On the Velocity and Acceleration Estimation from Discrete Time-Position Sensors ⋆

L.J. Puglisi, R.J. Saltaren, C.E. Garcia Cena

Centro de Automática y Robótica, UPM-CSIC,
José Gutierrez de Abascal, Madrid, 28006, Spain
(e-mail: lisandro.puglisi@ alumnos.upm.es, rsaltaren@etsii.upm.es, cgarcia@upm.es).

Abstract: In this work is addressed the topic of estimation of velocity and acceleration from digital position data. It is presented a review of several classic methods and implemented with real position data from a low cost digital sensor of a hydraulic linear actuator. The results are analyzed and compared. It is shown that static methods have a limited bandwidth application, and that the performance of some methods may be enhanced by adapting its parameters according to the current state.

Keywords: digital encoder, velocity estimation, acceleration estimation, Kalman Filter

1. INTRODUCTION

Many work has been done in the field of estimation of velocity and acceleration from digital encoder data (Brown et al. (1992), Merry et al. (2010), Liu (2002), Ovaska and Valiviita (1998), Baran et al. (2010), Song et al. (2009)). However, most of them are focussed on angular sensors and scarce information regarding their implementation in linear position encoder can be found.

In typical applications, angular sensors are not directly coupled to the measured variable instead there's a gear between the encoder and the angular variable. Consequently, the constraints imposed by the resolution of the encoder is somehow diminished, since the relative variation of the angular variable is mapped into the encoder amplified by the multiplication factor of the gear.

A linear position sensor is coupled directly to the measured variable and it is not possible to make use of this treacherous enhancement. In general, the resolution of medium cost sensors is allocated in the $0.1 - 0.5mm$ range. Even though this resolution could be considered appropriate for general applications, the implementation of these sensors in low velocity tasks brings several complications, especially when the control loop depends on the estimation of velocity and its time derivatives (Han et al. (2007), Choi and Jung (2011), Emaru et al. (2009), Ohmae et al. (1982), Tsuji et al. (2005)).

Therefore, in this work is presented a review of the classic methods used for the estimation of velocity and acceleration and they are implemented on a digital linear position sensor. Several concepts are introduced and combined providing more than twelve alternatives for the estimation of velocity and acceleration. This survey could be though as a small tutorial for those readers that are beginning to explore the field of velocity and acceleration estimation remarking the pros and cons of each method.

This work is organized as follows. Firstly, a brief introduction to the problem associated to digital signals and the calculation of its time derivatives is presented. Following this, some of the most common methods for velocity and acceleration estimation are introduced and tested with real data. An objective performance comparison is presented by introducing the definition of some Key Performance Indexes and finally, the results are discussed.

2. ENCODER SIGNAL ANALYSIS

In order to understand the problem associated with the estimation of velocity and acceleration from a digital position signal it is better to clarify some of its characteristics.

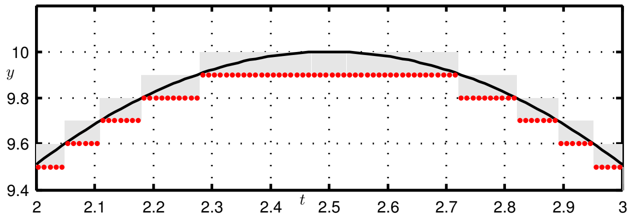
First of all, when dealing with the information provided by a digital sensor, there are two main issues that must be remarked:

(1) The resolution of the encoder is finite. Its output can only take some defined states, therefore the true value of the measured variable resides between two adjacent states. The output of the encoder should not be interpreted as the true value.
(2) The information corresponds only to a finite sample of time. There's no true notion of what's happening between samples.

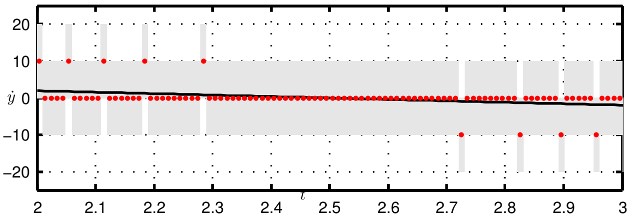
These characteristics are clearly depicted in Fig.1(a). The real data is presented in black line and the output of the encoder as red dots. As it can be observed, the signal obtained from the encoder clearly shows its discrete nature caused by the finite resolution and consequently the uncertainty of the measurement (represented in grey).

⋆ The authors would like to thank the financial support of the Spanish Government through the CICYT Project Ref. DPI2009-08778 and also to Comunidad de Madrid who supports the project ROBOCITY2030-II Ref. P2009/DPI-1559.

The problem with the finite resolution of the encoder is that for slow variations it delivers the same information for several consecutive samples, until the variation exceeds its resolution, presenting a laddered type signal (observe the red dots).



(a)



(b)

Fig. 1. Digital position signal analysis

Based on the afore mentioned, the output of the encoder y_k for every sample-time k can be expressed as (Shaowei and Shanming (2012)):

$$y_k = \left\lfloor \frac{y}{e_r} \right\rfloor e_r, \tag{1}$$

where $\lfloor \bullet \rfloor$ is the rounding to floor operator, and the error of the measurement is given by $y_{err} = y_k - y$, with $y_{err} \epsilon \, (-e_r, 0]$ and e_r is the resolution of the encoder.

Let us consider the output of the encoder presented as red dots in Fig.1(a), and let us apply the backward finite difference method (FD) to obtain its time derivative. The result is presented in Fig.1(b) in red dots, and it can be observed that two consecutive identical position data result in a zero flat line, while a change in the position turns into a spike. This estimation of the first time derivative does not resemble at all the real velocity (see black curve).

On the other hand, since the data has an intrinsic error given by the finite resolution of the encoder it can be easily found, by error propagation theory, that the estimated velocity will have an error bounded to $\dot{y}_{err} \epsilon (-e_r/T, e_r/T)$ (represented in grey in Fig.1(b)) , where T is the sample time.

From the bounded error it can be observed that the error of the estimated time derivative is related proportionally with the resolution of the encoder, therefore to reduce it the resolution should improve. It can also be extracted that by reducing the sampling period T, the bounded error increases.

3. EXPERIMENTAL SETUP

The experimental setup used in this work is a linear hydraulic actuator with a low cost digital linear position sensor LX-EP40 with a resolution of 2.45 $counts/mm$. A dSPACE DS1103 board is implemented as the control hardware, it captures data from the position sensor (y_k), and it also provides the control signal (x_v) to the servo-valve (see Fig. 2).

In order to objectively analyze the performance of each method, the following Key Performance Indexes (KPI) are defined:

(1) Average error:

$$KPI_1 = \frac{1}{n} \sum_{k=1}^{n} (\theta_k - \hat{\theta}_k). \tag{2}$$

(2) Absolute maximum relative error:

$$KPI_2 = \max_{k \epsilon [1,n]} \left| \frac{\theta_k - \hat{\theta}_k}{\theta_k} \right|. \tag{3}$$

(3) Average relative error:

$$KPI_3 = \frac{1}{n} \sum_{k=1}^{n} \left| \frac{\theta_k - \hat{\theta}_k}{\theta_k} \right|. \tag{4}$$

Where θ_k is the real data and $\hat{\theta}_k$ is the estimated variable at the kth sampled time. These KPI are evaluated for half a period (discarding the first and last 25 samples in order to avoid $\theta_k \approx 0$ and thus $KPI_i \approx \infty$), of the reference trajectory: $y = 100 + 25\,sin(2\pi f)$, with $f = 0.1Hz$, $f = 0.5Hz$ and $f = 1Hz$ (y is in mm), implementing a proportional position controller.

