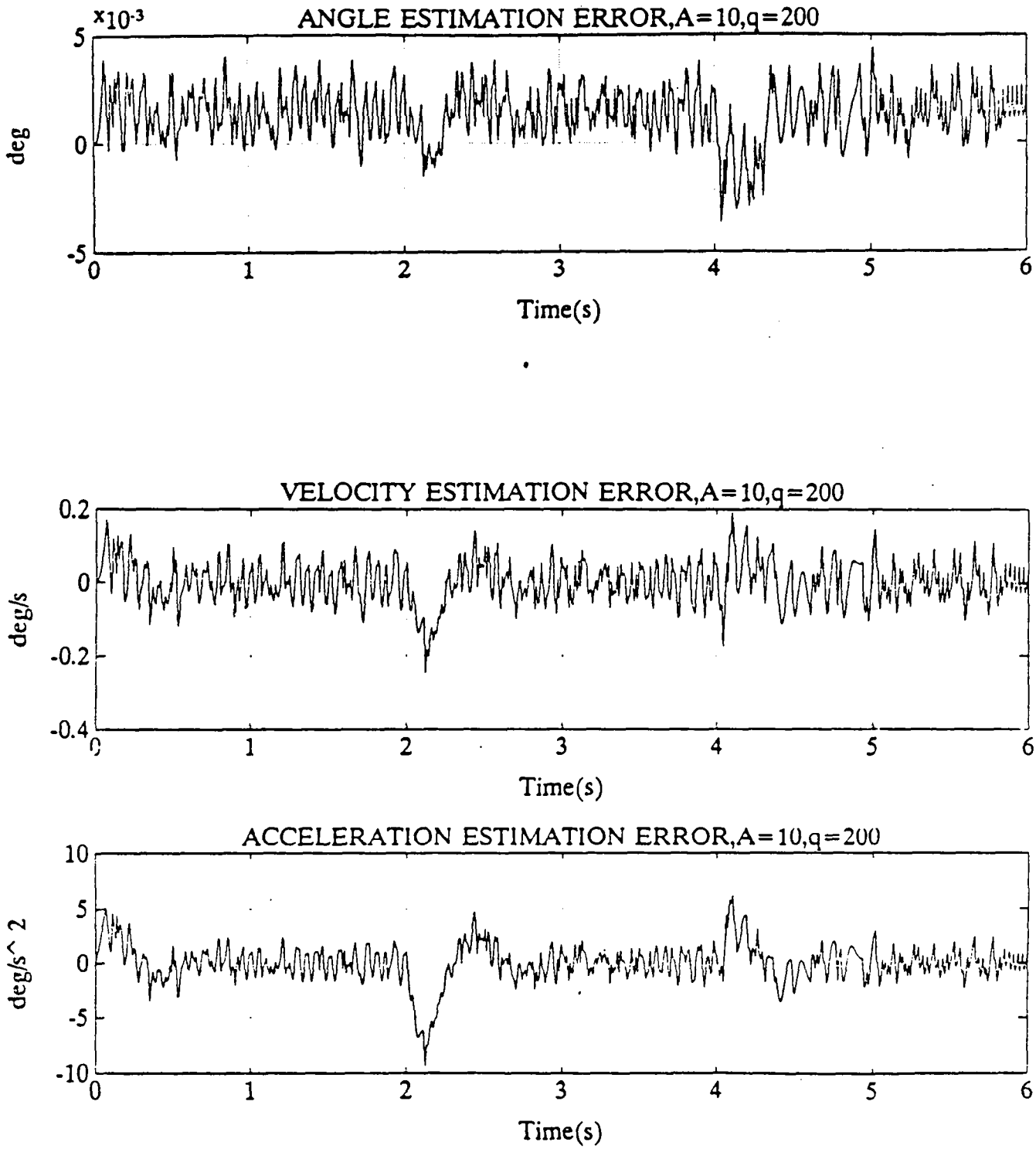


Figure 8a Fast trajectory, Kalman filter error for angle, velocity and accelerations, $q = 200$, triple integrator.

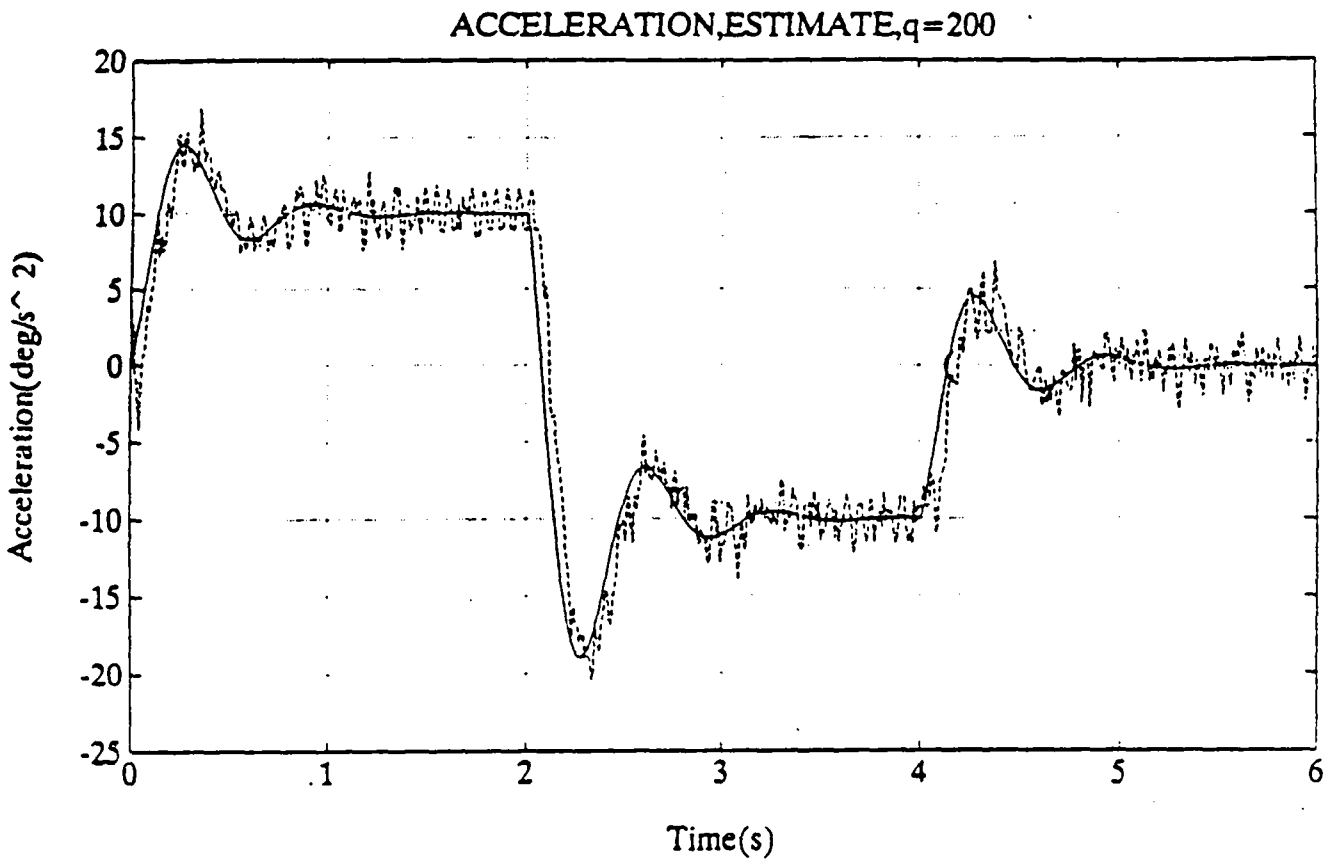


Figure 8b Angular acceleration: true (full curve) and Kalman filter estimate (dotted), $q = 200$, triple integrator.

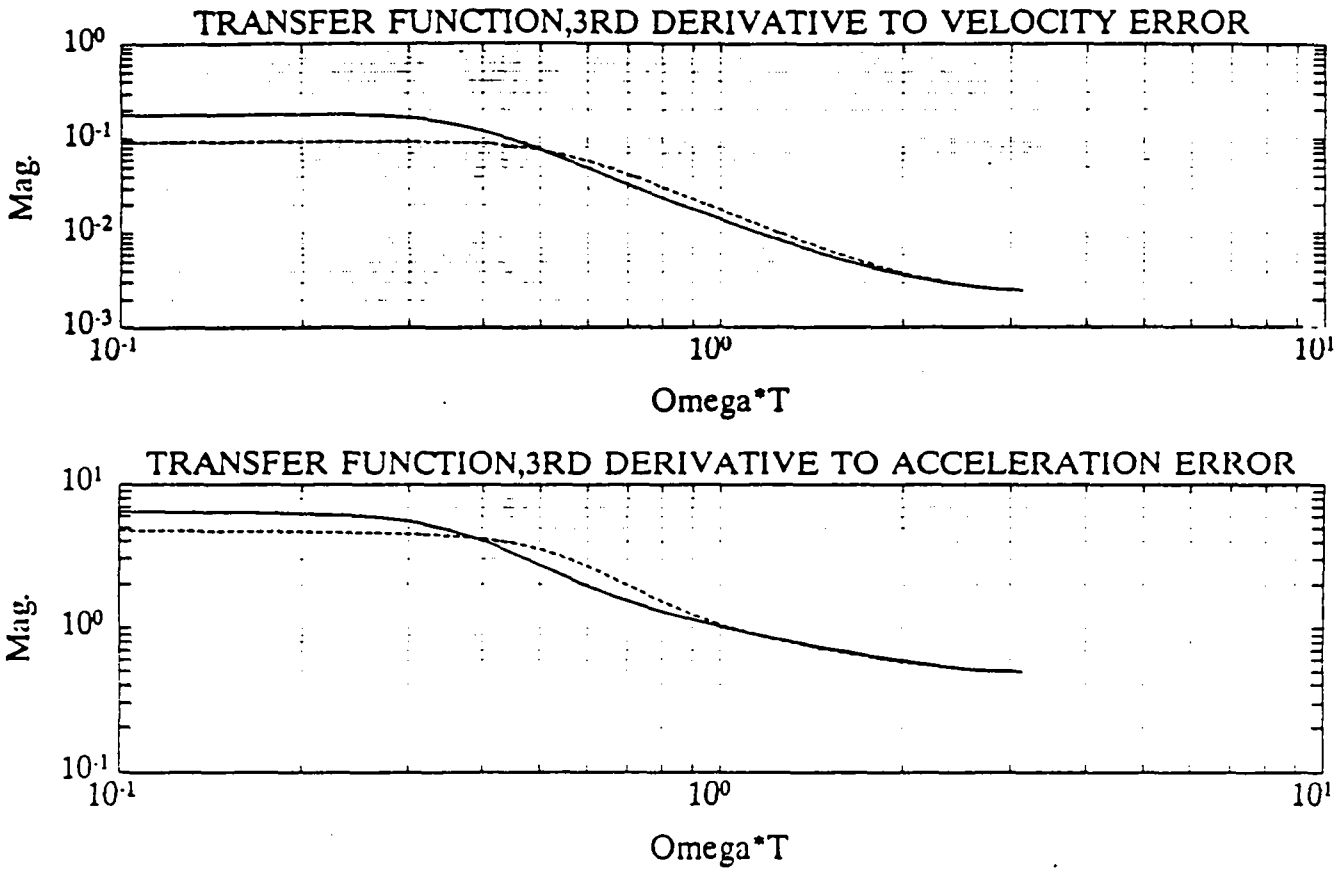


Figure 9 Transfer functions, 3rd derivative to angular velocity and acceleration, triple integrator, $q = 20$ (full curve), $q = 200$ (dotted).

The numbers in bracket are the standard deviation values for the Kalman filters. They vary much more with q than the simulated standard deviations; that is explained by the fact that q is a design parameter rather than the input variance. The input is, of course, the same in all cases. The "theoretical" values do give an order of magnitude estimate of the simulated values, however.

Example 5

This example is similar to the previous one, except that a double-integrator model is used. Acceleration estimates are not produced, of course. Results for $q = 20$ are shown in Fig. 10, and summarized in Table I, along with results for $q = 2$. There is little difference between the results for the two values of q. The estimates are worse than the finite difference estimates, and significantly worse than those obtained with the triple-integrator model.

Example 6

This is a follow-up on Example 5. Since the error standard deviation goes as $1/T$ for the finite-difference estimate of velocity, and since Theorem 1 shows that the dependence is $T^{1/8}$ for the Kalman filter, it follows that a reduction in the sampling period T should give the edge to the Kalman filter. Table I shows that the improvement is as expected.

This example shows that there is a crossover point, in terms of the sampling period, below which the Kalman filter becomes better than the finite-difference estimate. This point occurs at relatively large sampling periods for angular acceleration, between $0.005s$ and $0.001s$ for angular velocity, and at some smaller value for the angle (recall that the covariance goes to zero as $T \rightarrow 0$).

Example 7

In this example, the use of a model is evaluated, specifically the one where $J = 2$, as opposed to $J = 1$. The function $\ddot{\theta}_{\text{mod}}$ is used as an input to the Kalman filter, for the triple-integrator model. The means and standard deviation of the estimation errors are given in Table I, for $q = 20, q = 2$ and $q = 0.2$. As can be seen, the results for the values $q = 0.2$ and

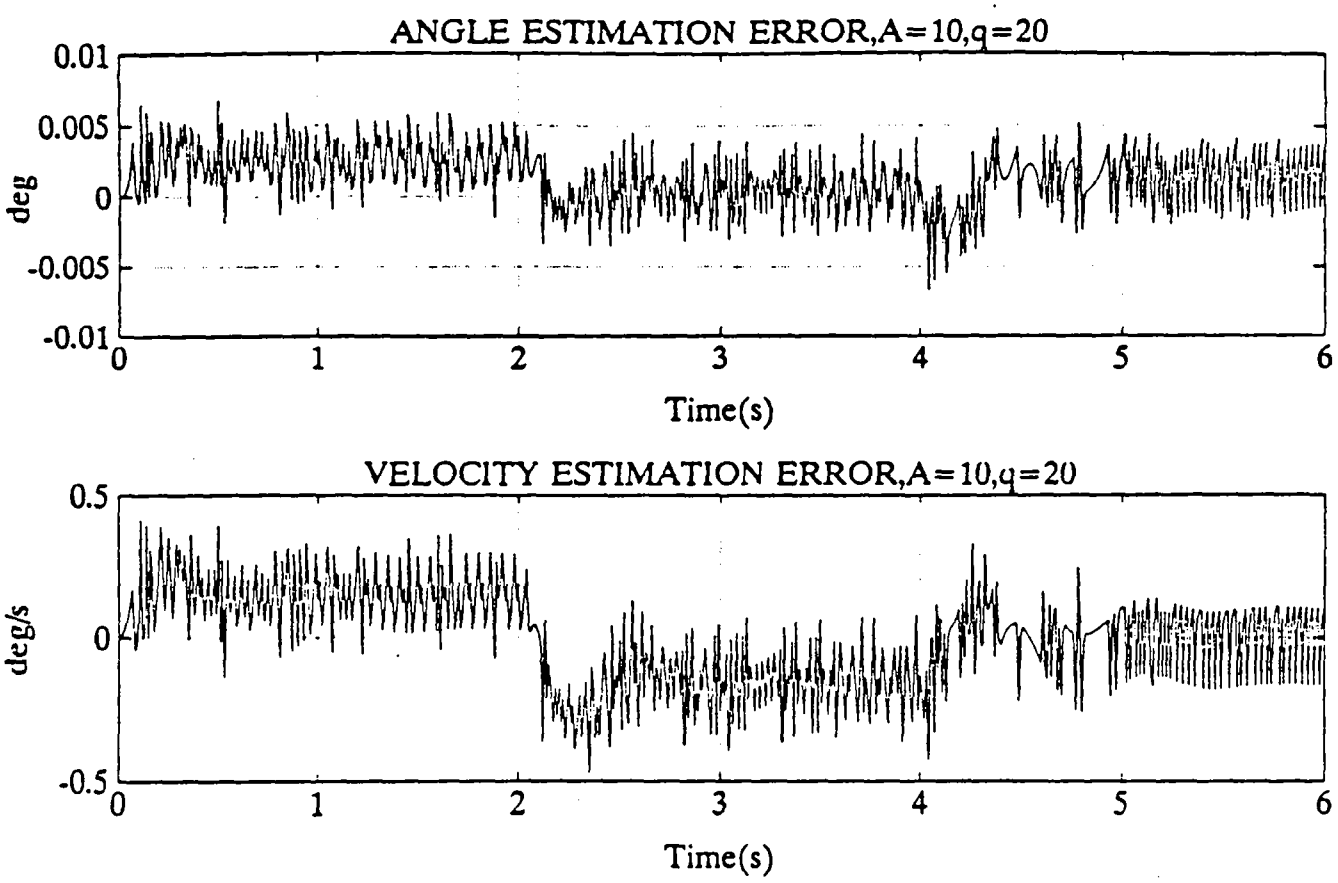


Figure 10 Fast trajectory, Kalman filter error for angle and velocity, $q = 20$, double integrator.

$q = 20$ are significantly worse than for $q = 2$, and the results for $q = 2$ are only marginally better than the results for $q = 200$ without model. Fig. 11 shows that the difference between the actual and modeled third derivative functions is almost as large as the actual function, so that the use of $\dddot{\theta}'_{\text{mod}}$ does not really reduce the magnitude of the function to be modeled by white noise.

Example 8

This example is similar to Example 7, but uses a double-integration model and the modeled acceleration a_{mod}. Results are given in Table I. The improvement is more significant here than was the case in Example 7: one observes a reduction of the order of 30% in the standard deviations. As Fig. 12 shows, the angular acceleration model is more accurate than that of the third derivative of Fig. 11, which explains the improvement. It will usually be the case that the higher derivatives are less accurately modeled.

Example 9

In this and the next example, a slow trajectory is used, with $A = 1$. The peak angular velocity is only 2 rad/s; the actual velocity and its finite-difference estimate are given in Fig. 13.

Since all signals are cut by a factor of 10, compared to the case where $A = 10$, it would make sense to decrease q by a factor of 100. The value $q = 2$ is used, with the triple-integrator model, to design the Kalman filter; the measurement noise covariance is as before. Fig. 14a) shows the encoder output error (dotted) and the Kalman filter estimation error for the angle. Figures 14b) and c) display the angular velocity and acceleration estimation errors. The means and standard deviations are given in Table II.

As expected, the finite-difference data are about the same as in the case where $A = 10$. The Kalman filter estimate means and standard deviations are given for $q = 0.2, 2$ and 20. While there appears to be a minimum at or near $q = 2$, it is a very shallow minimum, and performance does not depend heavily on q. Comparing the results with those obtained for

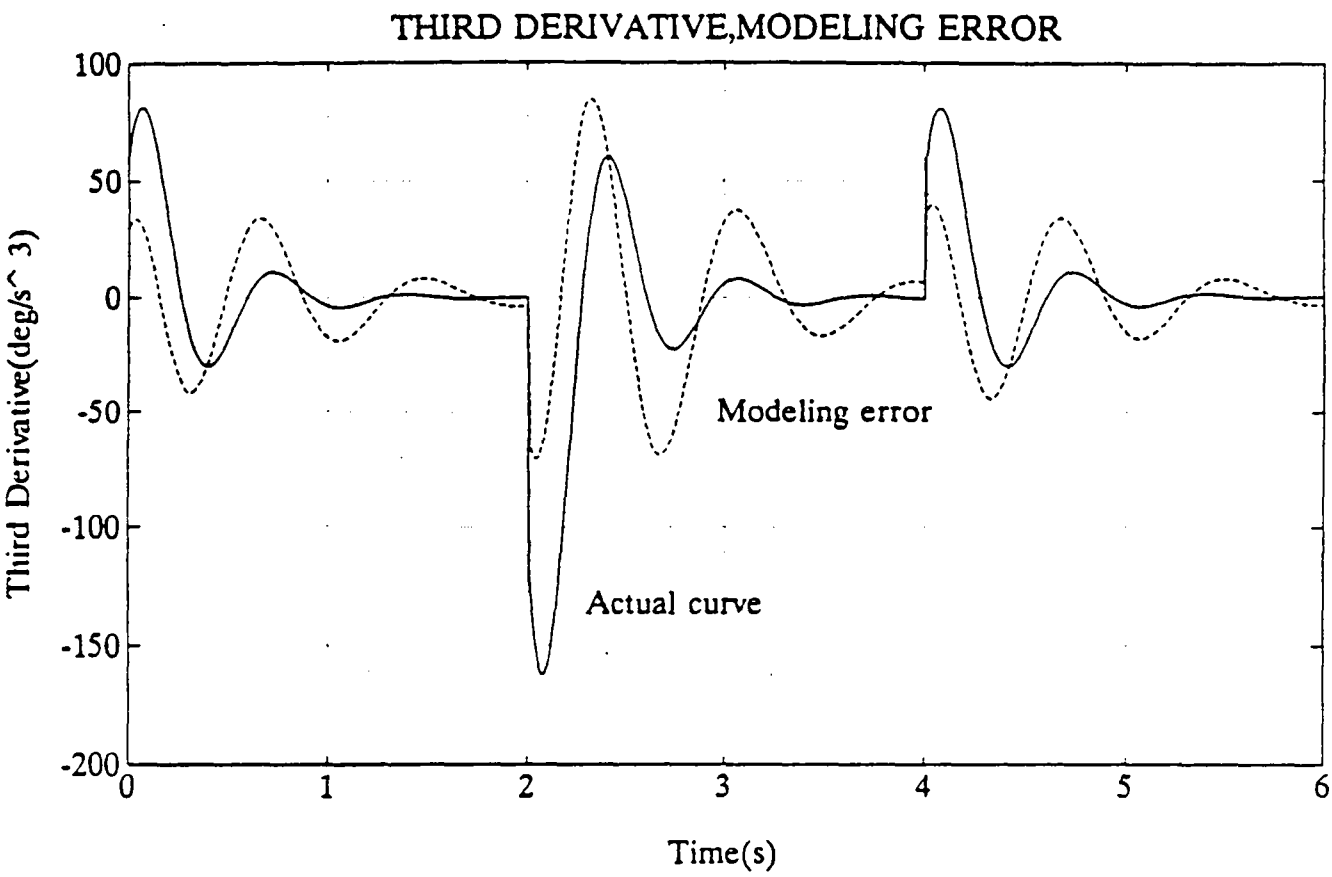


Figure 11 Actual 3rd derivative and error between actual 3rd derivative and its model.

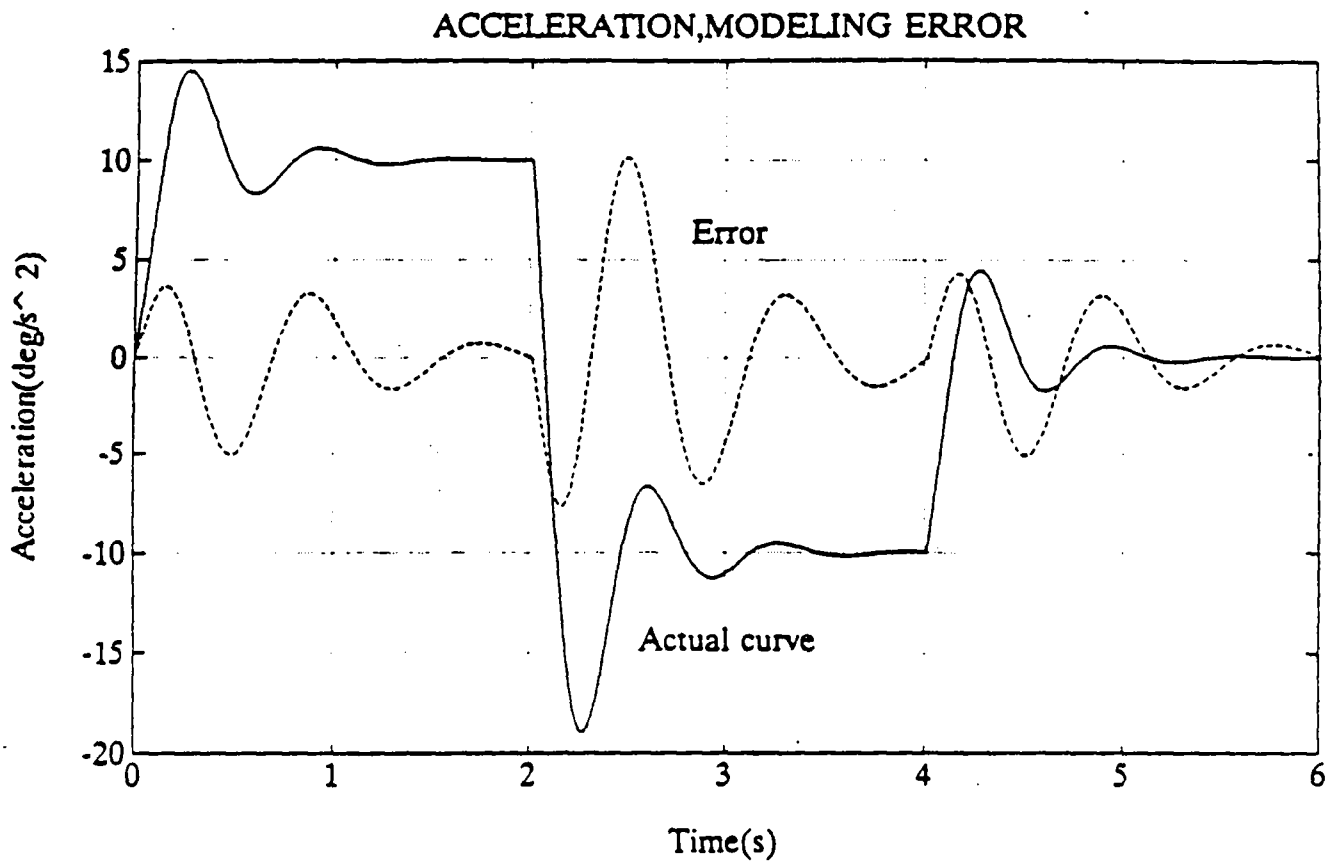


Figure 12 Actual acceleration and error between actual acceleration and its model.

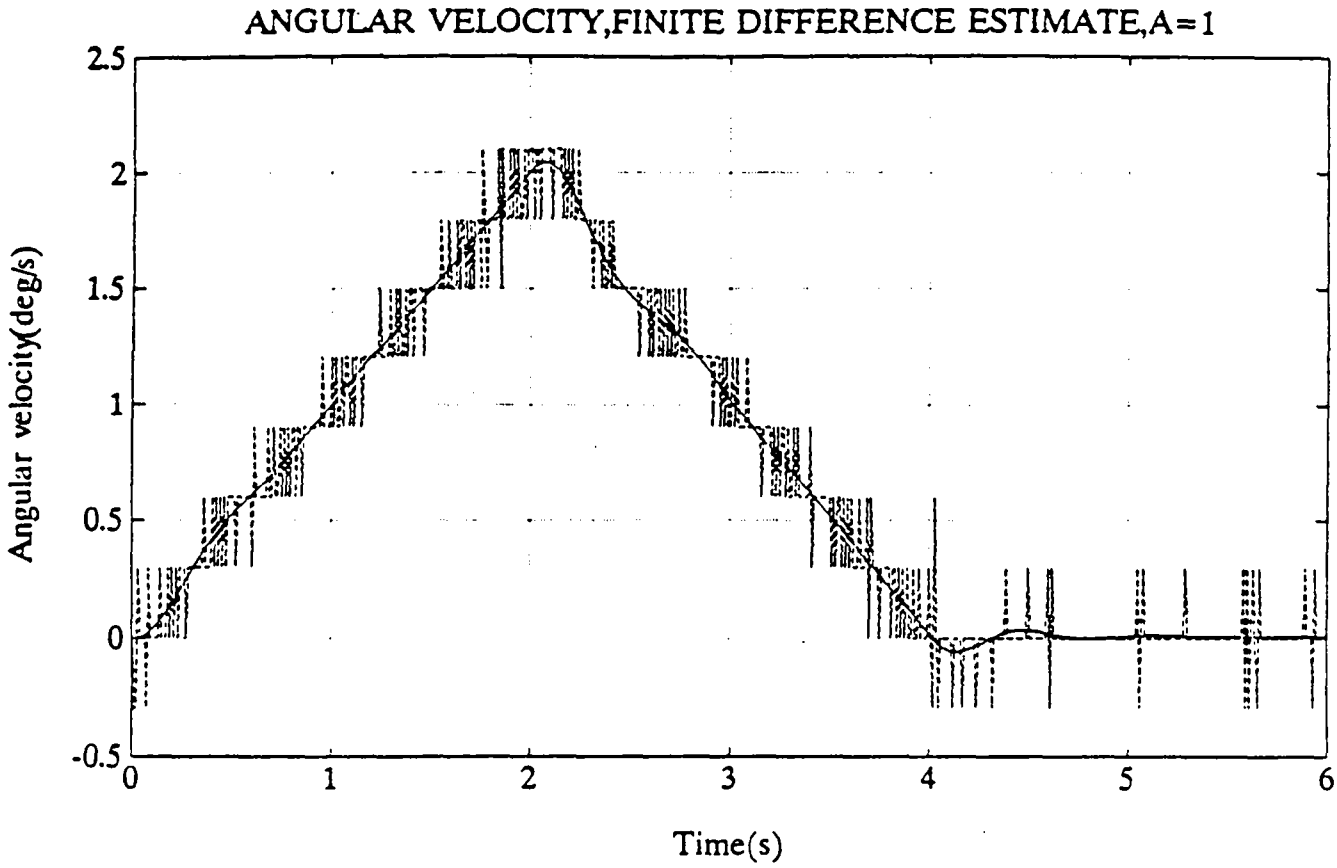


Figure 13 Slow trajectory ($A = 1$), angular velocity and its finite-difference estimate.

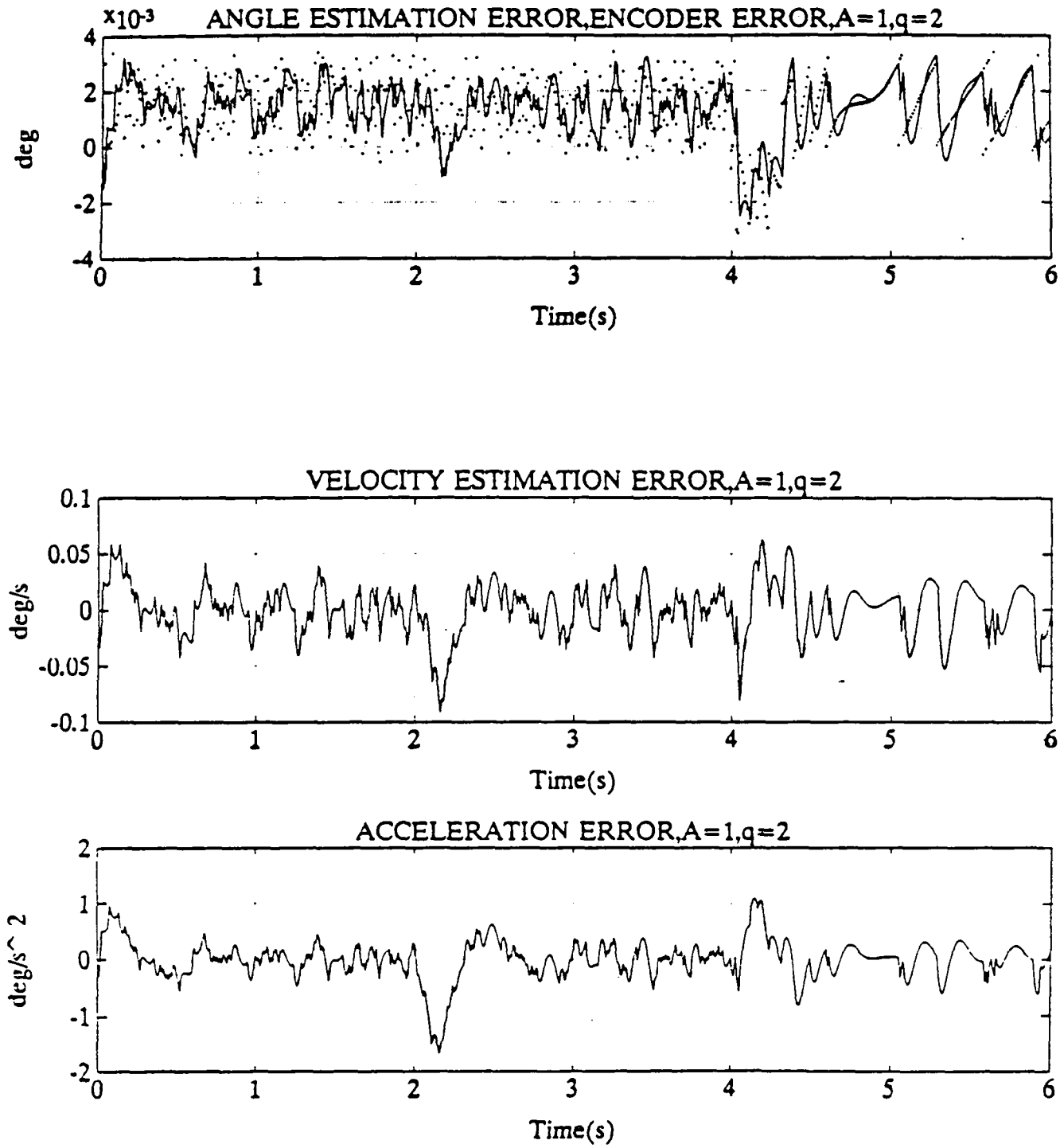


Figure 14 Slow trajectory ($A = 1$), Kalman filter error of angle (dots are encoder error), velocity and acceleration, $q = 2$, triple integrator.

$A = 10$, the standard deviation decreases by about 25% for the angle, by a factor of about 3 for the velocity and 5 for the acceleration. Since all signals are reduced by a factor of 10, the relative error is obviously larger for the smaller signals.

Example 10

The double integrator model is used in this example, with $q = 0.02$ and 0.2 as the Kalman filter design parameters. The means and standard deviations are also shown in Table I. Here again, the results are relatively insensitive to the choice of q. The standard deviation is greater than that obtained with the triple integrator model, but by a factor of less than 2.

2.6 Conclusions

It is permissible to draw certain conclusions from these results, albeit with some caution because of the limited nature of the simulations.

1. Kalman filtering results in only marginal improvement in the angle estimates, over those provided directly by the encoder. The velocity estimate standard deviations are generally improved by a factor of 2 to 4, even more at shorter sampling periods. There is an order of magnitude improvement in the estimate of the angular acceleration.
2. If only velocity estimates are sought, the use of the double integrator is possible, but the standard deviation of the estimation error is 2-4 times worse than in the case of the triple integrator.
3. The results are relatively insensitive to the choice of q. It is reasonable to use the square of the maximum slope of the acceleration or velocity, as the case may be.
4. The use of a model of the third derivative or of the acceleration is beneficial only if, roughly speaking, the modeled signal is in error by less than the signal magnitude. Since that was not the case in our simulations, the improvement was marginal.

Method	A	T	q	Angle (deg)		Angular Veloc. (deg/s)		Angular Accel. (deg/s^2)	
				Mean	Std	Mean	Std	Mean	Std
Finite Diff	10	.01	-	1.33e-3	1.16e-3 (9.01e-4)	2.10e-4	1.37e-1 (2.47e-1)	-8.33e-2	2.17e+1 (3.50e+1)
Kalman, Triple Integ.	"	"	200	1.33e-3	1.33e-3 (1.25e-3)	3.73e-4	6.08e-2 (7.51e-2)	1.62e-3	1.85 (3.10)
"	"	"	20	1.33e-3	1.27e-3 (9.39e-4)	3.13e-4	6.63e-2 (3.96fe-2)	2.44e-4	2.04 (1.14)
Kalman, Double Integ.	"	"	20	1.34e-3	2.22e-3 (4.86e-3)	7.06e-4	1.77e-1 (16.43e-1)	-	-
"	"	"	2	1.34e-3	2.10e-3 (2.10e-3)	6.92e-4	1.77e-1 (2.18e-1)	-	-
Finite Diff.	"	.005	-	1.19e-3	1.27e-3 (9.01e-4)	2.10e-4	2.60e-1 (4.95e-1)	-	-
Kalman, Double Integ.	" "	" "	2 -	1.19e-3	1.50e-3 (1.25e-3)	2.13e-4	1.31e-1 (1.74e-1)	-	-
Kalman, Triple Int, with Model	"	.01	0.2	1.36e-3	6.51e-3	8.63e-4	2.11e-1	7.45e-3	3.38
"	"	"	2	1.34e-3	1.32e-3	4.05e-4	5.75e-2	2.95e-3	1.79
"	"	"	20	-6.62e-2	6.76e-2	3.03e-4	1.11e-1	5.67e-3	1.644
Kalman, double Int, with Model	"	"	0.02	1.34e-3	2.84e-3	6.76e-4	1.58e-1	-	-
"	"	"	0.2	1.34e-3	1/51e-3	5.56e-4	1.14e-1	-	-

TABLE I

Method	A	T	q	Angle (deg)		Angular Veloc. (deg/s)		Angular Accel. (deg/s)	
				Mean	Std	Mean	Std	Mean	Std
Finite Diff	1	.01	-	1.34e-3	1.40e-3	6.69e-4	1.28e-1	-4.8.3e-2	2.19e+1
Kalman, Triple Integ.	"	"	0.2	1.34e-3	1.12e-3	1.27e-4	2.80e-2	-1.11e-3	4.05e-1
"	"	"	2	1.34e-3	1.12e-3	3.88e-6	2.34e-2	-2.23e-3	3.57e-1
"	"	"	20	1.34e-3	1.12e-3	6.93e-5	3.16e-2	-7.24e-4	5.24e-1
Kalman, Double Integ.	"	"	.02	1.34e-3	1.28e-3	1.86e-4	4.00e-2	-	-
"	"	"	.2	1.34e-3	1.24e-3	1.67e-4	4.08e-2	-	-

TABLE II

Chapter 3 ESTIMATION USING PULSE ARRIVAL TIMES

3.1 The Inverse-Time Method

The inverse-time method calculates angular velocity from the duration of the interval between successive pulses. With reference to Fig. 1, the angular velocity estimate is

$$\hat{v}(t_{n+1}) = \frac{(n+1)\theta_m + \epsilon_{n+1} - n\theta_m - \epsilon_n}{t_{n+1} - t_n} \tag{36}$$

In practice, time measurement is inexact; that is taken into account as a measurement error δt_n, assumed to be a zero-mean, white process with variance r_1. Thus, Equation (36) becomes

$$\hat{v}(t_{n+1}) = \frac{\theta_m + \epsilon_{n+1} - \epsilon_n}{t_{n+1} + \delta t_{n+1} - t_n - \delta t_n} \tag{37}$$

One linearizes, under the assumption that $|\delta t_{n+1} - \delta t_n| \ll |t_{n+1} - t_n|$, to obtain

$$\hat{v}(t_{n+1}) = \frac{\theta_m}{t_{n+1} - t_n}\left[1 + \frac{\epsilon_{n+1} - \epsilon_n}{\theta_m} - \frac{\delta t_{n+1} - \delta t_n}{t_{n+1} - t_n}\right] \tag{38}$$

If $t_{n+1} - t_n$ is small, $\theta_m/(t_{n+1} - t_n)$ is a good approximation for $v(t_{n+1})$, as the Taylor series of Equation (6) showed. To the extent that this is true, $\hat{v}(t_{n+1})$ is an unbiased estimate of $v(t_{n+1})$. The standard deviation of the error is

$$\begin{aligned} \text{Std. dev } (v - \hat{v}) &= |v(t_{n+1})| \left[\frac{2r}{\theta_m^2} + \frac{2r_1}{(t_{n+1} - t_n)^2}\right]^{1/2} \\ &= |v(t_{n+1})| \left[\frac{2r}{\theta_m^2} + \frac{2v^2(t_{n+1})r_1}{\theta_m^2}\right]^{1/2} \end{aligned} \tag{39}$$

The corresponding expressions for acceleration are

$$\hat{a}(t_{n+1}) = \Big|v(t_{n+1})\Big(1 + \frac{\epsilon_{n+1}) - \epsilon_n}{\theta_m} - \frac{\delta t_{n+1} - \delta t_n)}{t_{n+1} - t_n}\Big) - v(t_n)$$
$$\Big(1 + \frac{\epsilon_n - \epsilon_{n-1}}{\theta_m} - \frac{\delta t_n - \delta t_{n-1}}{t_n - t_{n-1}}\Big)\Big]/(t_{n+1} - t_n)$$

and

$$\text{Std. dev } (a - \hat{a}) = \frac{|a(t_{n+1})|}{|v(t_{n+1}) - v(t_n)|}\Big[2(v^2(t_{n+1}) + v(t_{n+1})v(t_n) + v^2(t_{n-1}))\frac{r}{\theta_m^2}$$
$$+ 2(v^4(t_{n+1}) + v^2(t_{n+1})v^2(t_n) + v^4(t_N))\frac{r_1^2}{\theta_m^2}\Big]^{1/2} \tag{40}$$

Example 11

Consider the encoder of Example 1, with $\theta_m = 0.003$ deg, $\epsilon_m = \theta_m/4, r = \epsilon_m^2/6$. Assume time measurements are accurate to the microsecond, or $r_1 = 10^{-12}s^2$.

The bracket in Equation (39) is the relative error. The two terms in the bracket are

$$\frac{2r}{\theta_m^2} = \frac{1}{48}$$
$$2\frac{v^2 r_1}{\theta_m^2} = 0.222 \times 10^{-6}v^2$$

For the last quantity, due to time measurement error, to be comparable to the other, v would need to be of the order of 300 deg/s. A (reasonable) velocity of 30 deg/s corresponds to $100\mu s$ between pulses, so the $1\mu s$ error is not serious.

In this example, the relative error is about 14%, contributed almost entirely by the encoder level error.

Now consider the acceleration, for the specific values $a = 10$ deg/s^2 and $v = 30$ deg/s. From $a = \frac{v(t_{n+1}) - v(t_n)}{t_{n+1} - t_n}$, and since $t_{n+1} - t_n = 0.0001s$, one obtains $v(t_{n+1}) - v(t_n) = 0.001$. Since $v(t_{n+1}) \approx v(t_n)$, Equation (40) becomes

$$\text{Std. dev } (a - \hat{a}) = \frac{|a(t_{n+1})|}{0.001}\Big[6v^2(t_{n+1})\frac{r}{\theta_m^2} + 6v^4(t_{n+1})\frac{r_1}{\theta_m^2}\Big]^{1/2}$$
$$\frac{|a(t_{n+1})|}{0.001}\Big[56.2 + 0.534\Big]^{1/2}$$
$$= 7530|a(t_{n+1})|$$

which shows this estimate is practically useless.

3.2 Filtering

Clearly, the situation calls for filtering. It is possible, of course, to generate the raw estimates and then filter; however, given a model of the signal, the optimum filter is the Kalman filter. Kalman filtering was applied to this problem in [9], to good effect. In this case, the observations are not equally spaced in time: an output is generated whenever an encoder pulse is produced, an event whose occurence time depends on velocity. This means that the full, time-varying Kalman filter must be used. It is difficult in practice to propagate the Riccati equation at such a fast rate, so the application of the full Kalman filter is limited to low-velocity, or low-resolution, applications.

It is pertinent to investigate the usefulness of the stationary Kalman filter developed previously, in particular the asymptotic form for small time intervals. In order to do that it is necessary to examine the asymptotic form of the dynamic Riccati equation.

Recall that, from Theorem 1, the solution of the Riccati equation tends to the form $P_{ij} = P^*_{ij} T^{\frac{2n-i-j+1}{2n}}$, where P^*_{ij} satisfies a continuous-time Riccati equation. The matrix P is actually the a priori covariance $P(t_{k+1}|t_k)$, where t_k is the kth pulse occurence time. With each t_k is associated a duration $\Delta_k = t_{k+1} - t_k$, so that both P and Δ are time-varying. Note that Δ is distinct from T: T is the control system sampling period, while Δ is the interval between pulses. If, as is normally the case P and Δ change slowly compared to the sampling rate, it is possible to speak of derivatives $\dot{P}$ and $\dot{\Delta}$, defined by the usual finite difference formulae. The following result is then established:

Theorem 3 The elements of the a priori covariance matrix P^* satisfies the equation

$$\Delta^{1/2n} \dot{P}^*_{ij} = [A_0 P^* + P^* A_0^T - P^* C^T R^{-1} C P^* + q\Gamma\Gamma^T]_{ij} + \left(\frac{2n-i-j+1}{2n}\right) P^*_{ij} \frac{\dot{\Delta}}{\Delta^{\frac{2n-1}{2n}}} \tag{41}$$

where A_0 is the system A-matrix with $a_0 = a_1 = \cdots = a_{m-1} = 0$, and $[]_{ij}$ refers to the ij element.