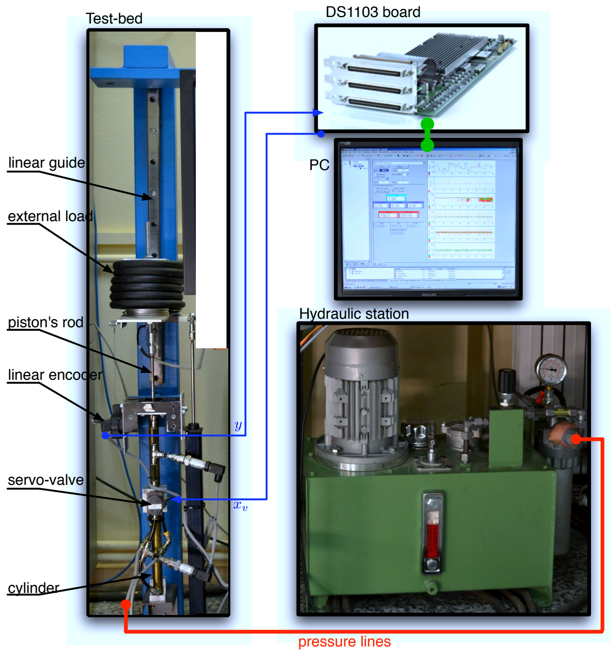


Fig. 2. Experimental Setup.

4. VELOCITY ESTIMATION

In this section, it is introduced some of the techniques implemented for the estimation of velocity. Hereafter, y_k is the output of the sensor and $\hat{\bullet}_k$ is the estimation for $\bullet$ at the kth sampled time.

4.1 Finite Difference and Filtering

The most simple practice for the estimation of the velocity and acceleration is to implement finite difference (FD) on the discrete time position data and then filter the result with a low pass filter (LPF).

The first order FD applied to the position data (in its backward formulation), leads to the following expression:

$$\dot{y}_k = \frac{y_k - y_{k-1}}{T}. \tag{5}$$

The analysis presented in (Belanger (1992)) suggests that exists an optimal sampling time to estimate the velocity and acceleration with FD, though in this work the sample time period is settled to $T = 1$ms for all the methods.

The result of the implementation of the FD over the raw data (y_k) is presented in Fig.5(a) - Fig.5(c) in red. As it was mentioned before, the estimation depends on the change between two consecutive samples. Therefore, in the low-speed range the estimation is rather poor and it improves while increasing the real velocity (i.e. more position changes between two consecutive samples). In fact, if the real velocity is $\dot{y} > \frac{e_r}{T}$, the estimation with FD will be very accurate with a time-lag T.

For the improvement of the estimation on the low-speed range a LPF is implemented. The design of the LPF is crucial for the performance of the estimated variable. A common practice is to implement Butterworth filters since they provide maximally flat magnitude for the wanted frequencies. The cut-off frequency of the filter must be selected depending on the frequency range of the application.

In Fig.5(a) - Fig.5(c) are presented the estimation of velocity implementing a $4th$ order Butterworh LPF with cut-off frequencies: $f_c = 10$Hz (blue curve), $f_c = 1$Hz (cyan curve) and $f_c = 0.1$Hz (green curve).

As it can be seen, the filtering with $f_c = 10$ Hz, does not eliminate the noise introduced by the derivative action of the FD in the low velocity range (see blue curve in Fig.5(a)). On the other hand, for greater velocity the filtering seems to work properly (i.e. it eliminates the noise), but it introduces a time-lag in the estimated velocity (see the blue curve and compare it with the real velocity in Fig.5(b) and Fig.5(c)).

On the pursuit of the suppression of the noise in the low-speed range, it is implemented a filter with $f_c = 0.1$ Hz. The action of this filter eliminates the noise in the low velocity range (see green curve in Fig.5(a)), but it also affects the amplitude and time-lag of the estimation for higher velocities (see green curve in Fig.5(b) and Fig.5(c)).

However, if the frequency range of work of y_k is restricted to a narrowed bandwidth, an acceptable performance can be achieved by selecting the cut-off frequency at the mid of this bandwidth. Even though, it must be remarked that the estimation will have some noise in the low-velocity range and time-lag for the high-velocity range.

4.2 Finite Difference from encoder events

The problem that arises from the finite resolution of the encoder is that for slow motions the signal provided by the encoder delivers the same data for several consecutive samples. In order to overcome this issue two classic methods based on two different sampling approaches (Goodwin et al. (2013), Astrom and Bernhardsson (2002)) are commonly implemented (Tsuji et al. (2005)):

- M method (fixed-time method): it counts the number of pulses from the optical encoder during a fixed interval of time.
- T method (fixed-position method): it measures the time that it takes to count a predefined number of pulses.

The latter one provides better results in the low-speed range, while the first one deteriorates it.

Based on the T-method approach and defining only one pulse to count, it is detected the minimum change of the encoder. This minimum change and the time of its occurrence is defined as an *encoder event* (EE) (Merry et al. (2010)) and it is given by the pair (t_{ee}, y_{ee}), i.e. the time when the event occurs and the current measure, respectively. An analysis of event sampling for nonlinear filtering is presented in (Cea and Goodwin (2012)).

It must be recalled that in this work the sampling time is constant, therefore $t_{ee} = nT$, where n is the number of consecutive samples until an event is detected.

By definition every EE represents a change on the output of the encoder, thus there's no identical consecutive EE (see the blue dots in Fig.3(a)). Hence, when the FD method is implemented over the EE, the estimation results in a spike-free curve as shown in Fig.3(b) in blue (compare it with the red dots in Fig.1(b)). On the other hand, it can also be observed that the estimation is much more like the real velocity (black curve in Fig.3(b)), even though it presents a laddered wave form.

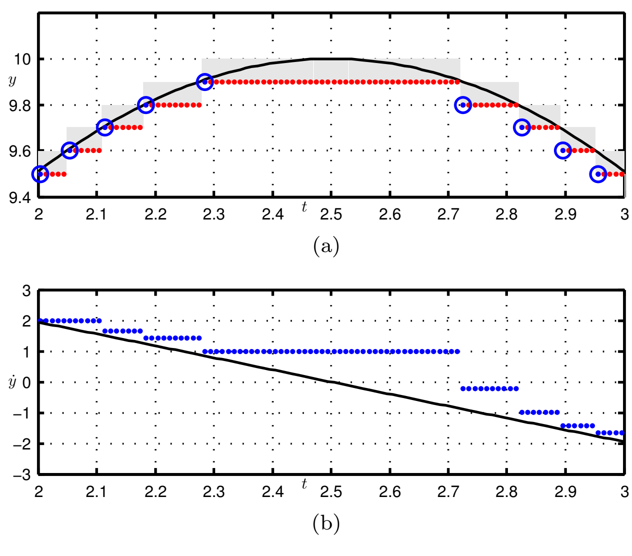


Fig. 3. Definition of encoder events. 3(a): position data (real position in black line, encoder's signal in red, and EE in blue dots). 3(b): Velocity estimation implementing FD over EE. (real velocity in black and velocity estimation in blue dots.