Proof: The Riccati equation, for small Δ, is

$$\begin{aligned} P(t_{k+1}|t_k) &= (I + A\Delta_k)[P(t_k|t_{k-1}) - P(t_k|t_{k-1})C^T(CP(t_k|t_{k-1})C^T + R)^{-1}CP(t_k|t_{k-1})] \\ &\quad (I + A\Delta_k)^T + q\Gamma\Gamma^T\Delta_k \\ &= P(t_k|t_{k-1}) + A\Delta_k P(t_k|t_{k-1}) + P(t_k|t_{k-1})A^T\Delta_k \\ &\quad - P(t_k|t_{k-1})C^T R^{-1}CP(t_k|t_{k-1}) + q\Gamma\Gamma^T\Delta_k \end{aligned}$$

Now,

$$\begin{aligned} \frac{P(t_{k+1}|t_k) - P(t_k|t_{k-1})}{\Delta_k} &= AP(t_k|t_{k-1}) + P(t_k|t_{k-1})A^T \\ &\quad + Q\Gamma\Gamma^T - P(t_k|t_{k-1})C^T\frac{R^{-1}}{\Delta_k}CP(t_k|t_{k-1}) \end{aligned}$$

or

$$\dot{P} = AP + PA^T + q\Gamma\Gamma^T - PC^T\frac{R^{-1}}{\Delta}CP \tag{42}$$

From Theorem 1, $P_{ij} = P^*_{ij}\Delta^{\frac{2n-i-j+1}{2n}}$ so that

$$\dot{P}_{ij} = \Delta^{\frac{4n-i-j+1}{2n}}\dot{P}^*_{ij} + \left(\frac{2n-i-j+1}{2n}\right)P^*_{ij}\Delta^{\frac{2n-i-j+1}{2n}}\dot{\Delta} \tag{43}$$

For $i, j < n$, combining Equations (42) and (43) yield, to lowest power of T,

$$\begin{aligned} \Delta^{\frac{4n-i-j+1}{2n}}\dot{P}^*_{ij} &= \left(-\frac{P^*_{1i}P^*_{1j}}{R} + P^*_{j,i+1} + P^*_{i,j+1}\right)\Delta^{\frac{4n-i-j}{2n}} \\ &\quad - \frac{2n-i-j+1}{2n}P^*_{ij}\Delta^{\frac{2n-i-j+1}{2n}}\dot{\Delta} \end{aligned}$$

or, dividing by $\Delta^{\frac{4n-i-j}{2n}}$,

$$\Delta^{1/2n}\dot{P}^*_{ij} = \left(-\frac{P^*_{ij}P^*_{ij}}{R} + P^*_{j,i+1} + P^*_{i,j+1}\right) - \left(\frac{2n-i-j+1}{2n}\right)P^*_{ij}\frac{\dot{\Delta}}{\Delta^{\frac{2n-1}{2n}}}$$

The cases where i or j (or both) are equal to n follow in similar fashion.

Remarks:

1. At constant Δ, P^* tends to the steady state, as expected, and does so with a time constant of the order of $\Delta^{1/2n}$, i.e. rapidly if Δ is small.

2. The Riccati equation is in quasi steady state if $|\dot{\Delta}|/\Delta^{\frac{2n-1}{2n}}$ is small, say less than $M, M > 0$. This can be translated into a condition on the angular velocity and acceleration, by using the approximation $v \approx \theta_m/\Delta$. It is straightforward to show that the condition becomes

$$|a| < M\frac{|v|^{1+1/2n}}{\theta_m^{1/2n}}.$$

Taken together, these remarks suggest that a calculation of the gain and covariance, on an instant-by-instant basis using the steady-state results, is at least plausible. Thus it is permissible to speak of a covariance that depends instantaneously on Δ. It is also possible to do the same for the Kalman gain; this was not pursued, because the approximation scheme of the next section was thought to be better.

3.3 A Suboptimal Filtering Solution

A suboptimal solution is presented, motivated by the following consideration: it is not necessary to have estimates at each pulse arrival time, only at the sampling intervals of the control system. This points to the possibility of a "batch" processing of all data that become available during the sampling interval T. These data, of course, are the pulse occurrence times. Depending on the angular velocity, there can be anywhere from 0 to several hundred pulses generated in the interval T.

The idea is based on the fact that, in the absence of plant noise, the Kalman filter solution is equivalent to that of the Weighted Least Squares problem, which admits a batch as well as a sequential solution. Plant noise is therefore considered absent ($q = 0$) during each sampling interval, $nT < t \leq (n+1)T$. The estimate error covariance is updated (upwards) at each sampling instant, to counteract the effect of ignoring plant noise.

In terms of modeling, this is equivalent to assuming a piecewise quadratic model (piecewise linear, for the double integrator); the joint angle function is described as $a_0 + a_1 t + \frac{1}{2}a_2 t^2$

within a sampling period, but the coefficients change from one period to the next. Given the deterministic behavior of robot motion, this type of model does not appear inappropriate.

The model is

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w$$
$$y = [1 \quad 0 \quad 0]\mathbf{x} + e \tag{44}$$

With $w(t) = 0$,

$$y(t_k) = x_1(0) + x_2(0)t_k + \frac{1}{2}x_3(0)t_k^2 + e(t_k).$$

Define

$$J(\mathbf{x}(0)) = \sum_{k=1}^{N} \frac{1}{R}(y(t_k) - \mathbf{x}^T(0)\ \underline{\phi}\ (t_k))^2 + (\mathbf{x}(0) - \mathbf{x}_0)^T P_0^{-1}(\mathbf{x}(0) - \mathbf{x}_0) \tag{45}$$

where

$$\underline{\phi}^T\ (t) = [1 \quad t \quad \tfrac{1}{2}t^2]$$

$\mathbf{x}_0$ = Prior estimate of $\mathbf{x}(0)$

P_0 = Error covariance of prior estimate,

assumed to be positive definite.

The Least Squares solution is

$$\hat{\mathbf{x}}(0) = \arg\min J(\mathbf{x}(0))$$

or

$$\hat{\mathbf{x}}(0) = \left[\frac{1}{R}\sum_{k=1}^{N} \underline{\phi}\ (t_k)\ \underline{\phi}^T\ (t_k) + P_0^{-1}\right]^{-1} \left[\frac{1}{R}\sum_{k=1}^{n} y(t_k)\ \underline{\phi}\ (t_k) + P_0^{-1}\mathbf{x}_0\right] \tag{46}.$$

If $\mathbf{x}(0)$ is Gaussian, this is the optimum as well as the Least Squares solution.

With $\tilde{\mathbf{x}}(0) = \mathbf{x}(0) - \hat{\mathbf{x}}(0)$, it is well known that

$$\text{cov}\ (\tilde{\mathbf{x}}(0)) = \left[\frac{1}{R}\sum_{k=1}^{N} \underline{\phi}\ (t_k)\ \underline{\phi}^T\ (t_k) + P_0^{-1}\right]^{-1}. \tag{47}$$

The optimum estimate of $\mathbf{x}(T)$ is

$$\hat{\mathbf{x}}(T) = e^{AT}\hat{\mathbf{x}}(0) \tag{48}$$

where A is the system matrix in Equation (44). This is the estimate that would be generated by the Kalman filter, with $q = 0$.

Substituting for $\underline{\phi}$ in Equation (47),

$$\text{cov}\ (\tilde{\mathbf{x}}(0)) = \frac{1}{N}M^{-1} \tag{49}$$

$$M = \frac{1}{R}\begin{bmatrix} 1 & \frac{1}{N}\Sigma t_k & \frac{1}{2N}\Sigma t_k^2 \\ \frac{1}{N}\Sigma t_k & \frac{1}{N}\Sigma t_k^2 & \frac{1}{2N}\Sigma t_k^3 \\ \frac{1}{2N}\Sigma t_k^2 & \frac{1}{2N}\Sigma t_k^3 & \frac{1}{4N}\Sigma t_k^4 \end{bmatrix} + \frac{P_o^{-1}}{N} \tag{50}$$

It is assumed that M is nonsingular, which is the case since P_0 is assumed positive definite.

Equation (46) becomes

$$\hat{\mathbf{x}}(0) = \frac{M^{-1}}{R}\left\{\begin{bmatrix} \frac{1}{N} & \Sigma y(t_k) \\ \frac{1}{N} & \Sigma y(t_k)t_k \\ \frac{1}{2N} & \Sigma y(t_k)t_k^2 \end{bmatrix} + \frac{R}{N}P_0^{-1}\mathbf{x}_0\right\} \tag{51}$$

In the case of the double integrator (velocity estimate only), Equations (50) and (51) become

$$M = \frac{1}{R}\begin{bmatrix} 1 & \frac{1}{N}\Sigma t_k \\ \frac{1}{N}\Sigma t_k & \frac{1}{N}\Sigma t_k^2 \end{bmatrix} + \frac{P_o^{-1}}{N} \tag{52}$$

and

$$\hat{\mathbf{x}}(0) = \frac{M^{-1}}{R}\left\{\begin{bmatrix} \frac{1}{N}\Sigma y(t_k) \\ \frac{1}{N}\Sigma y(t_k)t_k \end{bmatrix} + \frac{R}{N}P_0^{-1}\mathbf{x}_0\right\} \tag{53}$$

Roughly speaking, if the initial estimates are good, P_0 is "small", P_0^{-1} is "large" and $\hat{\mathbf{x}}(0) \approx \mathbf{x}_0$ unless N is large. In essence, the estimate changes more rapidly at high pulse rate, i.e. high velocity.

Using Equation (48),

$$\text{cov}\ (\tilde{\mathbf{x}}(T)) = e^{AT}\ \text{cov}\ (\tilde{\mathbf{x}}(0))e^{A^T T} \tag{54}$$

The algorithm is applied recursely, by setting $\hat{\mathbf{x}}(T) \rightarrow \mathbf{x}_0$ and cov $(\hat{\mathbf{x}}(T)) \rightarrow P_0$ for the next sampling interval.

This produces a monotonically decreasing set of covariance matrices: a Kalman filter for an observable system, with $Q = 0$ "closes up", i.e. produces sequences of covariances and gains that tend to zero. In order to counteract this effect, the error covariance at $t = T$ is now calculated, for the case where the Least Square algorithm is applied with a non-zero q.

The stochastic component neglected up to now is described by

$$\begin{aligned} \dot{\mathbf{x}}_s &= A\mathbf{x}_s + \Gamma w \\ y_s &= C\mathbf{x}_s \end{aligned} \tag{55}$$

with $\mathbf{x}_s(0) = \mathbf{0}$. Inserting $y(t) = \underline{\phi}^T(t)\mathbf{x}(0) + e(t) + y_s(t)$ in Equation (46) yields

$$\begin{aligned} \hat{\mathbf{x}}(0) = &\left[\frac{1}{R}\Sigma\, \underline{\phi}(t_k)\, \underline{\phi}^T(t_k) + P_0^{-1}\right]^{-1} \left[\frac{1}{R}\Sigma\, \underline{\phi}(t_k)\, \underline{\phi}^T(t_k)\mathbf{x}(0)\right. \\ &\left. \frac{1}{R}\Sigma\, \underline{\phi}(t_k)e(t_k) + \frac{1}{R}\Sigma\, \underline{\phi}(t_k)y_s(t_k) + P_0^{-1}\mathbf{x}(0) + P_0^{-1}(\mathbf{x}_0 - \mathbf{x}(0))\right] \\ = &\mathbf{x}(0) + \tilde{\mathbf{x}}_1(0) + \tilde{\mathbf{x}}_2(0) \end{aligned}$$

where

$$\tilde{\mathbf{x}}_1(0) = \left[\frac{1}{R}\Sigma\, \underline{\phi}\underline{\phi}^T + P_o^{-1}\right]^{-1} \left[\frac{1}{R}\Sigma\, \underline{\phi}\, e + P_0^{-1}(\mathbf{x}_0 - \mathbf{x}(0))\right] \tag{56}$$

$$\tilde{\mathbf{x}}_2(0) = \left[\frac{1}{R}\Sigma\, \underline{\phi}\underline{\phi}^T + P_o^{-1}\right]^{-1} \left[\frac{1}{R}\Sigma\, \underline{\phi}\, y_s\right] \tag{57}$$

Note that $\tilde{\mathbf{x}}_1(0)$ and $\tilde{\mathbf{x}}_2(0)$ are independent, since the first depends on $e(t)$ and $\mathbf{x}_0 - \mathbf{x}(0)$, while the second is a function of $w(t)$.

Using Equation (48),

$$\begin{aligned} \tilde{\mathbf{x}}(T) &= e^{AT}\mathbf{x}(0) + \mathbf{x}_s(T) - e^{AT}\hat{\mathbf{x}}(0) \\ &= -e^{AT}\tilde{\mathbf{x}}_1(0) - e^{AT}\tilde{\mathbf{x}}_2(0) + \mathbf{x}_s(T). \end{aligned}$$

The error covariance is

$$\mathrm{E}[\tilde{\mathbf{x}}(T)\tilde{\mathbf{x}}^T(T)] = e^{AT}\mathrm{E}[\tilde{\mathbf{x}}_1(0)\tilde{\mathbf{x}}_1^T(0)]e^{A^T T} +$$

$$e^{AT}\mathrm{E}[\tilde{\mathbf{x}}_2(0)\tilde{\mathbf{x}}_2^T(0)]e^{A^TT} + \mathrm{E}[\mathbf{x}_s(T)\mathbf{x}_s^T(T)] - e^{AT}\mathrm{E}[\tilde{\mathbf{x}}_2(0)\mathbf{x}_s^T(T)]$$
$$- \mathrm{E}[\mathbf{x}_s(T)\tilde{\mathbf{x}}_2^T(0)]e^{A^TT} \tag{58}$$

From Equation (56),

$$\mathrm{E}[\tilde{\mathbf{x}}_1(0)\tilde{\mathbf{x}}_1^T(0)] = \left[\frac{1}{R}\Sigma\,\underline{\phi}\underline{\phi}^T + P_0^{-1}\right]^{-1} \tag{59}$$

$$\mathrm{E}[\tilde{\mathbf{x}}_2(0)\tilde{\mathbf{x}}_2^T(0)) = \left[\frac{1}{R}\Sigma\,\underline{\phi}\underline{\phi}^T + P_0^{-1}\right]^{-1}\frac{1}{R^2}$$
$$(\Sigma\Sigma\,\underline{\phi}\,(t_k)\mathrm{E}(y_s(t_k)y_s(t_j)]\,\underline{\phi}^T\,(t_j))\left[\frac{2}{R}\Sigma\,\underline{\phi}\underline{\phi}^T + P_0^{-1}\right]^{-1} \tag{60}$$

To calculate $\mathrm{E}[y_s(t_k)y_s(t_j))]$, use Equation (55):

$$\mathrm{E}[y_s(t_k)y_s(t_j)] = C\mathrm{E}[\mathbf{x}_s(t_k)\mathbf{x}_s^T(t_j)]C^T \tag{61}$$

For $t_k \geq t_j$,

$$\mathrm{E}[\mathbf{x}_s(t)k)\mathbf{x}_s^T(t_j)] = e^{A(t_k - t_j)}P_s(t_j) \tag{62}$$

where $P_s(t_j) = \mathrm{E}[\mathbf{x}_s(t_j)\mathbf{x}_s^T(t_j)]$ satisfies the Lyapunov equation

$$\dot{P}_s = AP_s + P_sA^T + q\Gamma\Gamma^T, P_s(0) = 0. \tag{63}$$

Finally,

$$\mathrm{E}[\tilde{\mathbf{x}}_2(0)\mathbf{x}_s^T(T)] = \left[\frac{1}{R}\Sigma\,\underline{\phi}\underline{\phi}^T + P_0^{-1}\right]^{-1}\frac{1}{R}\Sigma\,\underline{\phi}\,(t_k)C\mathrm{E}[\mathbf{x}_s(t_k)\mathbf{x}_s^T(T)] \tag{64}$$

For the triple integrator model,

$$e^{At} = \begin{bmatrix} 1 & t & \frac{1}{2}t^2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}$$

and Equation (63) can be solved analytically, to yield

$$P_s(t) = q\begin{bmatrix} \frac{1}{20}t^5 & \frac{1}{8}t^4 & \frac{1}{6}t^3 \\ \frac{1}{8}t^4 & \frac{1}{3}t^3 & \frac{1}{2}t^2 \\ \frac{1}{6}t^3 & \frac{1}{2}t^2 & t \end{bmatrix}$$

and, of course

$$\underline{\phi}^T(t) = [1 \; t \; \frac{1}{2}t^2]$$

For the double integrator,

$$e^{At} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$$

$$P_s(t) = q \begin{bmatrix} \frac{1}{3}t^3 & \frac{1}{32}t^2 \\ \frac{1}{2}t^2 & t \end{bmatrix}$$

$$\underline{\phi}^T(t) = [1 \; t]$$

3.4 Pratical Considerations

As it stands, the Least Squares algorithm would be difficult to apply, because of i) the often ill-conditioned nature of the matrix $\frac{1}{R}\Sigma\, \underline{\phi}\underline{\phi}^T + P_0^{-1}$ and ii) the exhorbitant calculation budget for the covariance correction Equation (58).

The conditioning is addressed, at least partially, by time scaling. To study this, consider

$$M = \frac{1}{N}\sum_{k=1}^{N} \underline{\phi}(t_k)\, \underline{\phi}^T(t_k) = \begin{bmatrix} 1 & \frac{1}{N}\Sigma t_k & \frac{1}{2N}\Sigma t_k \\ \frac{1}{N}\Sigma t_k & \frac{1}{N}\Sigma t_k^2 & \frac{1}{2N}\Sigma t_k^3 \\ \frac{1}{2N}\Sigma t_k^2 & \frac{1}{2N}\Sigma t_k^3 & \frac{1}{4N}\Sigma t_k^4 \end{bmatrix} \qquad (65)$$

for the case where the N instants are equally spaced, with spacing T/N. In that case,

$$M = \begin{bmatrix} 1 & T\frac{(N+1)}{2N} & T^2\frac{(1+3N+2N^2)}{12N^2} \\ T\frac{N+1}{2N} & T^2\frac{(1+3N+2N^2)}{6N^2} & T^3\frac{(1+2N+N^2)}{8N} \\ T^2\frac{(1+3N+2N^2)}{12N^2} & T^3\frac{(1+2N+N^2)}{8N} & T^4\frac{(-1+10N^2+15N^3+6N^4)}{120N^4} \end{bmatrix}$$

It is easy to see that M will be very ill-conditioned if T is either much smaller or much greater than 1. For large N,

$$M \to \begin{bmatrix} 1 & \frac{T}{2} & \frac{T^2}{6} \\ \frac{T}{2} & \frac{T^2}{3} & \frac{T^3}{8} \\ \frac{T^2}{6} & \frac{T^3}{8} & \frac{T^4}{20} \end{bmatrix}.$$

The condition numbers for a few values near $T = 1$ are:

T	Cond (M)
.4	30465
.6	6648
.8	2417
1.0	1182
1.2	705
1.4	486
1.6	374
1.8	314
2.0	280

The value $T = 1$ is chosen for convenience, even though it is not optimal. This means that time is to be expressed using the sampling period as the unit.

To apply scaling to the model equations, let the system be described in controllable canonical form, i.e.,

$$\frac{dx_i}{dt} = x_{i+1}, i = 1, 2, \cdots n-1$$
$$\frac{dx_n}{dt} = -a_0 x_1 - a_1 x_2 - \cdots - a_{m-1} x_n + w.$$

Let $\tau = t/T$. Then,

$$\frac{dx_1}{d\tau} = T x_2.$$

Now, let $x_2' = T x_2; x_2'$ is the normalized velocity. Then,

$$\frac{dx_2'}{d\tau} = T\frac{dx_2}{d\tau} = T^2 x_3.$$

The normalized acceleration is $x_3' = T^2 x_3$, and so on. The last equation is

$$\frac{dx_n'}{d\tau} = -a_0 T^n x_1 - a_1 T^{n-1} x_2' - \cdots - a_{n-1} T x_n' + T^n w.$$

The conclusion is that:

1) The equations in normalized time are the same as the original set, but with a_i replaced by $a_i T^{n-i}$. Since T is small, this lends further support to the use of a multiple integrator model.

ii) The normalized variance of the plant noise is $q' = T^{2n}q$.

Note that normalized derivatives are small. For example, if $T = 0.01s$, time is measured in "centiseconds" (cs). A velocity of 30 deg/s becomes 0.3 deg/cs; an acceleration of 10 deg/s^2 becomes 0.0001 deg/cs^2. This should be borne in mind when programming the algorithm.

The second consideration is the simplification of the covariance calculation. It is clear that the full implementation is much too heavy for practical use. The solution is to do the computation, as a function of N, for equally-spaced intervals. Since the velocity is almost constant over a sampling interval, this step is easily justified. This is carried out in Appendix B, for the triple and double integrators. This is *not* done for the matrix M of Equation (65), which is calculated in full. The fact is that that matrix, which must be inverted, is not very well conditioned. That should be no cause for surprise: after all, the values of the pulse occurrence times carry the information, and such information cannot be simply thrown away when calculating estimates.

3.5 Modification for Low Velocities

The Kalman filter and Least Squares solutions supply estimates from output observations at given times. For an encoder, the fact that a pulse was *not* received in a given time interval provides information, namely that no level was crossed during that particular interval. That information is not used by the Kalman filter or Least Squares algorithms. That is not too important at high angular velocity, but becomes relevant at low speeds. For example, the fact that no pulse has been received for a long period of time would lead one to conclude that the velocity is small, where as the Kalman filter based on a multiple integrator model would estimate the velocity as the last estimate plus the integral of the last acceleration estimate.

A modification, useful at low velocities, is introduced to improve estimation at low speed. This worked out for the triple and double integrator models. The effect of plant noise is assumed to be negligible, which is realistic over small time intervals.

Let $\chi(t)$ be the event defined as the absence of encoder level crossings between 0 and t. By Bayes' rule,

$$p(\mathbf{x}(0)|\chi(t)) = \frac{p(\chi(t)|\mathbf{x}(0))p(\mathbf{x}(0))}{p(\chi(t))} \tag{66}$$

In order to keep complexity within reasonable limits, $p(\mathbf{x}(0))$ is assumed to be Gaussian. The probability $p(\chi(t)|\mathbf{x}(0))$ is not Gaussian, but will be approximated by a Gaussian, so that the a posteriori probability $p(\mathbf{x}(0)|\chi(t))$ is Gaussian.

Given the fact that $p(\chi(t)|\mathbf{x}(0))$ is to be approximated, the expenditure of substantial analytical effort to derive an exact expression is not warranted. With reference to Figure 1, let $\epsilon_{n-1}=\epsilon_n=\epsilon_{n+1}=0$. For the double-integrator plant, $y(t) = x_1(0) + x_2(0)t$ and one may write:

For $n\theta_m < x_1(0) < (n+1)\theta_m$:

$$p(\chi(t)|\mathbf{x}(0)) = \begin{cases} 1 & \text{if } n\theta_m < x_1(0) + x_2(0)t < (n+1)\theta_m \\ 0 & \text{otherwise} \end{cases}$$

where n is any integer.

This function of $\mathbf{x}(0)$ is a rectangular pulse whose location along the axis $x_1(0) + x_2(0)t$ depends on $x_1(0)$. That is illustrated in Figure 15.

Since the denominator $p(\chi(t))$ is only a normalizing factor, Equation (66) calls for multiplying the Gaussian $p(\mathbf{x}(0)$ by a different pulse function for each interval $x_1(0)$ of length θ_m. The result is decidedly non-Gaussian. If each rectangular pulse is replaced by a Gaussian approximation, the result is a "piecewise Gaussian" function, i.e. a function consisting of different Gaussian pieces over the different intervals of $\mathbf{x}_1(0)$. If, however, $p(\mathbf{x}(0))$ were negligibly small for all $\mathbf{x}_1(0)$, say, outside the interval $n\theta_m < x_1(0) + x_2(0)t < (n+1)\theta_m$, then only one of the pulses of $p(\chi(t)|\mathbf{x}(0))$ would need to be approximated, namely that corresponding to that particular interval of $\mathbf{x}_1(0)$. That is in fact the situation if the last encoder level crossed was n, and if it was traversed at positive angular velocity. The density $p(\mathbf{x}(0))$ in that case will normally show an angle estimation ($= x_1(0)$) that is very close to

$n\theta_m$. If the sign of the angular velocity is not known, then $x_1(0)$ is known to be near θ_m, but it is not possible to select the pulse for $n\theta_m < x_1(0) + x_2(0)t < (n+1)\theta_m$ over the pulse for $(n-1)\theta_m < x_1(0) + x_2(0)t < n\theta_m$. Since the approximation must be a single Gaussian curve, one solution is to combine both pulses, and retain one rectangular pulse between the limits $(n-1)\theta_m$ and $(n+1)\theta_m$.

The simplest way to derive a Gaussian approximation is to fit the mean and variance. For the case when the sign of the velocity is not known, the function to be approximated is a uniform distribution, running from $(n-1)\theta_m$ to $(n+1)\theta_m$. The mean is $n\theta_m$, the variance is $\theta_m^2/3$. Therefore, one uses

$$\frac{p(X(t)|\mathbf{x}(0))}{p(X(t))} \approx \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{(x_1(0) + x_2(0)t - n\theta_m)^2}{2\sigma^2}$$

where $\sigma^2 = \theta_m^2/3$.

If the sign of the velocity is known, the mean is $(n \pm \frac{1}{2})\theta_m$, the variance is $\theta_m^2/12$. For the case of the triple integrator, the output is $(n \pm \frac{1}{2})\theta_m$, the variance is $\theta_m^2/12$.

For the case of the triple integrator, the output is $x_1(0) + x_2(0)t + \frac{1}{2}x_3(0)t^2$. This function need not be monotonic, so that its maximum or minimum may be reached at some time between 0 and t. Since maxima and minima are isolated points, they occur in relatively few sampling intervals and this possibility will be ignored. In any case, the difference between the value at t and the value at an extremum between 0 and t can always be kept below a desired bound, given bounds on $x_2(0)$ and $x_3(0)$, by making t small enough. In other words, the probability update of Equation (66) can be made more often. In any case, for the triple integrator, one uses

$$\frac{p(X(t)|\mathbf{x}(0))}{p(X(t))} \approx \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{(x_1(0) + x_2(0)t + \frac{1}{2}x_3(0)t^2 - n\theta_m)^2}{2\sigma^2} \tag{67}$$

Remark: It is possible to do the calculation under the assumption that ϵ_n satisfies a triangular distribution. The analysis is complicated by the fact that, for $n\theta_m - \epsilon_m < x_1(0) + x_2(0)t <$

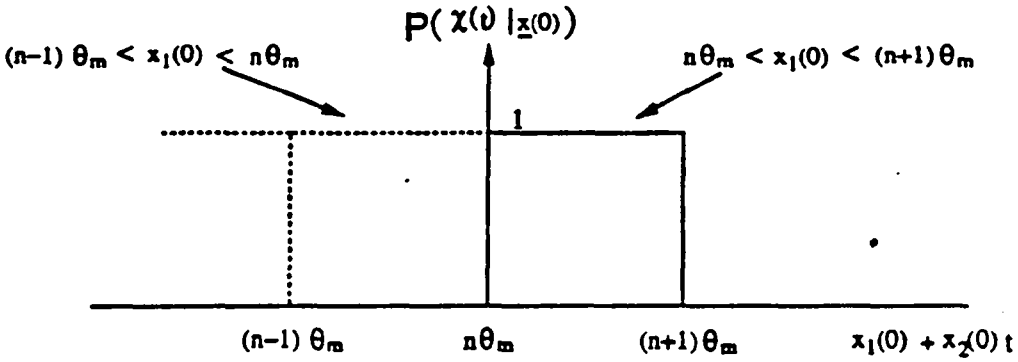


Figure 15 Conditional probability distribution.

$n\theta_m + \epsilon_m, x_1(0)$ may be above or below the level $n\theta_m + \epsilon_n$. The result is basically a blurring of the rectangular pulse, but the mean is the same and the variance not very different.

Remark: The standard deviation being $\theta_m/\sqrt{3}$, it follows that the Gaussian density will be relatively flat for values of $x_2(0)$ and $x_3(0)$ such that $|x_2(0)t| < \theta_m, |x_3(0)t^2| < \theta_m$. If t is small, this implies a flat distribution for relatively wide ranges of $x_2(0), x_3(0)$; that explains why this function will not really have an effect on $p(\mathbf{x}(0)|X(t)$ if t is small. Hence its usefulness at low velocities, where t is relatively large.

Now to calculate $p(\mathbf{x}(0)|X(t))$, according to Equation (66). The result is clearly Gaussian, so consideration may be limited to the exponent.

Let $p(\mathbf{x}(0)) = N(\hat{\mathbf{x}}, P)$ and $p(\mathbf{x}(0)|X(t)) = N(\mathbf{m}, M^{-1})$, with $\hat{\mathbf{x}}^T = [n\theta_m + \overline{x}_1 \;\; \hat{x}_2 \;\; \hat{x}_3]$ and $\mathbf{m}^T = [n\theta_m + \overline{m}_1 \;\; m_2 \;\; m_3]$, and let $S = P^{-1}$. Equating exponents on each side of Equation (66) yields

$$S_{11}(x_1 - n\theta_m - \overline{x}_1)^2 + 2S_{12}(x_1 - n\theta_m - \overline{x}_1)(x_2 - \hat{x}_2) + 2S_{13}(x_1 - n\theta_m - \overline{x}_1)(x_3 - \hat{x}_3) +$$

$$S_{22}(x_2 - \hat{x}_2)^2 + 2S_{23}(x_2 - \hat{x}_2)(x_3 - \hat{x}_3) + S_{33}(x_3 - \hat{x}_3)^2 +$$

$$\frac{(x_1 - n\theta_m)^2}{\sigma^2} + \frac{x_2^2 t^2}{\sigma^2} + \frac{x_3^2 t^4}{4\sigma^2} + \frac{2t(x_1 - n\theta_m)x_2}{\sigma^2} +$$

$$\frac{t^2(x_1 - n\theta_m)}{\sigma^2} x_3 + \frac{t^3 x_2 x_3}{\sigma^2} = M_{11}(x_1 - n\theta_m - \overline{m}_1)^2 +$$

$$2M_{12}(x_1 - n\theta_m - \overline{m}_1)(x_2 - m_2) + 2M_{13}(x_1 - n\theta_m - \overline{m}_1)(x_3 - m_3) +$$

$$M_{22}(x_2 - m_2)^2 + 2M_{23}(x_2 - m_2)(x_3 - m_3) + M_{33}(x_3 - m_3)^2.$$

Matching terms gives the following:

$$S_{11} + \frac{1}{\sigma^2} = M_{11}$$

$$S_{22} + \frac{t^2}{\sigma^2} = M_{22}$$

$$S_{33} + \frac{t^4}{\sigma^2} = M_{33}$$

$$S_{12} + \frac{t}{\sigma^2} = M_{12}$$

$$S_{13} + \frac{t^2}{2\sigma^2} = M_{13}$$
$$S_{23} + \frac{t^3}{2\sigma^2} = M_{23}$$

$$-S_{11}\overline{x}_1 - S_{12}\widehat{x}_2 - S_{13}\widehat{x}_3 = -M_{11}\overline{m}_1 - M_{12}m_2 - M_{13}M_3$$
$$-S_{12}\overline{x}_1 - S_{22}\widehat{x}_2 - S_{23}\widehat{x}_3 = -M_{12}\overline{m}_1 - M_{22}m_2 - M_{23}m_3$$
$$-S_{13}\widehat{x}_1 - S_{23}\widehat{x}_2 - S_{33}\widehat{x}_3 = -M_{13}\overline{m}_1 - M_{23}m_2 - M_{33}m_3$$

Those equations are written more compactly as

$$M = S + \frac{1}{\sigma^2}\begin{bmatrix} 1 \\ t \\ \frac{t^2}{2} \end{bmatrix}[1\ t\ t^2/2] \tag{68}$$

$$M\begin{bmatrix} \overline{m}_1 \\ m_2 \\ m3 \end{bmatrix} = S\begin{bmatrix} \overline{x}_1 \\ \widehat{x}_2 \\ \widehat{x}_3 \end{bmatrix} \tag{69}$$

Let $P_u = M^{-1}$, i.e. P_u is the updated P. Using the Matrix Inversion Lemma,

$$P_u = P - \alpha(t)P\mathbf{v}(t)\mathbf{v}^T(t)P \tag{70}$$

where

$$\alpha(t) = \frac{1}{\sigma^2 + \mathbf{v}^T(t)P\mathbf{v}(t)} \tag{71}$$
$$\mathbf{v}^T(t) = [\,1\ t\ \frac{1}{2}t^2] \tag{72}$$

Also,

$$\widehat{\mathbf{x}}_u = \begin{bmatrix} \overline{m}_1 \\ m_2 \\ m_3 \end{bmatrix} = [I - \alpha(t)P\mathbf{v}(t)\mathbf{v}^T(t)]\begin{bmatrix} \overline{x}_1 \\ \widehat{x}_2 \\ \widehat{x}_3 \end{bmatrix} \tag{73}$$

Here, m_2 and m_3 are the updated values of $\widehat{x}_2$ and $\widehat{x}_3$; the updated value of $\widehat{x}_1$ is $n\theta_m + \overline{m}_1$.

For the double integrator, the expressions are modified only to the extent that $\mathbf{v}^T(t) = [1\ t]$, and the third vector components in Equation (73) are removed.

Recall that P_u and $\hat{\mathbf{x}}_u$ are updated values at $t = 0$. Given those, the estimates at time t are

$$\hat{\mathbf{x}}(t) = A(t)\hat{\mathbf{x}}_u \tag{74}$$

$$P(t) = A(t)P_u A^T(t) + P_s(t) \tag{75}$$

where $P_s(t) = \mathrm{E}[\mathbf{x}_s(t)\mathbf{x}_s^T(t)]$ was given in the preceeding section, and is positive definite.

It is important to verify the asymptotic behavior of the estimates for large t, i.e. in the absence of encoder pulses. The following result addresses that question.

Theorem 4: In the absence of pulses, the estimate $\hat{x}_1$ tends to $n\theta_m$ and the estimates $\hat{x}_2$ and $\hat{x}_3$ to zero.

Proof: With no loss of generality, assume the estimates are carried out sequentially, with unity as the time step (normalized time). From Equations (74) and (75), using also Equations (70) and (73),

$$\hat{\mathbf{x}}(k+1) = A(1)[I - \alpha(k)P(k)\mathbf{v}(1)\mathbf{v}^T(1)]\hat{\mathbf{x}}(k) \tag{76}$$

$$P(k+1) = A(1)[P(k) - \alpha(k)P(k)\mathbf{v}(1)\mathbf{v}^T(1)P(k)]A^T(1) + P_s(1) \tag{77}$$

In the case of the triple integrator,

$$A(1) = \begin{bmatrix} 1 & 1 & \frac{1}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{v}(1) = \begin{bmatrix} 1 \\ 1 \\ \frac{1}{2} \end{bmatrix}$$

whereas , for the double integrator,

$$A(1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \mathbf{v}(1) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

From Equation (71),

$$\alpha(k) = \frac{1}{\sigma^2 + \mathbf{v}^T(1)P(k)\mathbf{v}(1)}. \tag{78}$$

Taking Equations (77) and (78) together, it is seen that Equation (77) is the discrete Riccati equation for a system where the A, C, R and Q matrices (using conventional notation)

are, respectively, $A(1), \mathbf{v}^T(1), \sigma^2$ and $P_s(1)$. The product $\alpha(k)P(k)\mathbf{v}(1)$ in Equation (76) is the corresponding Kalman gain, $K(k)$. In the usual notation of Kalman filtering, the RHS matrix in Equation (76) is $A[I - K(k)C]$. Now, it is easily verified that $(\mathbf{v}^T(1), A(1))$ is observable. Therefore, $A[I-K(k)C]$, actually the A-matrix of the error system, is a stability matrix, so that $\widehat{\mathbf{x}}(k)$ tends to zero. The result follows.

3.6 Summary of the Algorithm

The following assumes time-normalized variables.

For the triple integrator (estimation of velocity and acceleration):

Data: $\mathbf{x}(iT), P(iT)$, Pulse times $iT + t_k$, levels $y(t_k)$.

Step 1: Initialize $\mathbf{x}_0 = \mathbf{x}(iT), P_0 = P(iT)$

Step 2: If $N \leq N_{\text{low}}$, go to Step 8.

Step 3:

$$\mathbf{x} = \begin{bmatrix} \frac{1}{N}\Sigma y(t_k) \\ \frac{1}{N}\Sigma t_k y(t_k) \\ \frac{1}{2N}\Sigma t_k^2 y(t_k) \end{bmatrix} + \frac{R}{N} P_0^{-1}\mathbf{x}_0$$

when the summations are from $k = 1$ to N.

Step 4:

$$M = \begin{bmatrix} 1 & \frac{1}{N}\Sigma t_k & \frac{1}{2N}\Sigma t_k^2 \\ \frac{1}{N}\Sigma t_k & \frac{1}{N}\Sigma t_k^2 & \frac{1}{2N}\Sigma t_k^3 \\ \frac{1}{2N}\Sigma t_k^2 & \frac{1}{2N}\Sigma t_k^3 & \frac{1}{4N}\Sigma t_k^4 \end{bmatrix} + \frac{R}{N} P_0^{-1}$$

Step 5:

$$\widehat{\mathbf{x}}(1) = AM^{-1}\mathbf{x}.$$

$$A = \begin{bmatrix} 1 & 1 & \frac{1}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Step 6: Covariance update (See Tables B1, B2 for α^{ij}, β^{ij})

$$\{P_1\}_{ji} = \{P_1\}_{ij} = q\frac{\alpha_0^{ij} + \alpha_1^{ij} N + \cdots + \alpha_{4+i+j}^{ij} N^{4+i+j}}{N^{4+i+j}}$$

$$\{P_2\}_{ij} = q\frac{\beta_0^{ij} + \beta_1^{ij} N + \cdots + \beta_{n(i,j)}^{ij} N^{n(i,j)}}{N^{n(i,j)}}$$

$$P_3 = q\begin{bmatrix} 1/20 & 1/8 & 1/6 \\ 1/8 & 1/3 & 1/2 \\ 1/6 & 1/2 & 1 \end{bmatrix}$$

$$C = A\left[\frac{R}{N}M^{-1} + M^{-1}P_1M^{-1}\right]A^T + P_3 + AM^{-1}P_2 + P_2^TM^{-1}A^T$$

Step 7:

$$\mathbf{x}((i+1)T) = \mathbf{x}(1)$$

$$P((i+1)T) = C$$

Go To 1.

Step 8: Initialize: $i = 0, t_0 = 0, t_{N+1} = 1, \sigma^2 = \frac{1}{3}\theta_m^2, S = P_0^{-1}$.

Step 9:

$$\overline{x}_0 = x_{01} - n\theta_m$$

$$\delta t = t_{i+1} - t_i$$

$$\mathbf{v}^T = [1 \ \delta t \ \frac{1}{2}\ \delta t^2]$$

$$\alpha = \frac{1}{\sigma^2 + \mathbf{v}^T P_0 \mathbf{v}}$$

$$\begin{bmatrix} \overline{m}_1 \\ m_2 \\ m_3 \end{bmatrix} = [I \ - \alpha \ P_0 \ \mathbf{v}\mathbf{v}^T]\begin{bmatrix} \overline{x}_{10} \\ x_{20} \\ x_{30} \end{bmatrix}$$

Step 10:

$$P_0 = P_0 - \alpha P_0 \mathbf{v}\mathbf{v}^T P_0$$

$$P_0^{-1} = S = S + \frac{1}{\sigma^2}\mathbf{v}\mathbf{v}^T$$

Step 11:

$$\mathbf{x}_0 = \begin{bmatrix} \overline{m}_1 \\ m_2 \\ m_3 \end{bmatrix} + \begin{bmatrix} n\theta_m \\ 0 \\ 0 \end{bmatrix}.$$

IF $i = N$, GO TO 16.

Step 12:

$$M = RP_0^{-1} + \mathbf{v}\mathbf{v}^T$$

$$M^{-1} = \frac{1}{R}P_0 - \frac{1}{R}\frac{P_0\mathbf{v}\mathbf{v}^T P_0}{R + \mathbf{v}^T P_0 \mathbf{v}}$$

Step 13:

$$M^{-1}\mathbf{x} = \left[I - \frac{P_0 \mathbf{v}\mathbf{v}^T}{R+\mathbf{v}^T P_0 \mathbf{v}}\right][\mathbf{x}_0 + \tfrac{1}{R} P_0 \mathbf{v} y(t_{i+1})]$$

$$\hat{\mathbf{x}}(t_{i+1}) = \begin{bmatrix} 1 & \delta t & \frac{1}{2}\delta t^2 \\ 0 & 1 & \delta t \\ 0 & 0 & 1 \end{bmatrix} M^{-1}\mathbf{x}$$

Step 14: Covariance update

$$P_1 = q\frac{\delta t^5}{20}\begin{bmatrix} 1 & \delta t & \frac{1}{2}\delta t^2 \\ \delta t & \delta t^2 & \frac{1}{2}\delta t^3 \\ \frac{1}{2}\delta t^2 & \frac{1}{2}\delta t^3 & \frac{1}{4}\delta t^4 \end{bmatrix}$$

$$P_2 = q\begin{bmatrix} \frac{\delta t^5}{20} & \frac{\delta t^4}{8} & \frac{\delta t^3}{6} \\ \frac{\delta t^6}{20} & \frac{\delta t^5}{8} & \frac{\delta t^4}{6} \\ \frac{\delta t^7}{40} & \frac{\delta t^6}{16} & \frac{\delta t^5}{12} \end{bmatrix}$$

$$P_3 = q\begin{bmatrix} \frac{\delta t^5}{20} & \frac{\delta t^4}{8} & \frac{\delta t^3}{6} \\ \frac{\delta t^4}{8} & \frac{\delta t^3}{3} & \frac{\delta t^2}{2} \\ \frac{\delta t^3}{6} & \frac{\delta t^2}{2} & \delta t \end{bmatrix}$$

$$C = A[RM^{-1} + M^{-1}P_1 M^{-1}]A^T + P_3 + AM^{-1}P_2 + P_2^T M^{-1} A^T$$

with

$$A = \begin{bmatrix} 1 & \delta t & \frac{1}{2}\delta t^2 \\ 0 & 1 & \delta t \\ 0 & 0 & 1 \end{bmatrix}.$$

Step 15:

$$i = i+1, P_0 = C, \mathbf{x}_0 = \hat{\mathbf{x}}(t_{i+1})$$

GO TO 9.

Step 16:

$$\delta t = t_{N+1} - t_N = 1 - t_N$$

$$\hat{\mathbf{x}}(t_{N+1}) = \begin{bmatrix} 1 & \delta t & \frac{1}{2}\delta t^2 \\ 0 & 1 & \delta t \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_0$$

Step 17:

$$P_3 = q\begin{bmatrix} \delta t^5/20 & \delta t^4/8 & \delta t^3/6 \\ \delta t^4/8 & \delta t^3/3 & \delta t^2/2 \\ \delta t^3/6 & \delta t^2/2 & \delta t \end{bmatrix}$$

$$C = AP_0 A^T + P_3$$

where

$$A = \begin{bmatrix} 1 & \delta t & \frac{1}{2}\delta t^2 \\ 0 & 1 & \delta t \\ 0 & 0 & 1 \end{bmatrix}$$

Step 18:

$$\mathbf{x}((i+1)T) = \hat{\mathbf{x}}(t_{N+1}), P((i+1)T) = C$$

GO TO 1

The algorithm for the double integrator is the same, except that the third element of every vector is dropped and only the top left 2×2 corner of every matrix is retained. Of course, the coefficients α^{ij} and β^{ij} are to be found in Tables B3, B4. The matrices in Step 14 are

$$P_1 = q\frac{\delta t^3}{3}\begin{bmatrix} 1 & \delta t \\ \delta t & \delta t^2 \end{bmatrix}$$

$$P_2 = q\begin{bmatrix} \frac{1}{3}\delta t^3 & \frac{1}{2}\delta t^2 \\ \frac{1}{3}\delta t^4 & \frac{1}{2}\delta t^3 \end{bmatrix}$$

$$P_3 = q\begin{bmatrix} \frac{1}{3}\delta t^3 & \frac{1}{2}\delta t^2 \\ \frac{1}{2}\delta t^2 & \delta t \end{bmatrix}$$

and, in Step 6,

$$P_3 = q\begin{bmatrix} 1/3 & 1/2 \\ 1/2 & 1 \end{bmatrix}$$

Because the matrices are either 3×3 or 2×2, there is very little effort involved in matrix inversion or vector-matrix multiplication. Most of the computational work is in the summations. In steps 3 and 4, there are 5 multiplications for each k and 7 summations, i.e. $5N$ multiplications and $7N$ additions. For the triple integrator, there are about 250 additional multiplications. For example, with $N = 100(30\,\text{deg}/s$ with $T = 0.01s$ and $\theta_m = 0.003$ deg) there are about 750 multiplications and 750 additions. These must be performed in a small fraction of the sampling time, if the estimates are to be supplied to the controller with essentially no delay. In practice, it may be necessary to perform the estimation and control calculations during the following sampling period, at the price of a delay of one sampling period.