The concept of EE with FD is implemented on the position data and the results are depicted in Fig.5(d) - Fig.5(f) in red. As it is observed, there still exists some spurious spikes near the region of low accelerations (maximum velocity). In the low-speed region it is appreciated that the signal is distorted. This is caused by the fact that there's a long period of time without detecting any new event (see Fig. 3).

The spikes on the low-acceleration region can be eliminated by smoothing the estimation by *skipping* some events in the FD, providing a *filtering* effect (Merry et al. (2010)). This approach is presented in Fig.5(d) - Fig.5(f), by skipping one (blue curve), five (green curve) and ten (cyan curve) consecutive events.

As it can be seen, the skipping option provides filtering to the estimation (compare them with the red curve). The question now is how many events must be skipped in order to provide a good estimation. In these figures, it can be observed that skipping one event is not enough (observe the blue curves), and the smoothness of the estimation improves when more events are skipped. However, skipping events leads to time-lags in the estimated velocity, thus there is always a compromise between smoothness and time delays. An alternative could be to skip events according to the last estimated velocity as proposed in (Liu (2002)), providing adaptability to the method.

4.3 Polynomial Fitting

The lack of information between the states of the encoder could be surpassed by estimating the real value of the measured variable by fitting y_k to a polynomial of order n (Merry et al. (2010), Brown et al. (1992)), as follows:

$$\hat{y}_k = c_n t_k^n + c_{n-1} t_{k-1}^{n-1} + \cdots + c_1 t_{k-n-1} + c_0, \qquad (6)$$

where $\hat{y}_k$ is the estimation of y_k at sample k, and $t_k = kT, \forall k = 1, 2, \cdots, n$ is the time sample. The coefficients $c_n, \cdots, c_0$ are chosen to minimize $\|y_k - \hat{y}_k\|^2$. Based on the polynomial fitting, the estimated velocity can be found as:

$$\hat{\dot{y}}_k = n c_n t_k^{n-1} + (n-1) c_{n-1} t_{k-1}^{n-2} + \cdots + c_1. \qquad (7)$$

The polynomial fitting can be extended to the m most recent data points:

$$\hat{\boldsymbol{y}} = \boldsymbol{T}\boldsymbol{c} \qquad (8)$$

$$\begin{bmatrix} \hat{y}_k \\ \hat{y}_{k-1} \\ \vdots \\ \hat{y}_{k-m} \end{bmatrix} = \begin{bmatrix} t_k^n & t_{k-1}^{n-1} & \cdots & t_{k-n-1}^{k-n-1} & 1 \\ t_{k-1}^n & t_{k-2}^{n-1} & \cdots & t_{k-n-2}^{k-n-1} & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ t_{k-m}^n & t_{k-m-1}^{n-1} & \cdots & t_{k-m-n-1}^{k-n-1} & 1 \end{bmatrix} \begin{bmatrix} c_n \\ c_{n-1} \\ \vdots \\ c_0 \end{bmatrix}.$$

The coefficient vector $\boldsymbol{c}$ that minimizes $\|\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k\|^2$ is given by:

$$\boldsymbol{c} = (\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{T}^T\boldsymbol{y}. \qquad (9)$$

The challenge of the polynomial fitting is to find the appropriate order of the polynomial and number of previous points to use. The higher the order and number of points the better is the accuracy of the fitting, however the lag-time and the computation time will increase.

In this work is implemented a third order polynomial over four data points and the quality of the estimation is rather poor (see the red curves in Fig.5(g) - Fig.5(i)). This is due to the fact that the fitting is performed over data with the same information.

This problem leads once again to the concept of EE. In the same charts, it is presented the estimation using polynomial fitting over the EE considering three and five skipped EE (blue and cyan curves, respectively) providing better results. The estimation will improve with the quantity of skipped events. However if they are compared with the estimation obtained with FD with EE, (see Fig.5(d) - Fig.5(f)), it can be observed that they have worse noise to signal ratio.

The poor efficiency of the estimation based on the polynomial fitting, is that it requires of the analysis of the continuity and trend of the higher order derivatives when consecutive data points do not add new information, leading to the use of spline or nurbs approaches which require much computation efforts.

4.4 Estimation with Kalman Filter

The Kalman Filter (KF) is perhaps the most used technique for the estimation of state variables of a system. This recursive filter, which encloses the least square estimator proposed long ago by F. Gauss (Sorenson (1970)) was first published in 1960 (Kalman (1960)). Since then, it has been implemented in many applications and several formulations were derived from the former one (e.g. Extended KF (Einicke and White (1999), Ligorio and Sabatini (2013), Perala and Piché (2007), Quine (2006)) , Unscended KF (Julier and Uhlmann (2004),LaViola (2003)), among others), but keeping the core concept of a two step predictor-corrector type estimator that minimizes the estimated error covariance.

The KF tries to estimate the state $\boldsymbol{x} \in \mathcal{R}^n$ of a discrete-time controlled process that is governed by a linear stochastic difference equation with a measurement $\boldsymbol{z} \in \mathcal{R}^m$, given by the following general expressions:

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{w}_{k-1}, \qquad (10)$$

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k, \qquad (11)$$

where $\boldsymbol{u}_k$ is the control input and k is the time sample. The random variables $\boldsymbol{w}_{k-1}$ and $\boldsymbol{v}_k$ represent the process and measurement noise respectively, and it is assumed that they are independent with normal probability distributions, (i.e. $p(w_k) \sim N(0, \boldsymbol{Q}_k)$ and $p(v_k) \sim N(0, \boldsymbol{R}_k)$, where $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are the corresponding covariance matrices).

The estimation problem is solved recursively by implementing the following sequential relations at each sample time (Welch and Bishop (2001)):

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{B}\boldsymbol{u}_k, \qquad (12)$$

$$\boldsymbol{P}_k^- = \boldsymbol{A}\boldsymbol{P}_{k-1}\boldsymbol{A}^T + \boldsymbol{Q}, \qquad (13)$$

$$\boldsymbol{K}_k = \frac{\boldsymbol{P}_k^- \boldsymbol{H}^T}{\boldsymbol{H}\boldsymbol{P}_k^- \boldsymbol{H}^T + \boldsymbol{R}}, \qquad (14)$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_k^-), \qquad (15)$$

$$\boldsymbol{P}_k = (\boldsymbol{1} - \boldsymbol{K}_k\boldsymbol{H})\boldsymbol{P}_k^-, \qquad (16)$$

For the particular case of velocity and acceleration estimation from position data, the process model can be easily derived by considering the Taylor series expansion of each state variable and assuming that the plant noise is the discrete representation of the remainder of the Taylor series (Zhou et al. (2008), Sabatini (2003), Shaowei and

Shanming (2012), Belanger (1992)). Therefore, the process and measurement models are given by:

$$\begin{bmatrix}\hat{y}_k\\ \hat{\dot{y}}_k\\ \hat{\ddot{y}}_k\end{bmatrix} = \begin{bmatrix}1 & T & T^2/2\\ 0 & 1 & T\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}y_k\\ \dot{y}_k\\ \ddot{y}_k\end{bmatrix} + \begin{bmatrix}T^3/6\\ T^2/2\\ T\end{bmatrix}, \tag{17}$$

$$y_{k+1} = \hat{y}_k + v_k, \tag{18}$$

and the covariance matrices are:

$$\boldsymbol{Q} = q\begin{bmatrix}T^5/20 & T^4/8 & T^3/6\\ T^4/8 & T^3/6 & T^2/2\\ T^3/6 & T^2/2 & T\end{bmatrix}, \tag{19}$$

$$R = \frac{e_r^2/4 + 2r}{3}, \tag{20}$$

where $r = E[\epsilon_i]$ and ϵ_i is the error associated to the measurement. A comprehensive study of the stochastic components of the model is presented in (Belanger (1992)).