For the double integrator model, there are 3N multiplications and and 5N additions, plus about 100 extra multiplications.

3.7 Additional Features

The first modification concerns the use of a model. If the triple-integrator model is used that rests on the availability of an estimate of the third derivative $\dddot{\theta}$. This must be time normalized, by multiplying by T^3. The portion of the angular output due to this know input is obtained by solving

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3$$

$$\dot{x}_3 = \dddot{\theta}_{\text{mod}}$$

with $\dddot{\theta}_{\text{mod}}$ constant during the sampling period and $\mathbf{x}(0) = 0$. The solution is

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} 1/6t^3 \\ 1/2t^2 \\ t \end{bmatrix} \dddot{\theta}_{\text{mod}}.$$

This leads to the following modifications in the algorithm.

Step 1A Replace $y(t_k)$ by $y(t_k) - (1/6)t_k^3 \dddot{\theta}_{\text{mod}}(iT)$.

Step 5

$$\widehat{\mathbf{x}}(1) = AM^{-1}\mathbf{x} + \begin{bmatrix} 1/6 \\ 1/2 \\ 1 \end{bmatrix} \dddot{\theta}_{\text{mod}}(iT).$$

Step 9

$$\begin{bmatrix} \overline{m}_1 \\ m_2 \\ m_3 \end{bmatrix} = [I - \alpha P_0 \mathbf{v}\mathbf{v}^T]\left\{ \begin{bmatrix} \overline{x}_{10} \\ x_{20} \\ x_{30} \end{bmatrix} - \dddot{\theta}_{\text{mod}}(iT) P_0 \begin{bmatrix} \frac{1}{3}\delta t^3 \\ \frac{1}{3}\delta t^4 \\ \frac{1}{6}\delta t^5 \end{bmatrix} \right\}$$

Step 13 Replace $y(t_{i+1})$ by $y(t_{i+1}) - \frac{1}{6}\delta t^3 \dddot{\theta}_{\text{mod}}(iT)$

$$\widehat{\mathbf{x}}(t_{i+1}) = \begin{bmatrix} 1 & \delta t & 1/2\delta t^2 \\ 0 & 1 & \delta t \\ 0 & 0 & 1 \end{bmatrix} M^{-1}\mathbf{x} + \begin{bmatrix} 1/6\delta t^3 \\ 1/2\delta t^2 \\ \delta t \end{bmatrix} \dddot{\theta}_{\text{mod}}(iT).$$

Step 16

$$\widehat{\mathbf{x}}(t_{N+1}) = \begin{bmatrix} 1 & \delta t & \frac{1}{2}\delta t^2 \\ 0 & 1 & \delta t \\ 0 & 0 & 1 \end{bmatrix} n\mathbf{x}_0 + \begin{bmatrix} \frac{1}{6}\delta t^3 \\ \frac{1}{2}\delta t^2 \\ \delta t \end{bmatrix} \dddot{\theta}_{\text{mod}}(iT)$$

In the case of the double integrator, the modifications are:

Step 1A Replace $y(t_k)$ by $y(t_k) - \frac{1}{2}t_k^2 \ddot{\theta}_{\text{mod}}(iT)$.

Step 5

$$\hat{\mathbf{x}}(1) = AM^{-1}\mathbf{x} + \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix} \ddot{\theta}_{\text{mod}}(iT).$$

Step 9

$$\begin{bmatrix} \overline{m}_1 \\ m_2 \end{bmatrix} = [I - P_0 \mathbf{v}\mathbf{v}^T]\left\{ \begin{bmatrix} \overline{x}_{10} \\ x_2 \end{bmatrix} - \ddot{\theta}_{\text{mod}}(iT) P_0 \begin{bmatrix} \frac{1}{2}\delta t^2 \\ \frac{1}{2}\delta t^3 \end{bmatrix} \right\}.$$

Step 13 The second expression becomes

$$\hat{\mathbf{x}}(t_{i+1}) = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} M^{-1}\mathbf{x} + \begin{bmatrix} \frac{1}{2}\delta t^2 \\ \delta t \end{bmatrix} \ddot{\theta}_{\text{mod}}(iT)$$

Step 16

$$\hat{\mathbf{x}}(t_{N+1}) = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_0 + \begin{bmatrix} \frac{1}{2}\delta t^2 \\ \delta t \end{bmatrix} \ddot{\theta}_{\text{mod}}(iT)$$

Whether a model is useful or not depends crucially on the quality of the approximation of the third (or second) derivative. Given the relatively heavy high-frequency contact of the third derivative, in particular, it is often difficult to model that signal. As a rule of thumb, the error $\dddot{\theta} - \dddot{\theta}_{\text{mod}}$ must be less in magnitude than $|\dddot{\theta}|$.

Another way of taking into account the presence of the driving signal is to adapt by varying the plant noise variance q. Since $\dot{x}_3 = w$, the variance of $x_3(1)$, given that $x_3(0) = 0$, is q (time normalized, of course). The constant third-derivative signal leads to $x_3(1) = \dddot{\theta}$; it is thus reasonable to use

$$q = c|\dddot{\theta}_{\text{mod}}|^2 \tag{79}$$

where c is a constant.

The advantage of using this technique rather than using $\dddot{\theta}_{\text{mod}}$ as an input is that i) it depends on the magnitude of $\dddot{\theta}_{\text{mod}}$ only and ii) the effect of the variable q, which is to "open" the Kalman filter, is dynamic because it changes the time constant gradually through changes in the error covariance matrix. It therefore acts on a more-or-less short-term average of $\dddot{\theta}^2_{\text{mod}}$, which will usually be fairly close to the same average of $\dddot{\theta}$.

The algorithm is modified as follows:

Step 5A $\quad q = \mathrm{d}[\ddot{\theta}_{\mathrm{mod}}(iT)]^2$

Step 13A $\quad q = \mathrm{d}[\ddot{\theta}_{\mathrm{mod}}(iT)]^2.$

For the double integrator, $\ddot{\theta}_{\mathrm{mod}}$ is, of course, replaced by $\dot{\theta}_{\mathrm{mod}}$.

3.8 Effect of Time Measurement Errors

Because the algorithm depends on the pulse arrival times, it is important to examine the effects of errors in the measurement of those arrival times. The simplest way to approach this is to borrow an idea from [9]: the time measurement is assumed to be correct, and the time error is translated into an angle measurement error.

More precisely, let δt_i be the error in the measurement of the pulse arrival time t_i. The angle output at $t_i + \delta t_i$ is given approximately by

$$y(t_i + \delta t_i) \approx y(t_i) + v(t_i)\delta t_i \tag{80}$$

where $v(t_i)$ is the angular velocity at t_i. This suggests that $t_i + \delta t_i$ be considered as an exact time measurement, accompanied by an angle measurement error of $v(t_i)\delta t_i$.

If the δt_i form a sequence of independent random variables of variance σ_t^2, the error variance induced in the output is

$$\text{Variance due to time error } = v^2(t_i)\sigma_t^2. \tag{81}$$

Since this error is independent of the error due to encoder tolerance, the variance in Equation (81) is added to ϵ_m^2 /6; thus,

$$\text{Angle error variance } = r = v^2(t_i)\sigma_t^2 + \epsilon_m^2 /6. \tag{82}$$

This variance can be computed at the start of each sampling interval, using the velocity estimate as surrogate for $v(t_i)$. In many cases, however, that may not be necessary, if

$$\sigma_t \ll \frac{\epsilon_m}{\sqrt{6}|v(t_i)|_{\max}} \tag{83}$$

where $|v(t_i)|_{max}$ is the maximum absolute angular velocity. For example, with $\epsilon_m = 0.00075$ deg. and $|v(t_i)|_{max} = 20$ deg/s, the RHS of Equation (83) is 15.3 μs: a time measurement with a standard deviation of about 1.5 μs would justify neglecting the velocity. If not, incorporating the variable r in the algorithm is easily done.

3.9 Simulations

All simulations examples are run with $\theta_m = 0.003$ deg and a sampling time of 0.01s. The measurement noise is the mechanical error in the quantization level, and is represented by a triangular distribution with limits $\pm\ \epsilon_m$; this leads to a variance of ϵ_m^2 /6. The integer N_{low} is set to 5: with fewer than 6 pulses in the sampling interval, the algorithm uses the low-velocity modification. The initial estimates are equal to 1, and the initial error covariance matrix is 4I. Finally, the time functions are the same as in previous examples.

Example 12

Here, $A = 10$ (high-speed trajectory) and $\epsilon_m = 0.00075$ deg. The angle estimation errors are shown in Figure 16a). The encoder output is dotted, the estimation error for $q = 20$ is the lighter of the two curves. Figures 16b) and c) show, respectively, the angular velocity and acceleration estimation errors (the finite difference acceleration errors are not shown because they are too large to fit on the graph). In addition, Figure 17 presents the true acceleration and its estimate: this is not given for the angle and angular velocity because the two graphs could not be distinguished. Figure 18 shows error data for two more values of q (the dotted curve corresponds to $q = 10000$)

Mean and standard deviation figures are given in Table III. It is apparent that, as q increases, the transients that occur at rapid acceleration changes are damped out, at the expense of a noisier signal elsewhere. The standard deviation figures are better than those of Table I for the fixed-time Kalman filter by a factor of about 3. It should be noted that the mean and standard deviation figures are dominated by the transients; comparison of

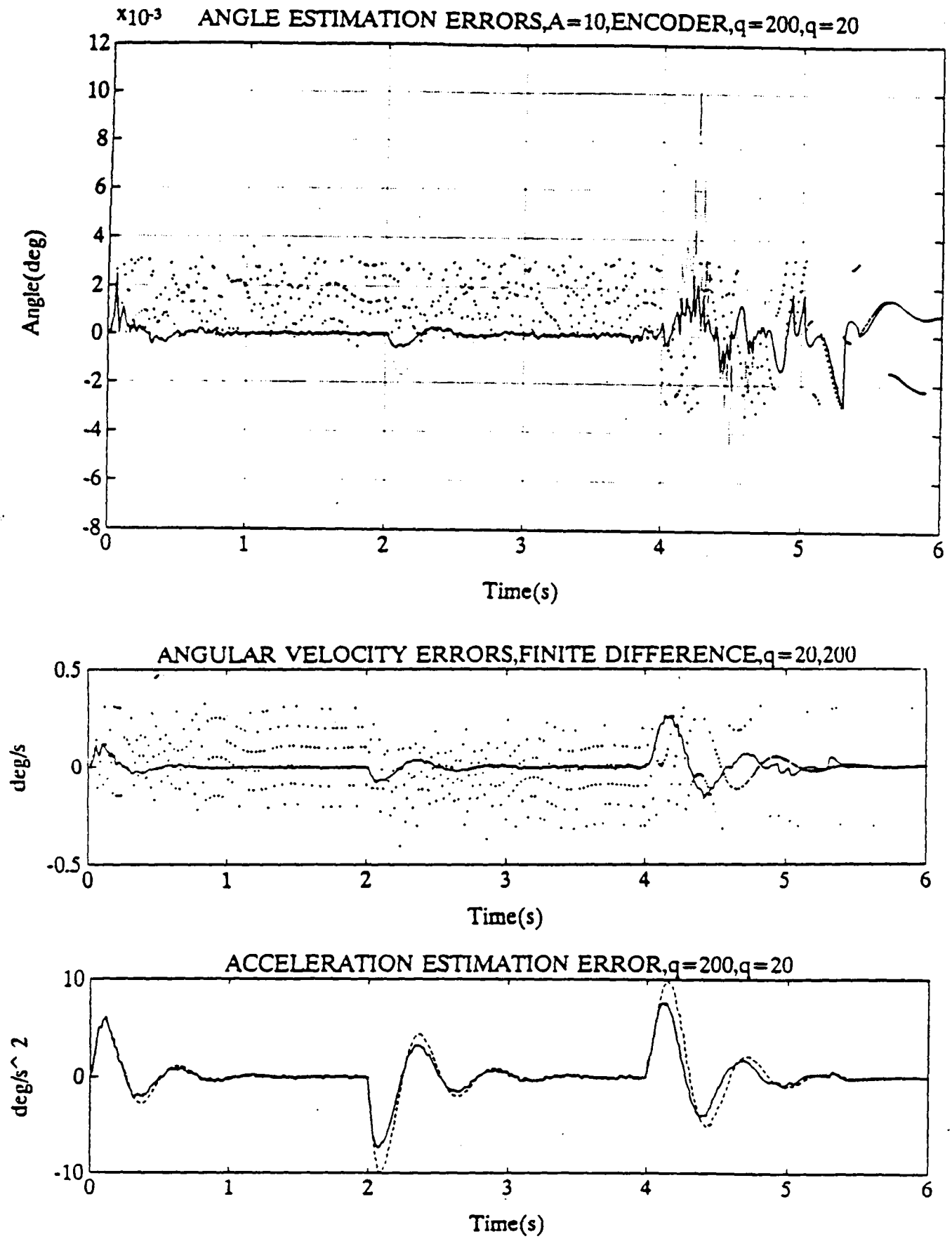


Figure 16 Fast trajectory (A = 10), new algorithm, estimation error for angle (dots are encoder error), velocity (dots are finite difference), acceleration, triple integrator, q = 20 (light curve) and 200 (heavy).

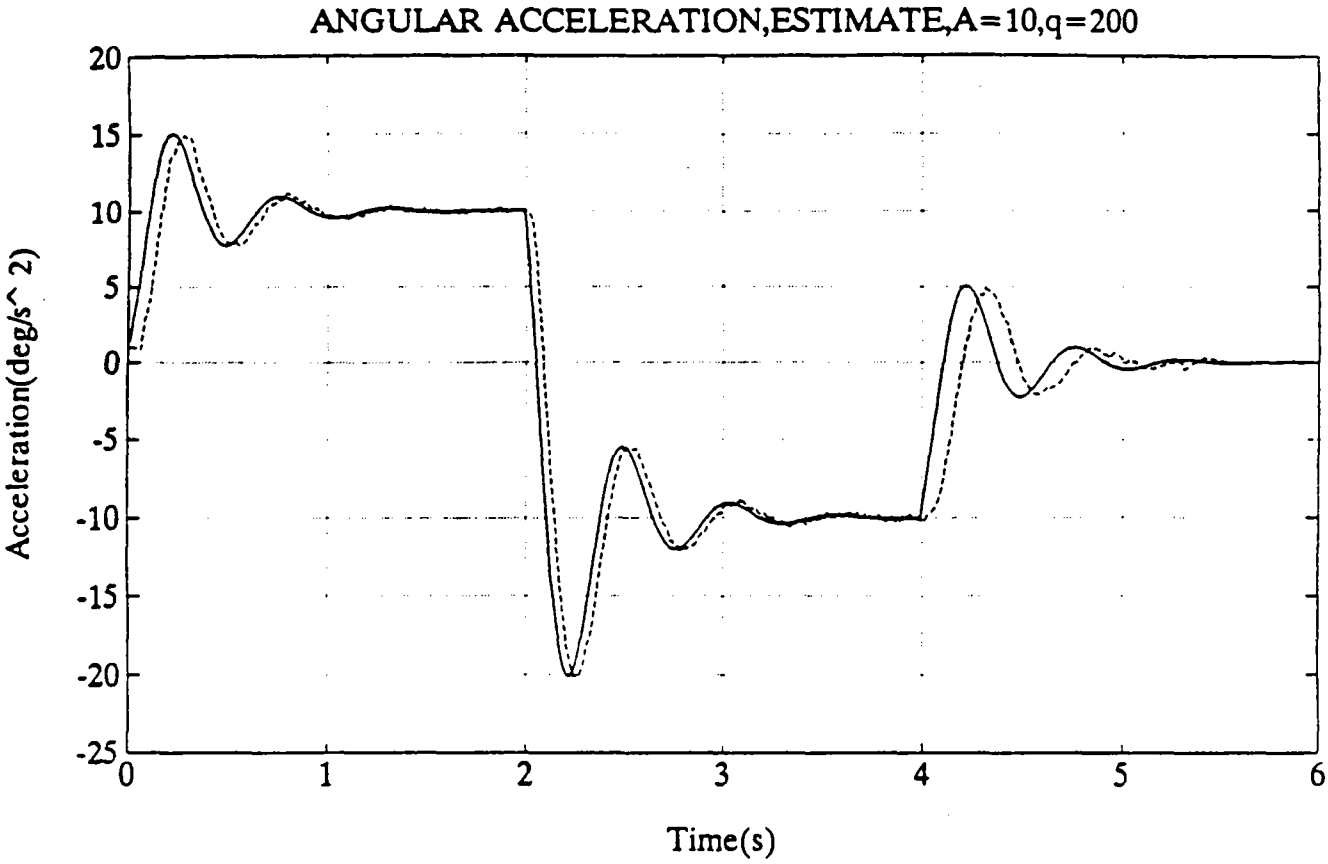


Figure 17 Fast trajectory, new algorithm, actual and estimated acceleration, q = 200.

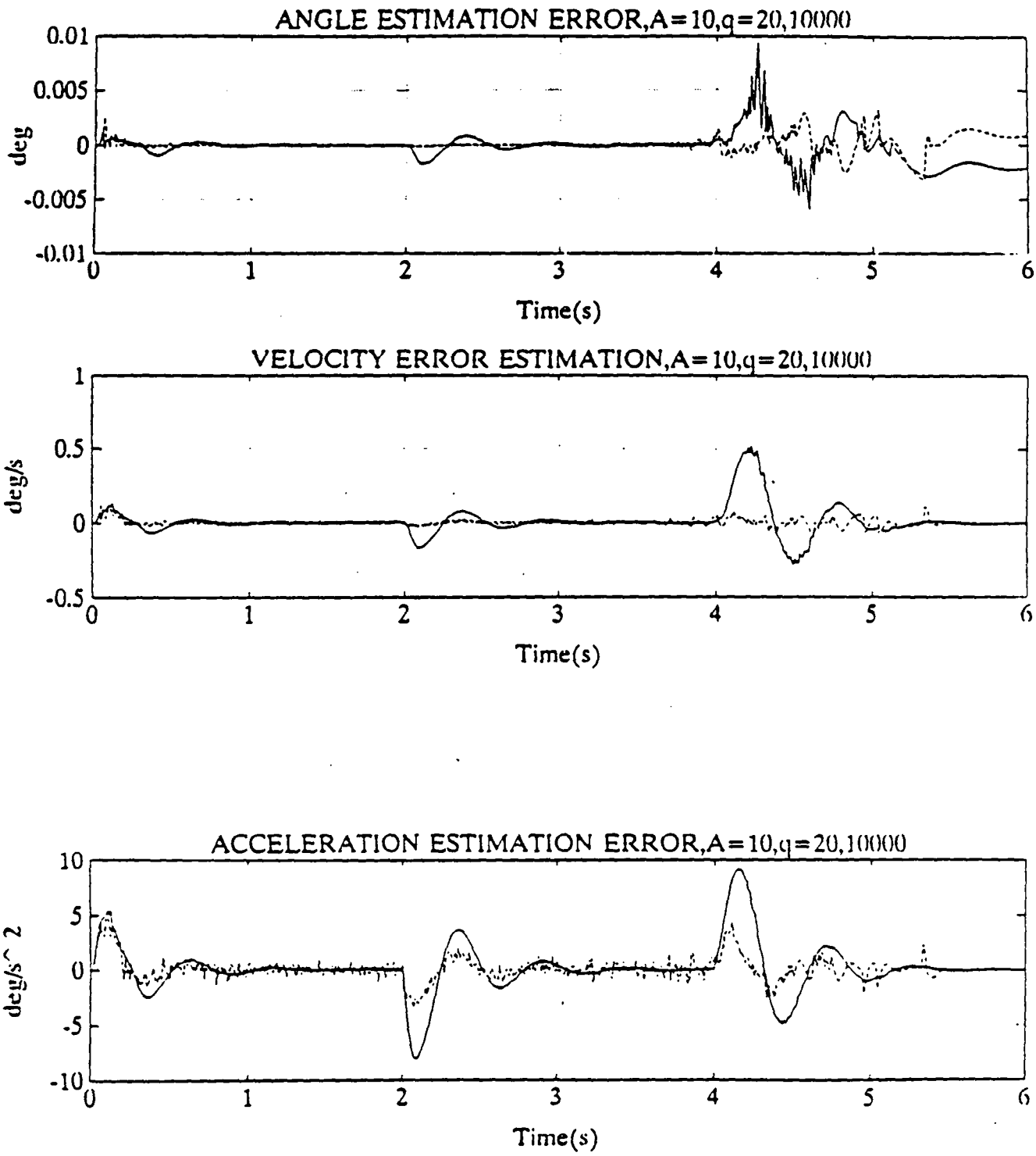


Figure 18 Fast trajectory, new algorithm, estimation errors for angle, velocity and acceleration, triple integrator, $q = 20$ (full curve) and 200 (dotted).

Figures 16 and 18 with Figure 8 shows that the pulse-time method yields estimates that are generally less noisy than the Kalman filter estimates.

Example 13

This example is as Example 12, but with $A = 1$ (slow trajectory). Results are presented in Figures 19, 20 ($q = 20$) and 21 ($q = 200$), and in Table IV. The figures should be compared to Figure 14, for the fixed-interval Kalman filter. The mean and standard deviation figures are better than those of Table II by a factor of 2 for the acceleration errors, but not as good for the velocities.. As was the case in Example 12, the estimation errors from the pulse-time method are generally smaller, especially as the velocity tends to zero (t approaching 6). Here again the mean and standard deviation result are dominated by the transients near $t = 2$ and $4s$, where the acceleration changes rather abruptly.

It is useful to compare the results for the fast and slow trajectories. Two countervailing effects are at work: the larger effective q for the trajectory with $A = 10$ tends to increase the estimation errors, but the higher velocity results in a larger number of samples. Theorem 1 is used to approach the situation. For the triple integrator plant, the error covariance matrix for the continuous-time Kalman filter has the following diagonal elements:

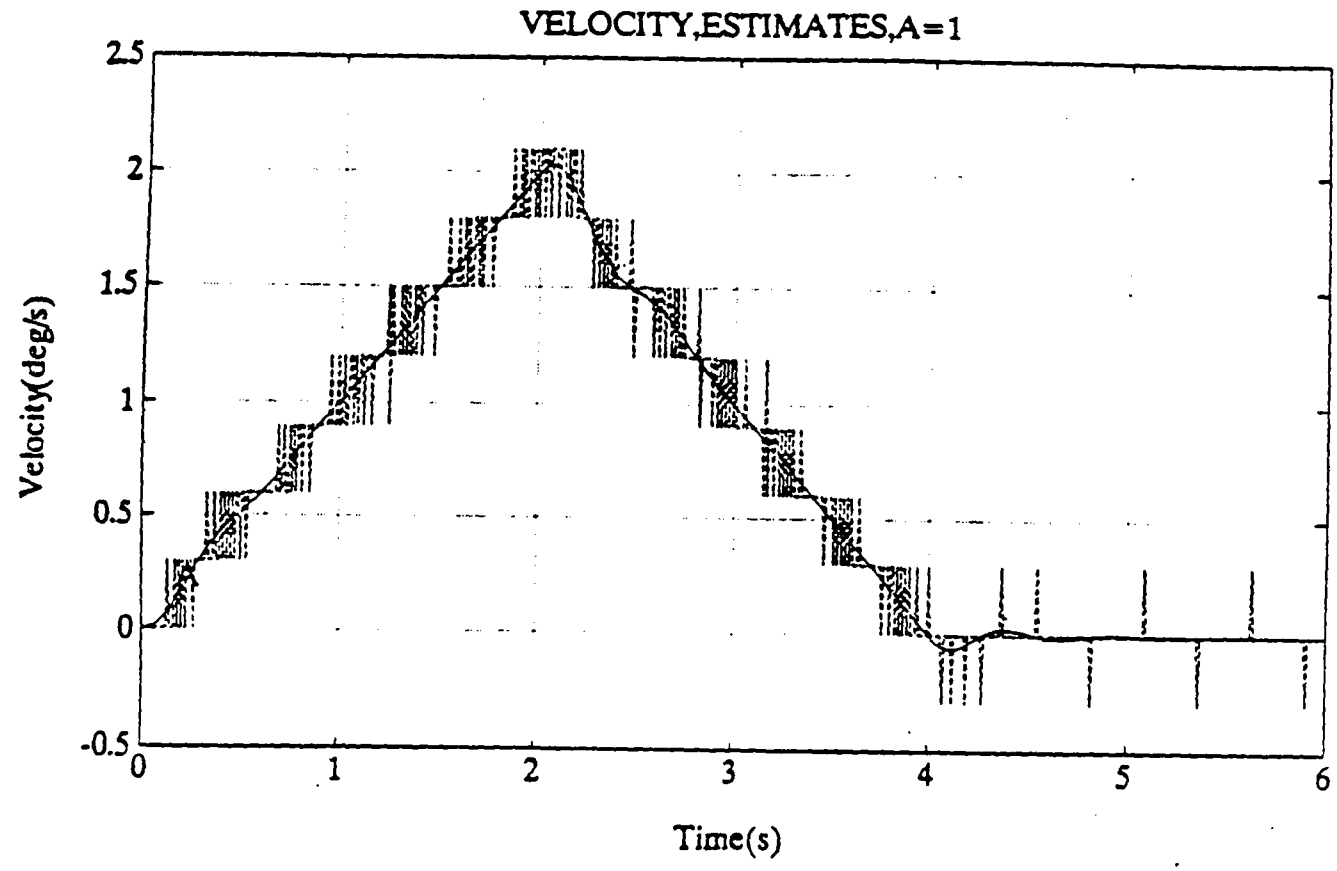
$$P_{11} = 2r(q/r)^{1/6}; P_{22} = 3r(q/r)^{1/2}; P_{33} = 2r(q/r)^{5/6}.$$

From Theorem 1, the covariance matrix of the discrete-time filter is related to that of the continuous-time filter through the expression

$$P_{ij}\ (\text{Discrete}) = P_{ij}\ (\text{Continuous})\ T^{\frac{2n-i-j+1}{2n}}$$

where T is the sampling period and n is the order.

It is not possible to assign a precise value of q to either trajectory, but one can assume a ratio of 10 to 100 between the two. The interpulse times are variable, but the higher velocity trajectory will on average result in interpulse times 1/10 of those of the lower velocity



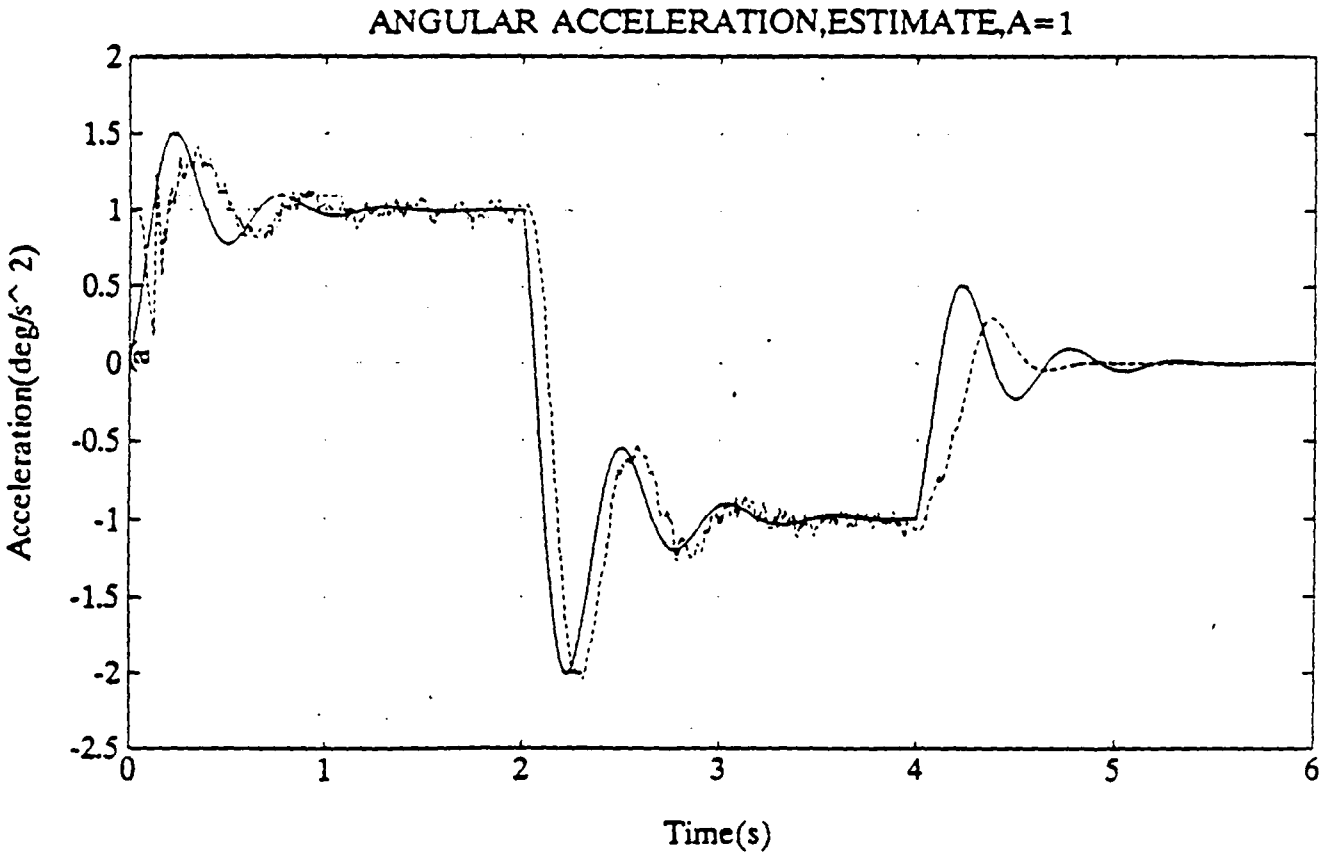


Figure 19 Slow trajectory ($A = 1$), new algorithm, true (heavy curve) and estimated (dotted) velocity and acceleration, triple integrator, $q = 20$. The finite-difference estimation is also shown for the velocity.

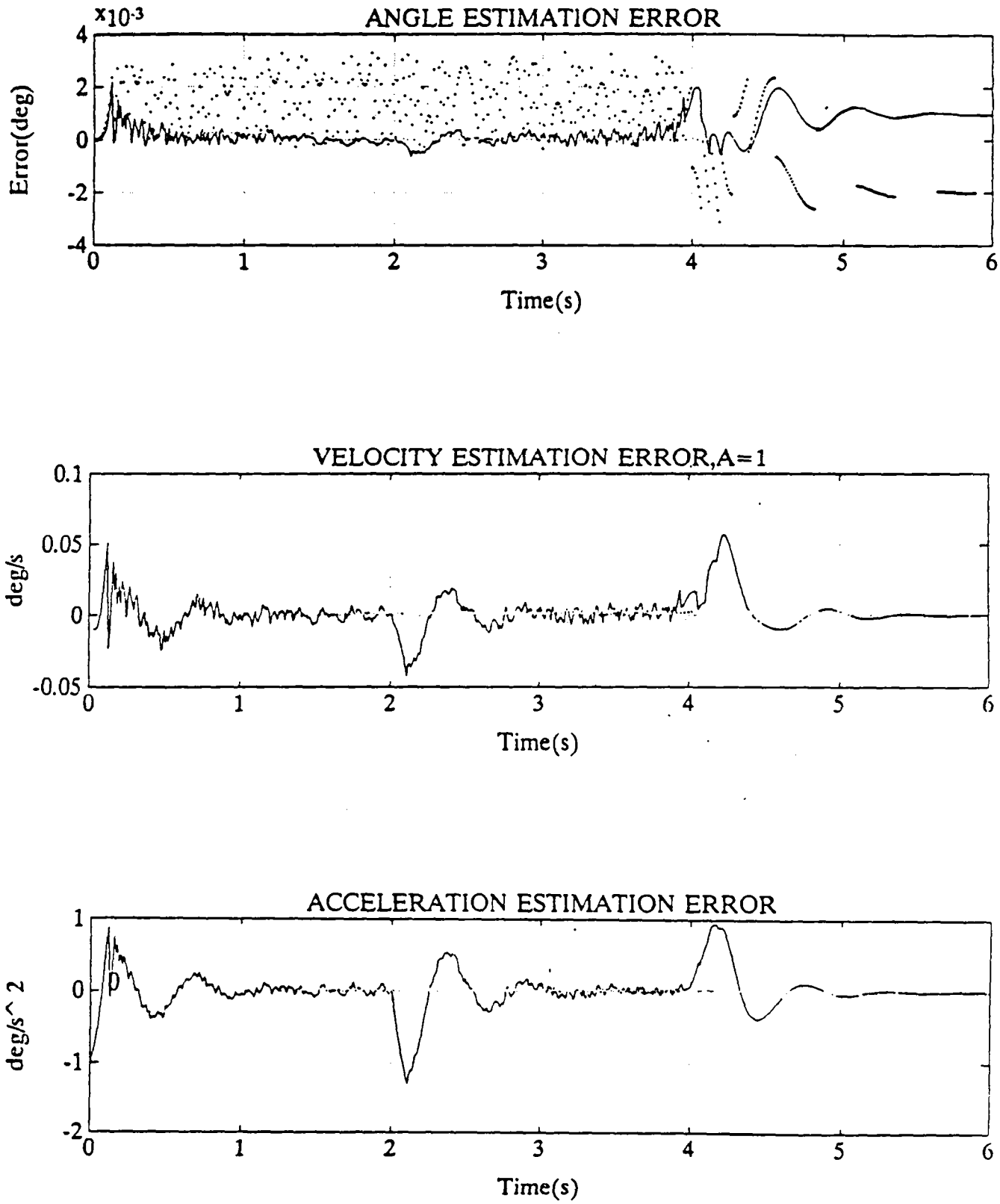


Figure 20 Slow trajectory, new algorithm, estimation errors for angle (dots are encoder errors), velocity and estimation, triple integrator, $q = 20$.

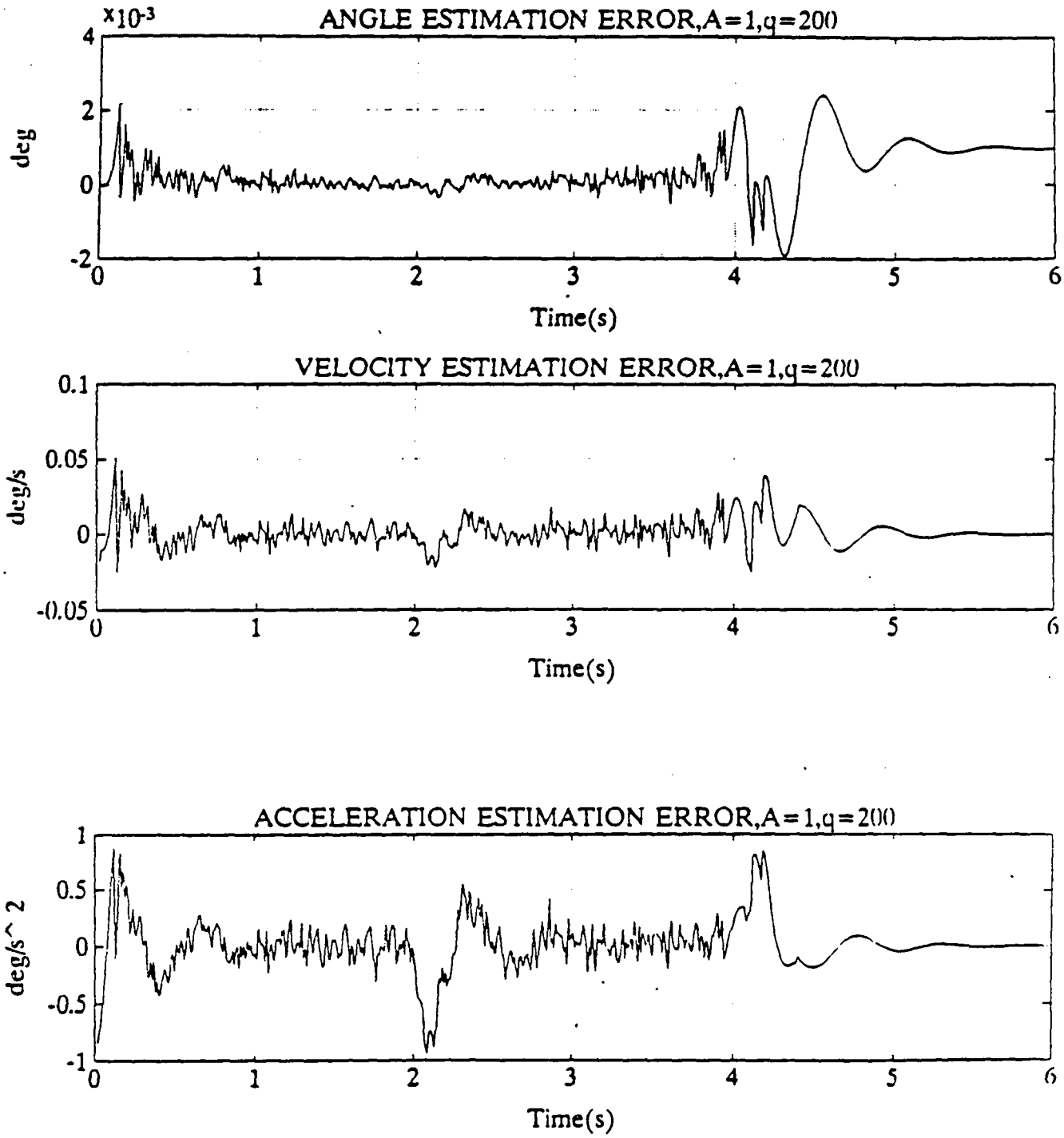


Figure 21 Slow trajectory, new algorithm, estimation errors for angle, velocity and acceleration, triple integrator, $q = 200$.

trajectory. Thus, the covariance elements for $A = 10$ and $A = 1$ can be expected to satisfy the following ratios (value for $A = 10$/value for $A = 1$):

$$P_{11} : (10 \text{ to } 100)^{1/6}(.1)^{5/6} [\text{Std: } .46 \text{ to } .56]$$

$$P_{22} : (10 \text{ to } 100)^{1/2}(.1)^{1/2} [\text{Std: } .1 \text{ to } 1.8]$$

$$P_{33} : (10 \text{ to } 100)^{5/6}(.1)^{1/6} [\text{Std: } 2.1 \text{ to } 5.6]$$

Comparing data for $A = 10, q = 200$ and $A = 1, q = 20$ in Table III shows rough agreement, except perhaps for the error in the angle estimate.

Example 14

Here a double integrator model is used, and the acceleration is not estimated. Table III summarized the performance figures, for a few values of q, for $A = 10$. Figure 22 shows the angle and velocity estimation errors, for $q = 20$. A major contribution to the standard deviation comes from the transients following rapid changes in acceleration.

The graphs of Figure 22 should be compared to those of Figure 16, where the triple integrator model was used. The angle estimation errors are similar, and the velocity estimation errors appear to be a bit larger, and more irregular, than in Figure 16, even though the standard deviations are comparable.

Example 15

This example is the same as the previous one, except for the fact that $A = 1$ (slow trajectory). Results are presented in Table IV and in Figure 23. The graphs should be compared with those of Figures 20 and 21. In this case, the velocity error is noticeably worse than for the case of the triple integrator, contrary to the situation in Example 14. It appears that for a slower trajectory, with less information (fewer pulses), the more complex model is beneficial.

Example 16

This, and the next few runs, adapts the value of q according to $q(t) = k\omega_d^2(t)$, where $\omega_d(t)$ is the derivative of the acceleration for the (incorrect) model where $J = 2$ rather than

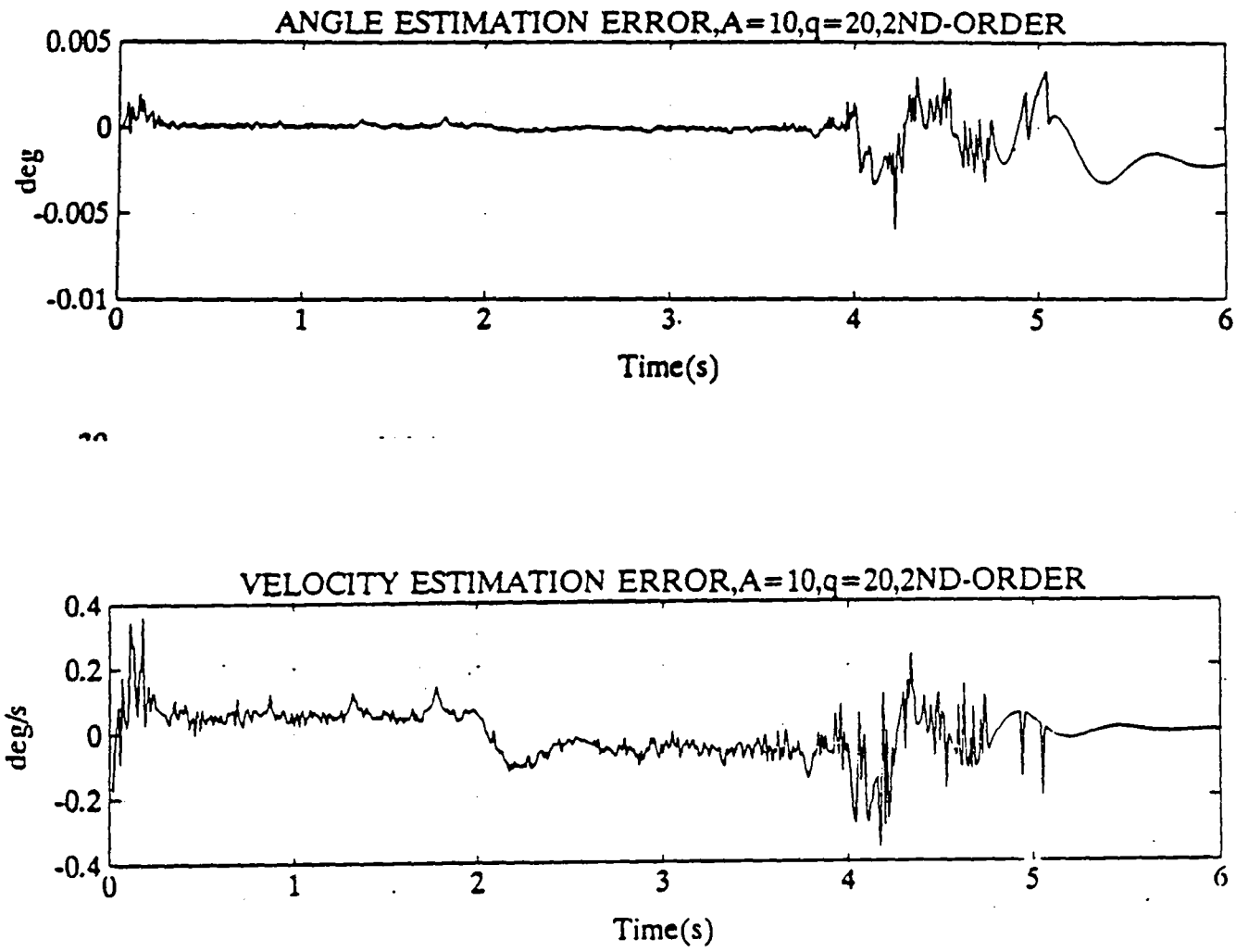


Figure 22 Fast trajectory, new algorithm, estimation errors for angle and velocity, double integrator, $q = 20$.