The position data is filtered with the KF and is presented in red in Fig.5(j) $-$ Fig.5(ℓ).

As it can be observed, the estimation can't keep its performance along all the range of work. In order to enhance the performance of the KF many authors propose to modify the noise covariance matrices considering the last estimation (Shaowei and Shanming (2012), Zhou et al. (2008)), or rescaling the gain of the filter (Duan et al. (2003)) turning it into an adaptive filter. In this work, it is proposed the following adaptive law:

$$Ra_k = \frac{10R}{1 + \hat{\dot{y}}_{k-1}^2}. \tag{21}$$

This adaptation law allows to preserve the performance of the estimation along the full range of work providing better result as it can be observed on the blue curves in Fig.5(j) - Fig.5(ℓ).

Even though this formulation gives a suitable estimation it may not be appropriate for embedded real time system, since the third order nature of the expressions requires of time consuming computation. Therefore, some authors propose to use a reduced order system (Shaowei and Shanming (2012), Sabatini (2003)). More precisely, in (Shaowei and Shanming (2012)) is implemented the following first order model, which leads to a Single Dimensional Kalman Filter (SDKF):

$$\theta_k = \theta_{k-1} + w_{k-1}, \tag{22}$$

$$\theta_{k+1} = \theta_k + v_k. \tag{23}$$

The implementation of the SDFK leads to two new approaches, one that implements the KF to estimate the position and the other to estimate the velocity.

Kalman Filter for position estimation In this approach, the SDKF is used to estimate the real position, thus $\theta_k = y_k$. The position estimation with the SDKF provides a smoother signal, as it can be observed in Fig. 4. From this estimated position the velocity can be estimated by any of the previous techniques mentioned.

In particular, the invariant sampled time result from the SDKF makes it suitable to implement the Savitzky-Golay (SG) method. This method provides a polynomial fitting with constant coefficients obtained from specific tables for digital signals with invariant sample time (Madden (1978), Gorry (1990), Schafer (2011)). Therefore, there is

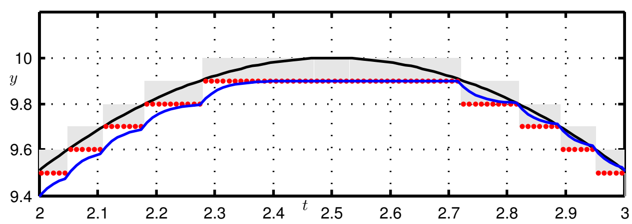


Fig. 4. Position estimation using SDKF. The real position is in black, the measurements from the sensor is in red, and the position estimation implementing SDKF is in blue.

no need to obtain the coefficients for each new incoming data reducing the computation time. Some properties of the SG method as digital differentiator are presented in (Luo et al. (2005))

The SG method is implemented with a fourth order polynomial fitting with ten points and the estimation is presented in magenta in Fig.5(j) - Fig.5(ℓ).

As it can be seen, the velocity estimation has improved comparing it with the results obtained with the FD method with the raw data (see Fig.5(d)) and the polynomial fitting with the EE (see Fig.5(g)).

Kalman Filter for velocity estimation In this approach, $\theta_k = \tilde{\dot{y}}_k$, where $\tilde{\dot{y}}_k$ is obtained from the FD method over EE. For this particular example it is implemented the FD method skipping two events.

The estimated velocity with the SDKF provides a smoother estimation than the FD method with practical no time-lag, as it can be appreciated in cyan in Fig.5(j) - Fig.5(ℓ).

4.5 Sliding Mode Differentiation

The main problem with differentiation is to combine accuracy and robustness in face of possible measurement errors and input noises. In (Levant (1998)) is proposed a robust first-order differentiator based on the sliding mode technique which features finite-time convergence and robustness against the measurement noise. This technique is implemented in (Damiano et al. (2004)) for the estimation of velocity and acceleration for a DC-drive motor controller.

The discrete formulation of the differentiator is given by:

$$h_k = -\lambda|z_k - x_k|^{1/2} sign(z_k - x_k) + w_k, \tag{24}$$

$$w_k = w_{k-1} - T\alpha sign(z_{k-1} - x_{k-1}), \tag{25}$$

$$z_k = z_{k-1} + Th_{k-1}, \tag{26}$$

where x_k is the measurable signal, $w_0 = z_0 = 0$, and α and λ are set according to the following inequalities:

$$\alpha > X_{dd}, \tag{27}$$

$$\lambda \geq 2\sqrt{X_{dd}\frac{\alpha + X_{dd}}{\alpha - X_{dd}}}, \tag{28}$$

and $X_{dd} > 0$ is the upperbound for the magnitude of the second derivative.

The Sliding Mode Differentiator (SMD) is implemented with real data and the velocity estimation is presented in red in Fig.5(m). The parameter λ and α are selected to provide the best result possible for low-speed range (see

Fig.5(m) - Fig.5(o)). However, the estimation deteriorates for higher velocities (see Fig.5(n) and Fig.5(o)).

In order to enhance the performance of the estimation, it is considered a variable acceleration upperbound:

$$Xa_k = X_{dd}\left|\frac{\hat{\dot{y}}_k - \hat{\dot{y}}_{k-1}}{T}\right|. \quad (29)$$

This small modification provides of adaptability to the method and the result improves along all the range of work as it can be observed in blue in Fig.5(o), however the estimation for the low speed range becomes more noisy (compare it with the red curve).

5. ACCELERATION ESTIMATION

In order to estimate the acceleration there are two possible approaches regarding the model implemented, i.e. the acceleration can be thought as the second time-derivative of the position or it can also be though as the first time-derivative of the velocity. The latter one, leads to the successive implementation of any of the first order methods presented in the previous section.

5.1 Finite Difference and Filtering

The most common method implemented to estimate the acceleration is also the finite difference method. Its formulation for the second order is given by (henceforth FD2):

$$\ddot{y}_k = \frac{y_k - 2y_{k-1} + y_{k-2}}{T^2}. \quad (30)$$

As it was shown in the velocity estimation section, working with FD over the raw data, doesn't lead to suitable results. Therefore, for the acceleration estimation it is considered only EE.

In Fig.8(a)-Fig.8(c) are presented the estimation using only FD2 over the EE considering two (red curve) and five (blue curve) skipped events. This formulation leads to very poor estimation. Therefore, the implementation of a LPF is mandatory for the acceleration estimation. As it can be seen in the green and cyan curves (filtered signal of the FD2 over EE with two and five skipped events, respectively), the filter improves the estimation but still it is not accurate enough for any application.