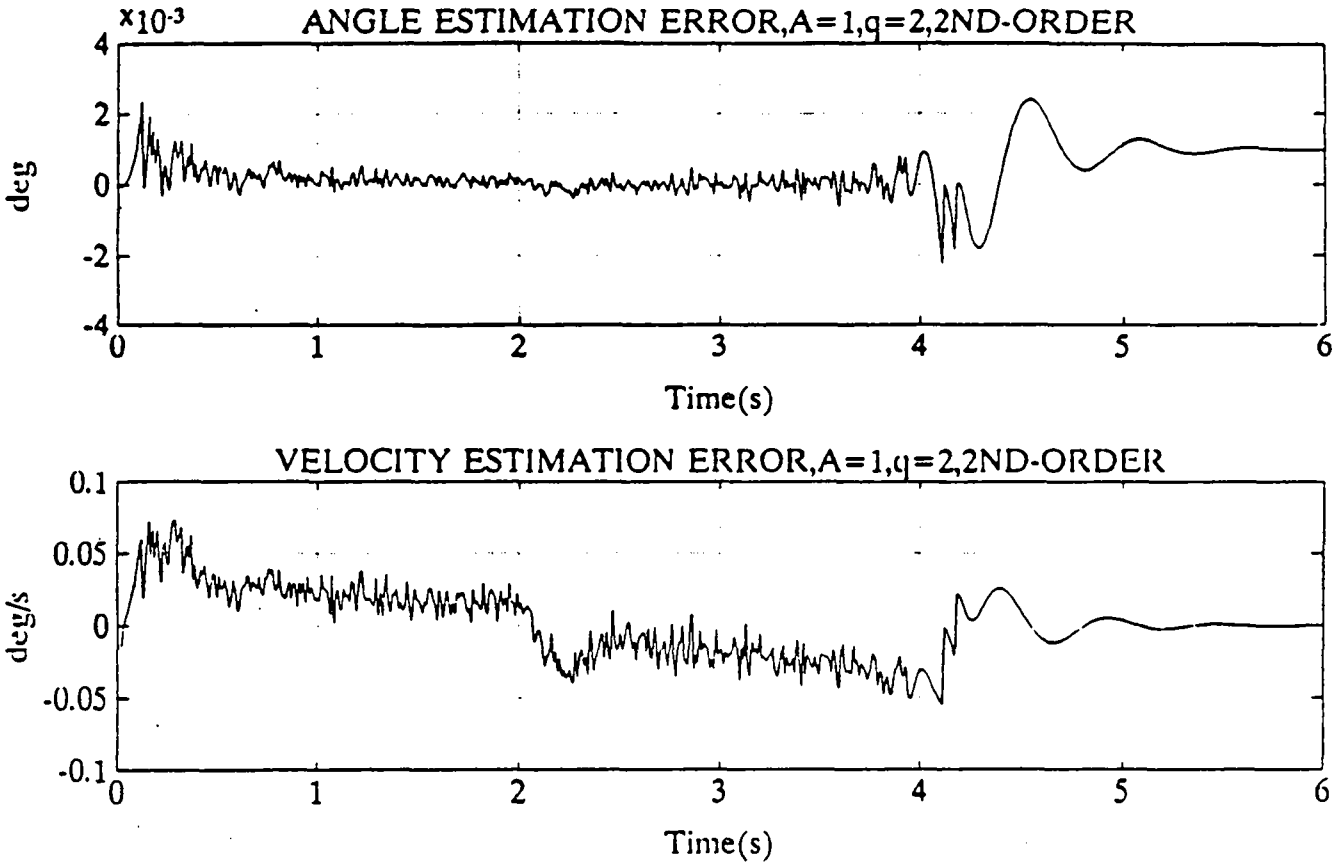


Figure 23 Slow trajectory, new algorithm, estimation errors for angle and velocity, double integrator, $q = 2$.

$J = 1$. This example is for the triple integrator and $A = 10$. The results are shown in Table III and in Figure 24. The mean and standard deviation figures vary relatively little as the proportionality factor k changes. The standard deviation figures are roughly half the values obtained with constant q. Figure 24 shows graphs for fixed (heavy) and adaptive (dotted) q. Adaptation is seen to reduce the error "bumps" due to rapid transitions, by allowing the filter to open when the derivative of the acceleration becomes large. In Figure 25, the computed standard deviation of the velocity and acceleration errors are shown. These are based on the quasi steady-state behavior of the solution of the Riccati equation: the covariance is computed on a pointwise basis, using $q(t)$ in the results of Theorem 1. If these curves are compared with the error curves of Figure 24, it appears that a useful estimate of error size can indeed be obtained in this manner. The low-velocity modification is not taken into account in the covariance computation, so that errors at low velocity are less than the covariance results would indicate.

Example 17

This example is a repeat of the previous one, except that $A = 1$. Results are given in Table IV and in Figure 26. By comparison with the case where $q = 20$, the mean and standard deviation figures for the angle and velocity errors are not significantly different. The acceleration error shows only a 25% improvement in standard deviation. The graphs of Figure 26 support these observations: the adaptation (dotted curve) is only mildly effective at smoothing out the error peaks.

Example 18

In this example, the double-integrator model is used with adaptive q and $A = 10$. Here, $q(t) = ka_d^2(t)$, where a_d is the acceleration given by the (erroneous) model. Figure 27 shows the errors for $k = 200$.

Mean and standard deviation are given in Table III. The improvement over the fixed-q case is marginal.

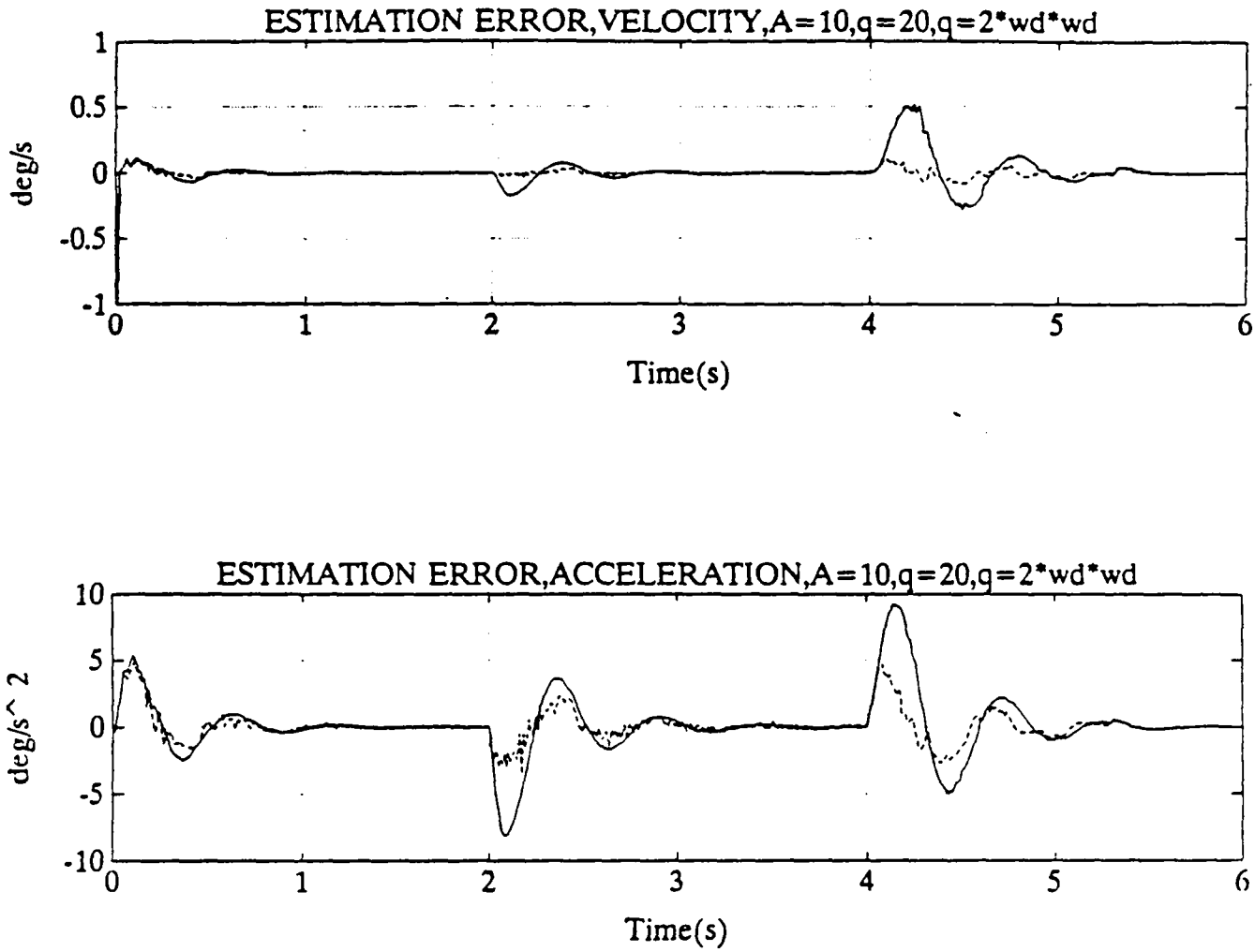


Figure 24 Fast trajectory, new algorithm, estimation errors for velocity and acceleration, triple integrator, $q = 20$ (heavy line) and adaptive q (dotted).

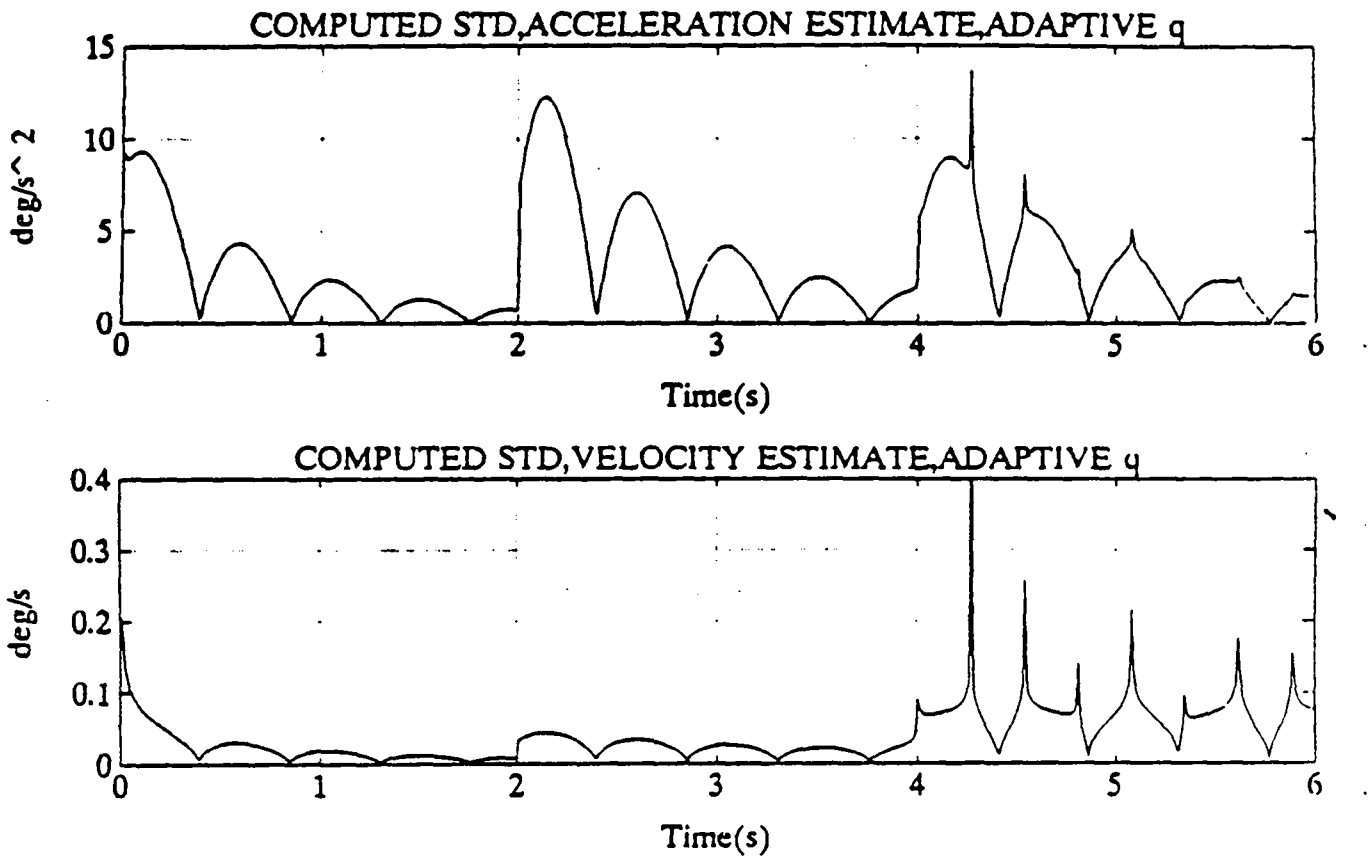


Figure 25 Fast trajectory, new algorithm, computed error standard deviations, $q = 2\omega_d^2$.

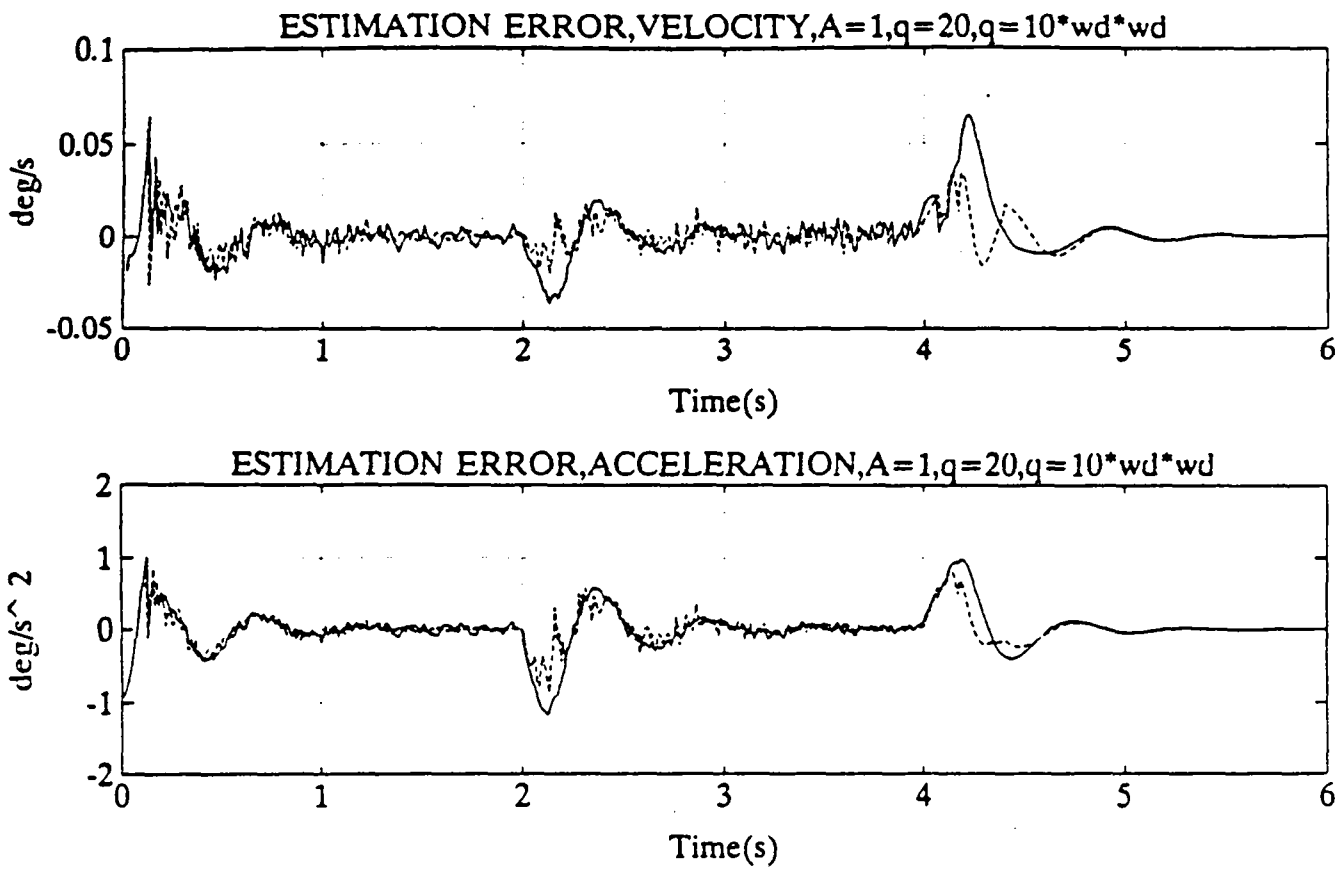


Figure 26 Slow trajectory, new algorithm, estimation errors for velocity and acceleration, triple integrator, $q = 20$ (heavy line) and adaptive q (dotted).

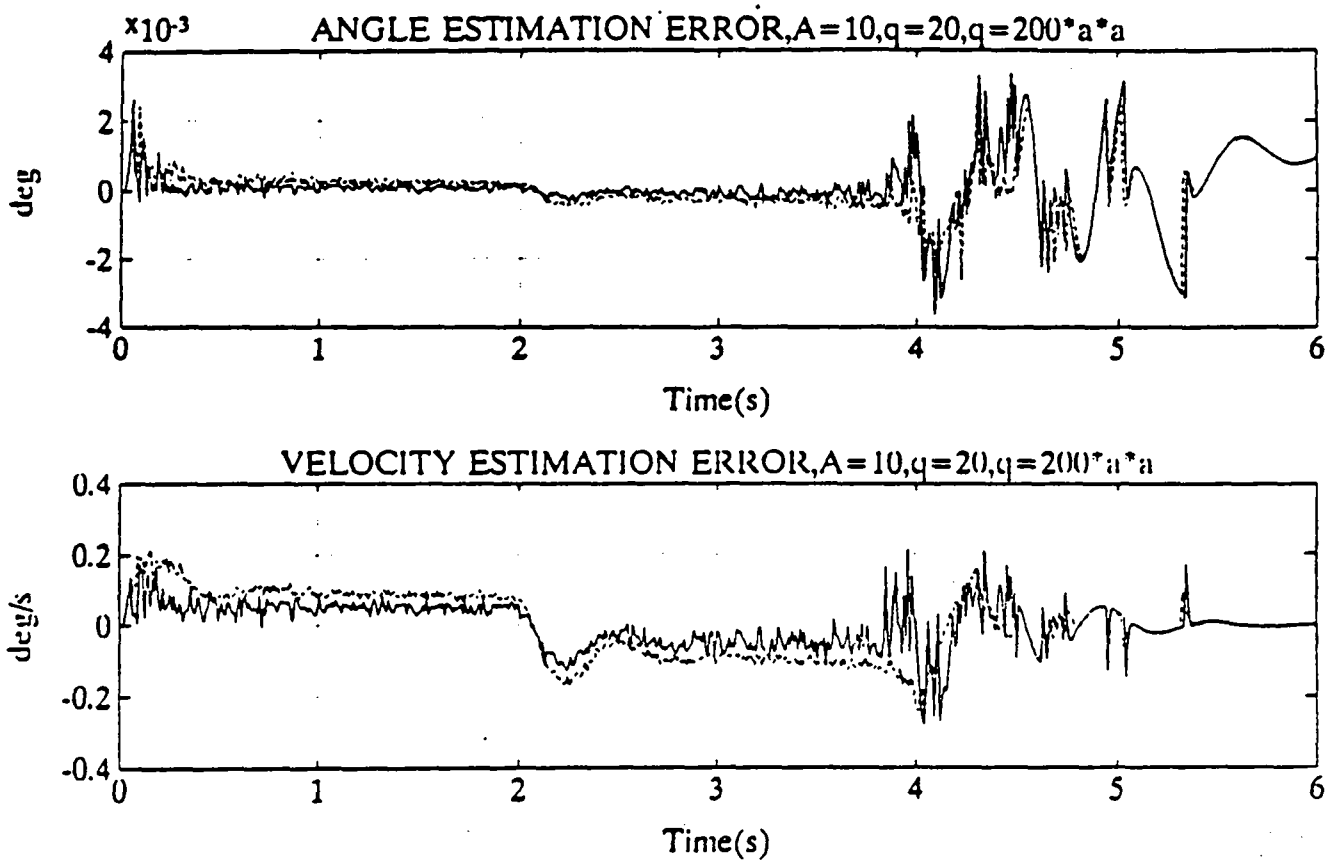


Figure 27 Fast trajectory, new algorithm, estimation errors for angle and velocity, double integrator, $q = 20$ (heavy line) and adaptive q (dotted).

Method	A	q	Angle (deg)		Velocity (deg/s)		Acceleration (deg/s^2)	
			Mean	Std	Mean	Std	Mean	Std
Finite Diff	10	-	9.13e-4	1.15e-3	1.47e-4	0.134	-16.0	18.8
New Method, Triple Integ.	"	20	-1.32e-4	1.89e-3	1.02e-2	9.85e-2	.135	2.23
"	"	200	1.20e-4	5.80e-3	4.26e-3	6.42e-2	.124	1.65
"	"	1000	-284e-4	8.79e-4	1.68e-3	5.09e-2	.113	1.33
"	"	10^4	9.99e-5	7.66e-4	2.15e-3	2.36e-2	.0818	1.04
New Method, Double Integ.	"	2	-3.35e-4	1.07e-3	-4.06e-3	8.83e-2	-	-
"	"	20	-3.27e-4	1.10e-3	-4.67e-3	8.75e-2	-	-
"	"	200	6.68e-5	8.12e-4	-2.54e-3	7.68e-2	-	-
"	"	1000	-3.19e-4	1.06e-3	-2.91e-3	7.66e-2	-	-
New Method, Triple Integ. Adaptive q	"	ω_d^2	8.69e-5	5.49e-4	5.78e-4	5.00e-2	.0887	1.22
"	"	$2\omega_d^2$	3.78e-5	6.32e-4	-2.24e-4	4.78e-2	.0782	1.11
"	"	$10\omega_d^2$	8.54e-5	7.25e-4	-9.66e-4	4.54e-2	.0368	.939
"	"	$20\omega_d^2$	5.08e-5	7.49e-4	-1.79e-3	4.64e-2	.0123	.938
"	"	$100\omega_d^2$	-3.28e-4	9.69e-4	-3.80e-3	4.96e-2	-.0563	1.26
New Method, Double Integ. Adapt. q	"	$10a^2$	-2.95e-4	9.95e-4	-1.16e-3	7.42e-2	-	-
"	"	$200a^2$	9.86e-5	9.06e-4	-3.06e-4	7.71e-2	-	-

TABLE III

Method	A	q	Angle (deg)		Velocity (deg/s)		Acceleration (deg/s^2)	
			Mean	Std	Mean	Std	Mean	Std
Finite Diff	1	-	8.43e-4	1.49e-3	1.64e-4	1.12e-1	-11.7	15.5
New Method Triple Integ.	"	20	3.78e-4	5.45e-4	-1.65e-4	4.26e-2	4.70e-3	.283
"	"	200	3.54e-4	6.57e-4	-3.50e-4	4.19e-2	9.79e-3	.229
"	"	1000	3.60e-4	7.04e-4	6.39e-5	4.26e-2	2.37e-2	.256
New Method Double Integ.	"	2	3.34e-4	6.56e-4	-5.58e-4	4.68e-2	-	-
"	"	20	3.76e-4	7.14e-4	1.37e-3	4.58e-2	-	-
"	"	200	4.06e-4	7.41e-4	4.46e-3	5.11e-2	-	-
New Method Triple Integ. Adaptive q	"	ω_d^2	3.65e-4	5.10e-4	-1.08e-4	4.32e-2	7.96e-3	.276
"	"	$2\omega_d^2$	3.59e-4	5.35e-4	-1.23e-4	4.23e-2	9.30e-3	.255
"	"	$10\omega_d^2$	3.46e-4	6.39e-4	-4.30e-4	4.17e-2	9.36e-3	.215
"	"	$100\omega_d^2$	3.45e-4	6.86e-4	-4.06e-4	4.22e-2	1.46e-2	.247

TABLE IV

Simulation runs were also made using a model of the third derivative signal. Results were only marginally improved: it appears that substantial improvement can only come from a very good model of the signal.

3.9.1 Conclusions

a) Compared to the fixed-interval Kalman filter, the error standad deviation figures show improvement by a factor of roughly two. That statement is somewhat misleading, however; it appears that the error transients during rapid changes in acceleration account for most of the error standard deviation. The new method is therefore only twice as effective at cutting down these transients. On the other hand, visual observation of the error graphs makes it obvious that the errors between transient pulses are much reduced. That is to be expected because: i) the sampling time is shorter, since it is the interarrival time of the encoder pulses rather than the control system sampling period; ii) the observation noise variance is smaller, since it is related to the encoder level error rather than the encoder resolutions.

b) As q increases, the filter "opens up" and transient performance improves, but the general error level increases. There appears to be an optimum, but it is rather broad.

c) The optimum q depends on the trajectory amplitude, in reality equivalent to velocity. The adaptive q method may be used to do the adjustment, if an estimate of the third derivative of the third derivative (acceleration, in the second-order case) is available. The improvement over the performance with a fixed q optimized for a given trajectory is marginal, but it appears that a single adaptive rule may work well for a number of trajectories.

d) The use of a signal model as a filter input is only of marginal use, unless the model is really quite close to the real signal. For higher derivatives, that is difficult to achieve. It is noted that the acceleration estimate from a third-order filter is much better than

most realistically obtainable a priori models of the acceleration. This suggests that a fourth-order filter would provide a good model of the third derivative, and that such a filter could be equivalent to a third-order filter with a good third derivative model.

Appendix A.

Proof of Lemma 1:

By Parseval's Theorem,

$$P_0 = \frac{q}{2\pi j}\int_{-j\infty}^{j\infty}(-sI - A^T + C^T K)^{-T}\Gamma\Gamma^T(sI - A^T + C^T K)^{-1}ds$$
$$= \frac{q}{2\pi j}\int_{-j\infty}^{j\infty}\frac{Adj(-sI - A^T + C^T K)^T\Gamma\Gamma^T Adj(sI - A^T + C^T K)}{p(s)p(-s)}ds \qquad (A.1)$$

where $p(s)$ is the closed-loop characteristic polynomial. For the all-pole system, it is known [eg 14] that the poles tend to the Butterworth pattern, i.e. satisfy $(-b_i)^{2n} = \frac{(-1)^{n+1}}{T}$ and $|b_i| \to \frac{1}{\mu}, \mu = T^{1/2n}$.

Now,

$$\Gamma^T Adj = [Adj_{n1} \quad Adj_{n2} \cdots Adj_{nn}].$$

To ease notation, let $M_i = [Adj(sI - A^T + C^T K)]_{ni}$

Now,

$$Adj(sI - A^T + C^T K).(sI - A^T + C^T K) = p(s)I. \qquad (A.2)$$

With $K = [k_1 \quad k_2 \cdots k_n]$,

$$sI - A^T + C^T K = \begin{bmatrix} s+k_1 & k_2 & k_3 & \cdots & \cdots & k_n \\ -1 & s & 0 & \cdots & \cdots & 0 \\ 0 & -1 & s & \cdots & \cdots & a_0 \\ 0 & 0 & -1 & \cdots & \cdots & a_1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & s & a_{m-2} \\ 0 & 0 & 0 & \cdots & -1 & a_{m-1} \end{bmatrix}$$

Writing out the last row of Equation (A.2) yields

$$(s + k_1)M_1 - M_2 = 0$$

$$k_2 M_1 + sM_2 - M_3 = 0$$

$$k_i M - 1 + sM_i - M_{i+1} = 0$$

$$k_1 M_1 + a_0 M_3 + a_1 M_4 + \cdots + a_{m-2} M_{n-1} + (s + a_{m-1}) M_n = p(s). \qquad (A.3)$$

There follows

$$M_1 = 1$$

$$M_i = s^{i-1} + k_1 s^{i-2} + \cdots + k_{i-1}, i = 2, 3, \cdots n.$$

Inserting in Equation (A.3),

$$s^n + (k_1 + a_{m-1}) s^{n-1} + \cdots + (k_i + a_{m-1} k_{i-1} + \cdots + a_{m-i}) s^{n-i} + \cdots =$$

$$p(s) = (s + b_1)(s + b_2) \cdots (s + b_n).$$

Matching powers yields

$$k_1 + a_{m-1} = \Sigma b_i \text{ so } k_1 \sim 1/\mu$$

$$k_2 + a_{m-1} k_1 + a_{m-2} = \Sigma b_{i1} b_{i2} \text{ so } k_2 \sim 1/\mu^2.$$

In general, $k_i \sim 1/\mu^i$ so that

$$M_1 = 1, M_2(b_k) \sim 1/\mu \cdots M_i(b_k) \sim 1/\mu^{i-1} \qquad (A.4)$$

Equation (A.1) can be written as

$$P_{ij} = \frac{q}{2\pi j} \int_{-j\infty}^{j\infty} \frac{M_i(-s) M_j(s)}{p(s) p(-s)} ds \qquad (A.5)$$

The numerator of the integrand is of degree at most $2n-2$, the denominator is of degree $2n$ so the integral can be evaluated as a contour integral enclosing the LHP.

For Equation (A.5),

$$\text{Residue at } s = -b_k = \frac{M_i(b_k) M_j(-b_k)}{p(b_k)(-b_k + b_1) \cdots (-b_k + b_{k-1})(-b_k + b_{k+1}) \cdots (-b_k + b_n)}$$

Because of Equation (A.4),

$$P_{ij} \sim \frac{(\frac{1}{\mu})^{i+1+j-1}}{(\frac{1}{\mu})^{2n-1}} = \mu^{2n-i-j+1}$$

$$= T^{\frac{2n-i-j+1}{2n}}$$

Proof of Theorem 2:

Note first that

$$CPC^T + R = P_{11} + R = P^*_{11}\mu^{2n-1} + R \approx R$$

Writing Equation (27) element by element, for $i, j \neq n$:

$$P_{ij} = P_{ij} - \frac{P_{1i}P_{ij}}{R} + T(P_{i+1,j} - \frac{P_{1,i+1}P_{1j}}{R}$$
$$+ T(P_{i,j+1} - \frac{P_{1i}P_{1,j+1}}{R}) + T^2(P_{i+1,j+1} - \frac{P_{1,i+1}P_{1,j+1}}{R})$$

or

$$- \frac{P^*_{1i}P^*_{1j}}{R}\mu^{4n-i-j} + P^*_{i+1,j}\mu^{4n-i-j} + P^*_{i,j+1}\mu^{4n-i-j}$$
$$- \frac{P^*_{1,i+1}P^*_{1j}}{R}\mu^{6n-i-j-1} - \frac{P^*_{1i}P^*_{1,j+1}}{R}\mu^{6n-i-j-1}$$
$$+ P^*_{i+1,j+1}\mu^{6n-i-j-1} - \frac{P^*_{1,i+1}P^*_{1,j+1}}{R}\mu^{8n-i-j-2} = 0$$

Dividing by μ^{4n-i-j} and dropping positive powers of μ leaves the first 3 terms, which are the LHS of Equation (24).

For $j = n, i \neq n$:

$$P_{in} = T\Big[-a_0(P_{i3} - \frac{P_{1i}P_{13}}{R}) - a_1(P_{i4} - \frac{P_{1i}P_{14}}{R}) -$$
$$- a_{m-1}(P_{in} - \frac{P_{1i}P_{1n}}{R})\Big] + P_{in} - \frac{P_{1i}P_{1n}}{R} +$$
$$+ T^2\Big[-a_0(P_{i+1,3} - \frac{P_{1,i+1}P_{i3}}{R}) - \cdots - a_{m-1}(P_{i+1,n} - \frac{P_{1,i+1}P_{1n}}{R})\Big] +$$
$$+ T(P_{i+1,n} - \frac{P_{1,i+1}P_{1n}}{R})$$

or

$$- \frac{P^*_{1i}P^*_{1n}}{R}\mu^{3n-i} + P^*_{i+1,n}\mu^{3n-i} - a_0P^*_{i3}\mu^{4n-i-2}$$
$$- a_1P^*_{i4}\mu^{4n-i-3} - \cdots - a_{m-1}P^*_{in}\mu^{3n-i+1} - a_0\frac{P^*_{1i}P^*_{13}}{R}\mu^{6n-i-3}$$
$$- \cdots - a_{m-1}\frac{P^*_{1i}P^*_{1n}}{R}\mu^{5n-i} - \frac{P^*_{1,i+1}P^*_{1n}}{R}\mu^{5n-i-1} = 0$$

Dividing by μ^{3n-i} gives the LHS of Equation (25), plus terms in $\mu, \mu^2, \cdots$.

For $i = j = n$:

$$-\frac{P_{1n}^2}{R} + qT - T[2a_0(P_{3n} - \frac{P_{13}P_{1n}}{R}) + 2a_1(P_{4n} - \frac{P_{14}P_{1n}}{R})$$
$$+ \cdots + 2a_{m-1}(P_{nn} - \frac{P_{1n}^2}{R})] + T^2[a_0^2(P_{33} - \frac{P_{13}^2}{R})$$
$$+ 2a_0a_1(P_{34} - \frac{P_{13}P_{14}}{R}) + \cdots + 2a_0a_{m-1}(P_{3n} - \frac{P_{13}P_{1n}}{R})$$
$$+ a_1^2(P_{44} - \frac{P_{14}^2}{R}) + \cdots + 2a_1a_{m-1}(P_{4n} - \frac{P_{14}P_{1n}}{R})$$
$$+ \cdots + a_{m-1}^2(P_{nn} - \frac{P_{1n}^2}{R}) = 0$$

so that

$$-\frac{P_{1n}^{*2}}{R}\mu^{2n} + q\mu^{2n} - 2a_0P_{3n}^*\mu^{3n-2} - 2a_1P_{4n}^*\mu^{3n-3} - \cdots$$
$$- 2a_{m-1}P_{nn}^*\mu^{2n+1} + 2a_0\frac{P_{13}^*P_{1n}^*}{R}\mu^{5n-3} + \text{ Higher over terms } = 0$$

Dividing by μ^{2n} leaves the LHS of Equation (26), plus terms in $\mu, \mu^2, \cdots$.

2

Appendix B.

For $t_k \geq t_j$, from Equation (62),

$$\mathrm{E}[\mathbf{x}_s(t_k)\mathbf{x}_s^T(t_j)] = \begin{bmatrix} 1 & t_k - t_j & \frac{1}{2}(t_k - t_j)^2 \\ 0 & 1 & t_k - t_j \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{20}t_j^5 & \frac{1}{8}t_j^4 & \frac{1}{6}t_j^3 \\ \frac{1}{8}t_j^4 & \frac{1}{3}t_j^3 & \frac{1}{2}t_j^2 \\ \frac{1}{6}t_j^3 & \frac{1}{2}t_j^2 & t_j \end{bmatrix} q.$$

For $t_k < t_j$,

$$\mathrm{E}[\mathbf{x}_s(t_k)\mathbf{x}_s^T(t_j)] = q \begin{bmatrix} \frac{1}{20}t_k^5 & \frac{1}{8}t_k^4 & \frac{1}{6}t_k^3 \\ \frac{1}{8}t_k^4 & \frac{1}{3}t_k^3 & \frac{1}{2}t_k^2 \\ \frac{1}{6}t_k^3 & \frac{1}{2}t_k^2 & t_k \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ t_j - t_k & 1 & 0 \\ \frac{1}{2}(t_j - t_k)^2 & t_j - t_k & 1 \end{bmatrix}.$$

From Equation (61),

$$\mathrm{E}[y_s(t_k)y_s(t_j)] = \begin{cases} q[\frac{t_j^5}{20} + (t_k - t_j)\frac{t_j^4}{8} + \quad (t_k - t_j)^2\frac{t_j^3}{12}], t_k \geq t_j \\ q[\frac{t_k^5}{20} + (t_j - t_k)\frac{t_k^4}{8} + \quad (t_j - t_k)^2\frac{t_k^3}{12}], t_k < t_j \end{cases}$$

Also,

$$\mathrm{E}[y_s(t_k)\mathbf{x}_s^T(T)] = q[\frac{t_k^5}{20} + \frac{t_k^4}{8}(T - t_k) + \frac{t_k^3}{12}(T - t_k)^2; \frac{t_k^4}{8} + \frac{t_k^3}{6}(T - t_k); \frac{t_k^3}{6}].$$

For equally-spaced observations, $t_k = \frac{k}{N}T$. Then, for $T = 1$ (normalized time)

$$\sum_{k=1}^{N}\sum_{j=1}^{N} \underline{\phi}\,(t_k)\mathrm{E}[y_s(t_k)y_s(t_j)]\,\underline{\phi}^T\,(t_j) = \mathcal{A}^T\mathcal{Y}\mathcal{A}$$

where

$$\mathcal{A} = \begin{bmatrix} 1 & 1/N & \frac{1}{2}1/N^2 \\ 1 & 2/N & \frac{1}{2}(2/N)^2 \\ \cdots & \cdots & \cdots \\ 1 & \frac{N-1}{N} & \frac{1}{2}(\frac{N-1}{N}^2 \\ 1 & 1 & \frac{1}{2} \end{bmatrix}$$

$$y_{kj} = \begin{cases} q'[\frac{1}{20}(\frac{j}{N})^5 + \frac{1}{8}(k - j)(\frac{j}{N})^4 + \quad \frac{(k-j)^2}{12}(\frac{j}{N})^3], k \geq j \\ q'[\frac{1}{20}(\frac{k}{N})^5 + \frac{1}{8}(j - k)(\frac{k}{N})^4 + \quad \frac{(j-k)^2}{12}(\frac{k}{N})^3], k \geq j \end{cases}$$

It can be shown that

$$\{[\mathcal{A}^T\mathcal{Y}\mathcal{A}\}_{kj} = q'\frac{\alpha_0 + \alpha_1 N + \cdots + \alpha_{4+k+j}N^{4+k+j}}{N^{2+k+j}} \qquad (B.1)$$

The α_i were obtained by i) calculating $\mathcal{A}^T\mathcal{Y}\mathcal{A}$ for $5+k+j$ values of N and ii) solving the resulting linear equations for the α_i. The values are given in Table B.1.

The other matrix calculation is, for $T=1$,

$$\sum_{k=1}^{N} \underline{\phi}(t_k)\mathrm{E}[y_s(t_k)\mathbf{x}_s^T(T)] = \mathcal{A}^T\mathcal{X}$$

$$k^{th} \text{ row of } \mathcal{X} = q'\Big[\frac{1}{20}(\frac{k}{N})^5 + \frac{1}{8}(\frac{k}{N})^4\frac{N-k}{N} + \frac{1}{12}(\frac{k}{N})^3(\frac{N-k}{N})^2;$$
$$\frac{1}{8}(\frac{k}{N})^4 + \frac{1}{6}(\frac{k}{N})^3\frac{N-k}{N}; \frac{1}{6}(\frac{k}{N})^3\Big]$$

The result is of the form

$$\{\mathcal{A}^T\mathcal{X}\}_{kj} = q'\frac{\beta_0 + \beta_1 N + \cdots \beta_{n(k,j)}N^{n(k,j)}}{N^{n(k,j)-1}} \qquad (B.2)$$

The coefficients are given in Table B.2.

For the double integrator,

$$\mathrm{E}[y_s(t_k)y_s(t_j) = \begin{cases} q'[\frac{1}{3}t_j^3 + (t_k - t_j)\frac{t_j^2}{2}, & t_k \geq t_j \\ q'[\frac{1}{3}t_k^3 + (t_j - t_k)\frac{t_k^2}{2}, & t_k < t_j \end{cases}$$

$$\mathrm{E}[y_s(t_k)\mathbf{x}_s^T(T)] = q'[\frac{1}{3}t_k^3 + (T - T_k)\frac{t_k^2}{2} \quad \frac{1}{2}t_k^2]$$

and

$$\mathcal{A} = \begin{bmatrix} 1 & 1/N \\ 1 & 2/N \\ \cdots & \cdots \\ 1 & \frac{N-1}{N} \\ 1 & 1 \end{bmatrix}$$

It is calculated that

$$\{\mathcal{A}^T\mathcal{Y}\mathcal{A}\}_{kj} = q'\frac{\alpha_0 + \alpha_1 N + \cdots + \alpha_{k+j+2}N^{k+j+2}}{N^{k+j}} \qquad (B.3)$$

The coefficients are displayed in Table B.3.

The matrix $\mathcal{A}^T\mathcal{X}$ has the same form as that of Equation B.2, and the coefficients are shown in Table B.4.

TABLE B.1

Matrix Element	11	12	13	22	23	33
$10^4.\alpha_0$	1.984	0.992	-0.298	0.794	0	0.035
$10^4.\alpha_1$	6.944	0.827	-0.744	1.984	0.149	0.00284
$10^4.\alpha_2$	27.778	0	-2.287	-3.417	-0.248	-0.086
$10^4.\alpha_3$	104.167	31.829	-4.340	-6.944	-5.518	-0.742
$10^4.\alpha_4$	180.556	125.000	17.824	41.204	-7.813	-5.234
$10^4.\alpha_5$	138.889	186.343	72.917	145.833	25.000	-6.076
$10^4.\alpha_6$	39.683	124.008	97.057	187.169	83.333	15.873
$10^4.\alpha_7$	-	31.022	57.168	109.127	95.424	46.875
$10^4.\alpha_8$	-	-	12.704	24.250	49.727	47.826
$10^4.\alpha_9$	-	-	-	-	9.945	22.445
$10^4.\alpha_{10}$	-	-	-	-	-	4.081

TABLE B.2

Matrix Element	11	12	13	21	22	23	31	32	33
$n(k_1 j)$	4	4	5	2	4	4	6	5	4
$10^4.\beta_0$	6.944	13.889	4.861	416.667	-20.833	-55.556	-1.488	-12.153	-69.44
$10^4.\beta_1$	0	0	-11.420	833.3333	0	0	0	-23.534	0
$10^4.\beta_2$	104.167	277.778	2.025	416.667	381.944	555.556	-12.153	-5.064	347.222
$10^4.\beta_3$	250.000	625.000	145.351	-	625.000	833.333	0	244.261	416.667
$10^4.\beta_4$	138.889	333.333	250.058	-	263.889	333.333	93.750	312.355	138.889
$10^4.\beta_5$	-	-	109.124	-	-	-	125.000	109.134	-
$10^4.\beta_6$	-	-	-		-	-	44.891		-

TABLE B.3

Matrix Element	11	12	22
$10^2.\alpha_0$	0.55555	0.27778	0.15874
$10^2.\alpha_1$	4.1667	1.2500	0.55555
$10^2.\alpha_2$	11.111	5.5556	1.9444
$10^2.\alpha_3$	12.500	11.806	6.9444
$10^2.\alpha_4$	5.000	10.833	11.944
$10^2.\alpha_5$	-	3.6111	9.1667
$10^2.\alpha_6$			2.6190

TABLE B.4

Matrix Element	11	12	21	22
$n(k_1 J)$	2	2	4	2
$10^2.\beta_0$	4.1667	8.3333	0.55555	12.500
$10^2.\beta_1$	16.667	25.000	0	25.000
$10^2.\beta_2$	12.500	16.667	6.9444	12.500
$10^2.\beta_3$	-	-	16.667	-
$10^2.\beta_4$	-	-	9.1667	-

REFERENCES

[1] An, C.H., G.C. Atkeson, J.M. Christopher and J.M. Hollerbach, "Estimation of inertial parameters of rigid body links of manipulators", IEEE Conf. on Decision and Control, Fort Lauderdale, (1985).

[2] Studenny, J and P.R. Bélanger, "Robot manipulator control by acceleration feedback: stability, design and performance issues", 25th IEEE Conf. on Decision and Control, Athens, Dec. 1986.

[3] Luo, G.L. and G.N. Saridis, "L-Q design of P ID controllers for robot arms", IEEE J. of Robotics and Automation, Vol. RA-1, pp 152-158 (1985).

[4] Sinha, N.K., B. Szavados and D. C. di Cenzo, "New high precision digital tachometer", Electron. Lett., vol 7, pp 174-176, (1971).

[5] Hoffman de Visme, G., "Digital processing unit for evaluating angular acceleration", Electron. Eng., vol 40, pp 183-188, (1968).

[6] Dunworth, A., "Digital instrumentation for angular velocity and acceleration", IEEE Trans. Instrum. Meas., vol IM-18, pp 132-138, (1969).

[7] Habibullah, B., H. Singh, K.L Soo and L.C. Ong, "A new digital speed transducer", IEEE Trans. Ind. Electron. Contr. Instr., vol IECI-25, pp 339-343, (1978).

[8] Wallingford, E.E. and J.D. Wilson, "High resolution shaft speed measurements using a microcomputer", IEEE Trans. Instr. and Meas., vol IM-26, pp 113-116 (1977).

[9] Glad, T., and L. Ljung, "Velocity estimation from irregular, noisy position measurements", IFAC 9th World Congress, Budapest, 1984.

[10] Christiansen, C.F., R. Battaitto, D. Fernandez and E. Tacconi, "Digital measurement of angular velocity for speed control", IEEE Trans. Ind. Electron., vol 36, pp 79-83 (1989).

[11] Kwakernaak, H. and R. Swan, "The maximally achievable accuracy of linear optimal regulators and linear optimal filters", IEEE Trans. on Aut. Control, vol AC-17, pp

79-86, (1972).

[12] Friedland, B., "Limiting forms of optimum stochastic linear regulators", J. Dyn. Syst. Meas. Contr., vol , pp 134-141, (1971).

[13] O'Malley, R.E. jr. and A. Jameson, "Singular perturbation and singular arcs - Part I", IEEE Trans. Auto. Control, vol AC-20, pp 218-227, (1975).

[14] Kailath, T., "Linear Systems", Prentice-Hall, 1980.

ISSN 0249 - 6399