5.2 Kalman Filter

In this section is implemented several combinations of the KF to estimate the acceleration.

The raw position data is filtered implementing a SDKF as presented in Fig.4, and then it is submitted to a FD2 operation. The result is presented in red in Fig.8(d) - Fig.8(f). As it can be seen, the estimation is poor, especially in the low-speed range, where the estimated signal is far from resemblance the real acceleration signal. Therefore, a second SDKF is implemented over the estimated signal, whose result is presented in blue in the same figures. Even though, the result improves considerably for high velocities, the estimation for the low-speed range still is unacceptable for any purpose.

As it was already presented, the filtering improves by considering only those states of the encoder that represent a change. Therefore, the FD2 estimation is implemented on the EE and then the signal is filtered using a SDKF. The result shows that the estimation improves (see cyan curve in Fig.8(d) - Fig.8(f)), though the signal still is far to be used in a control loop. Thus, another SDKF is implemented over this first estimated data, and the new estimation is presented in green in the same charts. The estimation improves drastically its performance in the low-speed range and in the high-speed range, though time-lag appears in the estimated data.

The SDKF as it was mentioned is a simplified formulation that saves computation efforts, by implementing a first order model. This simplification of the model reduces the performance of the estimation. This can be seen by simple comparison between the results obtained with the SDKF and the KF with the third order model presented in magenta in Fig.8(d) - Fig.8(f). This last result shows much better performance for all the range of velocity, despite the fact that for high velocity there is a considerably time-lag. However, if the adaptation rule (21) for the noise covariance matrix is implemented, the time-lag reduces considerably and the overall performance of the estimation enhances, as it can be observed in the yellow curve.

5.3 Sliding Mode Differentiation

Based on the estimator proposed in (Damiano et al. (2004)), the acceleration is estimated as a successive derivation using a first order SMD. The result is presented in red in Fig.8(g)-Fig.8(i). It can be observed that the result is not as promising as with the velocity estimation. The best estimation for the low-speed range has an unsatisfactory noise to signal ratio estimation, and furthermore for higher velocities, the acceleration estimation presents severe deformation and time-lag. Considering the adaptive behavior on the upper-bound parameter X_{dd} of the filter, it can be seen that the estimation does not improve (see blue curve), and it remains practically the same.

5.4 Integration within a Control Loop

In general, the second order time derivative of the position is the most implemented relation for the estimation of the acceleration. Even though, the implementation of differentiators inherently carries with them noise amplification in the high frequency range, therefore its performance will be always limited.

It must be recalled that the differentiation is not the only relation that links these two variables, there is also the integration. Therefore, some authors have developed approaches based on the implementation of integrators within a closed-loop. One example can be found in (Tilli and Montanari (2001)), where feedforward with a integrator feedback is proposed as velocity and acceleration estimator.

In this work two different approaches are analyzed.

The first approach considers the estimation problem as a classical position control of a ordinary double integrator type system, as presented in Fig.6. This approach has as many alternatives as possible controllers. A comprehensive study and comparison between several control strategies

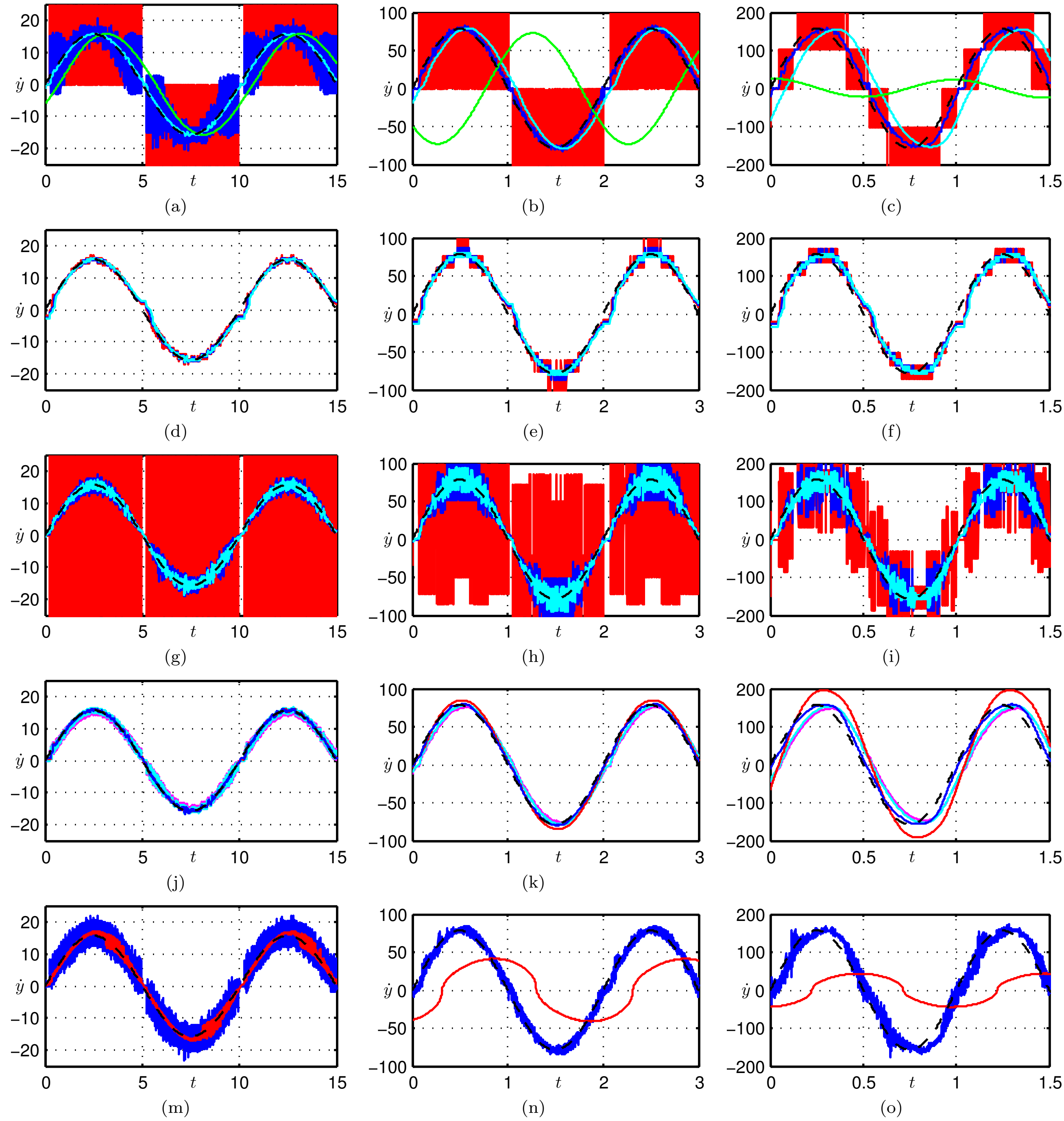


Fig. 5. Velocity estimation with different methods for different velocity ranges: first column $f = 0.1$Hz, second column $f = 0.5$Hz. and third column $f = 1$Hz.

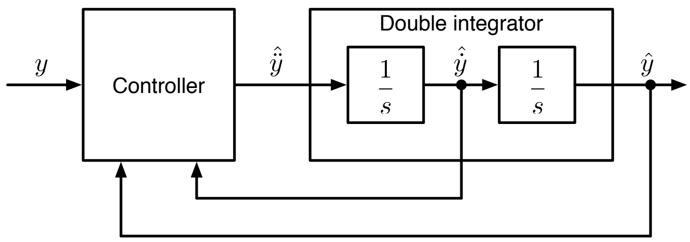


Fig. 6. Acceleration estimation based on a controlled double integrator type system.

for the classical double integrator type system is presented in (Rao and Bernstein (2001)), where it is shown that the classical PD controller exhibits good robustness.

In particular, in this survey is considered the algorithm presented in (Lee and Song (2001)), which implements the following PD controller:

$$\hat{\ddot{y}} = K_p(y - \hat{y}) - K_d \frac{d\hat{y}}{dt} \tag{31}$$

that tries to force to the estimated position $\hat{y}$ to follow the actual displacement y. Based on the later relation, the following transfer function can be obtained:

$$\frac{\hat{\ddot{y}}}{\ddot{y}} = \frac{K_p}{s^2 + K_d s + K_p}, \tag{32}$$

where K_p defines the bandwidth of the acceleration estimation.

The result found implementing this formulation presents a suitable estimation for the low-range acceleration, however it also posses a lag-time for higher accelerations, as it can be observed in blue in Fig.8(j) - Fig.8(ℓ).

In (Shaowei and Shanming (2012)) it is presented an algorithm based on the Phase Locked Loop structure (PLL) in order to obtain the estimation of the acceleration by integral calculation using the previously estimated velocity. The overall concept is presented in Fig.7, where $\hat{\dot{y}}$ is the SDKF filtered velocity and $\dot{y}'$ is the velocity estimated through integral computation of the estimated acceleration $\hat{\ddot{y}}$ through the PLL method. The parameters τ, K_p and K_i are chosen to satisfy the tracking performance of the velocity loop.

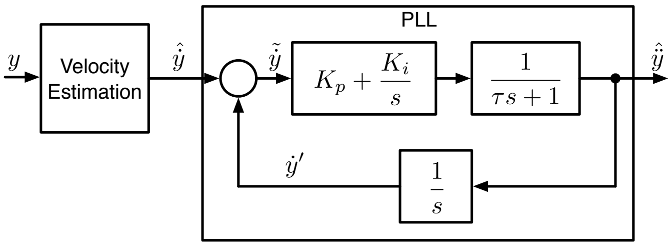


Fig. 7. Acceleration estimation based on a PLL structure.

The estimated acceleration implementing this approach presents noise in the low acceleration region, even though the time-lag for high acceleration is improved in comparison with the double-integrator type controller (see the red curve in red in Fig.8(j) - Fig.8(ℓ)).

6. KPI ANALYSIS

In Table.1 is summarized the results from the performance analysis of all the methods presented in this work using the KPI defined previously. The table is divided into two parts. The first part (the first fifteen rows) shows the results of the KPI obtained for the velocity estimation, and the second part (the last fourteen rows) shows the results of the KPI obtained for the acceleration estimation.

For the velocity estimation, the KPI analysis confirms that the methods based on the FD over EE with the skipping option and the methods based on KF has better performance over all the frequency range analyzed.

In particular, the KPI analysis shows that the KF implementing a third order model with adaptive covariance matrix (KF $3^{rd}adap$) has the best performance of all the methods implemented.

For the acceleration estimation, the KPI analysis also shows that the KF $3^{rd}adap$ formulation presents better performance for all the frequency range. However, the KPI analysis also demonstrates that the second order integrator with a PD controller method (2^{nd}Int+PD), presents the best performance of all the methods analyzed for the low-speed range ($f = 0.1$Hz).

7. CONCLUSION AND DISCUSSION

In this work the problem of the velocity and acceleration estimation from position data measured with a general digital position sensor from a linear actuator was assessed.

The main issue regarding a digital encoder is its finite resolution. This gap between two consecutive data from the digital sensor derives in the impossibility of detecting small changes of the measured variable. Therefore, the estimation of velocity and acceleration will be mostly affected in the low-speed range.

Several methods were introduced and their results were compared.

It was shown that independently of the method implemented for the estimation it is clearly settled that a static method provides an appropriate solution for reduced bandwidth applications, even though with small variations it is easy to convert the method into an adaptable solution enhancing the overall performance and increasing its bandwidth.

The estimation of the acceleration can be addressed as a first order time-derivative relation of the velocity or a a second order time-derivative relation of the position. The first one implies that a successive implementation of a first order derivation or integration must be carried on which may lead to modeling simplification, such as the SDKF approach. This simplification in the model has the drawback that the performance will also be limited. If a complete model is implemented the performance is enhanced with the counterpart that the computation effort rises and therefore this approach could not be embedded in simple RT hardware.

From all the methods implemented it was found that the KF with a third order model with an adaptable covariance matrix provides better result for all the velocity range analyzed.

REFERENCES

Astrom, K. and Bernhardsson, B. (2002). Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 2, 2011–2016. IEEE. doi:10.1109/CDC.2002.1184824.

Baran, E.a., Golubovic, E., and Sabanovic, A. (2010). A new functional observer to estimate velocity, acceleration and disturbance for motion control systems. *2010 IEEE International Symposium on Industrial Electronics*, 384–389. doi:10.1109/ISIE.2010.5637690.

Belanger, P. (1992). Estimation of angular velocity and acceleration from shaft encoder measurements. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, 585–592. IEEE Comput. Soc. Press. doi:10.1109/ROBOT.1992.220228.

Brown, R., Schneider, S., and Mulligan, M. (1992). Analysis of algorithms for velocity estimation from discrete position versus time data. *IEEE Transactions on Industrial Electronics*, 39(1), 11–19. doi:10.1109/41.121906.

Cea, M.G. and Goodwin, G.C. (2012). Event based sampling in non-linear filtering. *Control Engineering Practice*, 20(10), 963–971. doi: 10.1016/j.conengprac.2011.11.008.

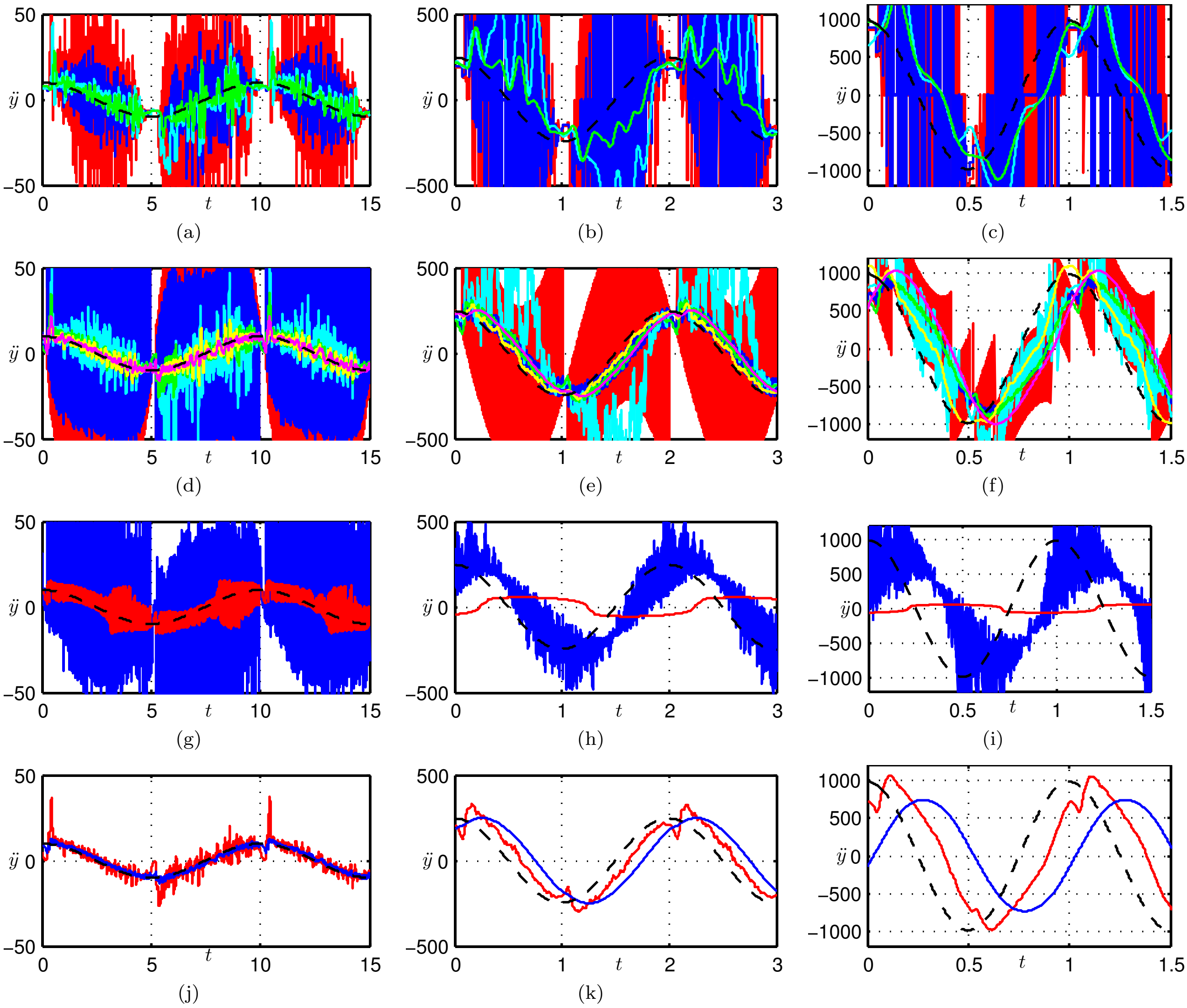


Fig. 8. Aceleration estimation with different methods for different acceleration ranges: first column $f = 0.1$Hz, second column $f = 0.5$Hz. and third column $f = 1$Hz.

Choi, H.H. and Jung, J.W. (2011). Fuzzy speed control with an acceleration observer for a permanent magnet synchronous motor. *Nonlinear Dynamics*, 67(3), 1717–1727. doi:10.1007/s11071-011-0099-y.

Damiano, A., Gatto, G., Marongiu, I., and Pisano, A. (2004). Second-Order Sliding-Mode Control of DC Drives. *IEEE Transactions on Industrial Electronics*, 51(2), 364–373. doi:10.1109/TIE.2004.825268.

Duan, Z., Han, C., and Dang, H. (2003). An Adaptive Kalman Filter with Dynamic Rescaling of Process Noise. In *Information Fusion, 2003. Proceedings of the Sixth International Conference of*, 3, 1310–1315. IEEE, Cairns, Queensland, Australia. doi:10.1109/ICIF.2003.177389.

Einicke, G. and White, L. (1999). Robust Extended Kalman Filtering. *IEEE Transactions on Signal Processing*, 47(9), 2596–2599.

Emaru, T., Imagawa, K., Hoshino, Y., and Kobayashi, Y. (2009). Velocity and Acceleration Estimation by a Nonlinear Filter Based on Sliding Mode and Application to Control System. *Journal of Robotics and Mechatronics*, 21(5), 1–2.

Goodwin, G.C., Aguero, J.C., Cea Garridos, M.E., Salgado, M.E., and Yuz, J.I. (2013). Sampling and Sampled-Data Models: The Interface Between the Continuous World and Digital Algorithms. *IEEE Control Systems*, 33(5), 34–53. doi:10.1109/MCS.2013.2270403.

Gorry, P.a. (1990). General least-squares smoothing and differentiation by the convolution (Savitzky-Golay) method. *Analytical Chemistry*, 62(6), 570–573. doi:10.1021/ac00205a007.

Han, J., He, Y., and Xu, W. (2007). Angular acceleration estimation and feedback control: An experimental investigation. *Mechatronics*, 17(9), 524–532. doi:10.1016/j.mechatronics.2007.05.006.

Julier, S. and Uhlmann, J. (2004). Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, 92(3), 401–422. doi:10.1109/JPROC.2003.823141.

Kalman, R.E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transaction of the ASME - Journal of Basic Engineering*, 82(Series D), 35–45.

LaViola, J. (2003). A comparison of unscented and extended Kalman filtering for estimating quaternion motion. In *Proceedings of the 2003 American Control*

Conference, 2003., volume 3, 2435–2440. IEEE. doi:10.1109/ACC.2003.1243440.

Lee, S.h. and Song, J.b. (2001). Acceleration estimator for low-velocity and low-acceleration regions based on encoder position data. *IEEE/ASME Transactions on Mechatronics*, 6(1), 58–64. doi:10.1109/3516.914392.

Levant, A. (1998). Robust exact differentiation via sliding mode technique. *Automatica*, 34(3), 379–384. doi:10.1016/S0005-1098(97)00209-4.

Ligorio, G. and Sabatini, A.M. (2013). Extended Kalman filter-based methods for pose estimation using visual, inertial and magnetic sensors: comparative analysis and performance evaluation. *Sensors (Basel, Switzerland)*, 13(2), 1919–41. doi:10.3390/s130201919.

Liu, G. (2002). On velocity estimation using position measurements. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, volume 2, 1115–1120. American Automatic Control Council, Anchorage, Alaska, USA. doi:10.1109/ACC.2002.1023168.

Luo, J., Ying, K., He, P., and Bai, J. (2005). Properties of SavitzkyGolay digital differentiators. *Digital Signal Processing*, 15(2), 122–136. doi:10.1016/j.dsp.2004.09.008.

Madden, H.H. (1978). Comments on the Savitzky-Golay convolution method for least-squares-fit smoothing and differentiation of digital data. *Analytical Chemistry*, 50(9), 1383–1386. doi:10.1021/ac50031a048.

Merry, R., van de Molengraft, M., and Steinbuch, M. (2010). Velocity and acceleration estimation for optical incremental encoders. *Mechatronics*, 20(1), 20–26. doi:10.1016/j.mechatronics.2009.06.010.

Ohmae, T., Matsuda, T., Kamiyama, K., and Tachikawa, M. (1982). A Microprocessor-Controlled High-Accuracy Wide-Range Speed Regulator for Motor Drives. *IEEE Transactions on Industrial Electronics*, IE-29(3), 207–211. doi:10.1109/TIE.1982.356665.

Ovaska, S. and Valiviita, S. (1998). Angular acceleration measurement: a review. *IEEE Transactions on Instrumentation and Measurement*, 47(5), 1211–1217. doi:10.1109/19.746585.

Perala, T. and Piché, R. (2007). Robust extended Kalman filtering in hybrid positioning applications. *Positioning, Navigation and Communication ...*, 2007, 55–63. doi:10.1109/WPNC.2007.353613.

Quine, B.M. (2006). A derivative-free implementation of the extended Kalman filter. *Automatica*, 42(11), 1927–1934. doi:10.1016/j.automatica.2006.06.013.

Rao, V. and Bernstein, D. (2001). Naive control of the double integrator. *IEEE Control Systems Magazine*, 21(5), 86–97. doi:10.1109/37.954521.

Sabatini, A.M. (2003). Real-time kalman filter applied to biomechanical data for state estimation and numerical differentiation. *Medical & Biological Engineering & Computing*, 41(1), 2–10. doi:10.1007/BF02343532.

Schafer, R. (2011). What Is a Savitzky-Golay Filter? [Lecture Notes]. *IEEE Signal Processing Magazine*, 28(4), 111–117. doi:10.1109/MSP.2011.941097.

Shaowei, W. and Shanming, W. (2012). Velocity and acceleration computations by single-dimensional Kalman filter with adaptive noise variance. *Przegld Elektrotechniczny*, (2), 283–287.

Song, Y., Gao, H., Zhang, S., Tian, Y., and Wang, D. (2009). The analysis and design of low velocity estimation based on observer. In *2009 IEEE International Conference on Automation and Logistics*, August, 766–771. IEEE, Shenyang, China. doi:10.1109/ICAL.2009.5262819.

Sorenson, H.W. (1970). Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum*, 7(7), 63–68. doi:10.1109/MSPEC.1970.5213471.

Tilli, A. and Montanari, M. (2001). A Low-Noise Estimator of Angular Speed and Acceleration from Shaft Encoder Measurements. *Automatika*, 42(3-4), 169–176.

Tsuji, T., Mizuochi, M., Nishi, H., and Ohnishi, K. (2005). A velocity measurement method for acceleration control. In *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.*, 6 pp. IEEE. doi:10.1109/IECON.2005.1569202.

Welch, G. and Bishop, G. (2001). An Introduction to the Kalman Filter. In L. Harrison (ed.), *28th International Conference on Computer Graphics and Interactive Techniques -SIGGRAPH2001*, 1–81. ACM, Inc, North Carolina.

Zhou, J., Shen, X., Petriu, E.M., and Georganas, N.D. (2008). Linear velocity and acceleration estimation of 3 DOF haptic interfaces. In *2008 IEEE International Workshop on Haptic Audio visual Environments and Games*, 137–142. IEEE. doi:10.1109/HAVE.2008.4685313.

Table .1. KPI comparison

Method	$f = 0.1$Hz			$f = 0.5$Hz			$f = 1.0$Hz			Figure
	KPI_1	KPI_2	KPI_3	KPI_1	KPI_2	KPI_3	KPI_1	KPI_2	KPI_3	
FD + LPF $f_c = 10$Hz	14.96	21.07	0.53	21.29	3.9	0.15	376.31	1.44	0.24	blue, 5(a)-5(c)
FD +LPF $f_c = 1$Hz	0.45	8.46	0.17	205.06	4.1	0.46	3267.59	4.2	0.72	cyan, 5(a)-5(c)
FD +LPF $f_c = 0.1$Hz	22.54	33.44	1.13	10313.82	11.07	2.01	17724.96	2.3	1.14	green, 5(a)-5(c)
FD with EE (skip=1)	1.11	10.63	0.27	55.59	2.51	0.21	480.63	2.04	0.26	red, 5(d)-5(f)
FD with EE (skip=5)	1.26	12.7	0.32	51.65	3	0.24	448.86	2.24	0.26	blue, 5(d)-5(f)
FD with EE (skip=10)	2.05	15.4	0.41	74.49	3.59	0.29	593.4	2.63	0.31	cyan, 5(d)-5(f)
Pol. with raw data	2800.9	198.89	3.22	6886.17	16.62	1.57	5592.49	6.52	0.66	red, 5(g)-5(i)
Pol. with EE (skip=3)	1.79	5.19	0.17	286.79	1.81	0.25	1077.8	1.13	0.28	blue, 5(g)-5(i)
Pol. with EE (skip=5)	0.8	4.81	0.15	48.98	1.82	0.16	396.52	0.94	0.2	cyan, 5(g)-5(i)
SDKF(y) + SG	1	3.96	0.17	82.63	2.96	0.29	1342.29	2.37	0.45	magenta, 5(j)-5(ℓ)
FD(EE) + SDKF	0.66	3.43	0.16	64.98	2.78	0.26	1113.3	2.16	0.42	cyan, 5(j)-5(ℓ)
KF 3^{rd}	0.1	4.46	0.07	41.53	1.22	0.17	2189.63	2.75	0.54	red, 5(j)-5(ℓ)
KF 3^{rd}*adap.*	0.18	4.41	0.08	11.03	1.06	0.11	229.6	0.51	0.16	blue, 5(j)-5(ℓ)
SMD	0.71	4.42	0.15	2346.42	8.81	1.15	14212.64	3.28	1.04	red, 5(m)-5(o)
SMD adap.	6.3	9.41	0.33	29.3	1.99	0.15	548.75	1.81	0.25	blue, 5(m)-5(o)
FD2(EE)	1274.75	456.11	9.44	3647951.5	578.71	12.13	45893518.83	95.46	11.25	red, 8(a)-8(c)
FD2(EE)	129.53	114.96	3.24	267806.51	107.11	4.16	2088631.79	22.83	1.83	blue, 8(a)-8(c)
FD2(EE) +LPF	137.29	39.44	2.42	116178.24	13.96	2.61	391160.92	3.9	1.11	green, 8(a)-8(c)
FD2(EE) +LPF	35.87	43.69	1.77	18608.5	7.86	1.22	252869.78	4.11	0.91	cyan, 8(a)-8(c)
SDKF(y)+FD2	89501.13	5804.06	73.81	202674.65	45.33	3.75	388375.48	8.77	0.91	red, 8(d)-8(f)
SDKF(y)+FD2+SDKF	881.03	274	7.97	5360.59	9.47	0.68	242745.68	5.05	0.88	blue, 8(d)-8(f)
SDKF(EE) + FD2	129.53	114.96	3.24	267806.51	107.11	4.16	2088631.79	22.83	1.83	cyan, 8(d)-8(f)
SDKF(EE) + FD2 +SDKF	29.59	22.54	1.34	14992.37	16.62	1.14	122605.79	3.38	0.62	green, 8(d)-8(f)
KF 3^{rd}	3.35	14.96	0.46	6319.67	7.77	0.8	421449.17	6.73	1.2	magenta, 8(d)-8(f)
KF 3^{rd}*adap.*	14.76	16.39	1	1444.55	3.78	0.36	82299.66	3.17	0.52	yellow, 8(d)-8(f)
SMD	21.26	36.85	0.86	44491.77	4.87	1.2	614764.04	1.42	1.09	red, 8(g)-8(i)
SMD adap.	697.97	289.69	8.56	17087.03	10.27	1.07	650052.41	8.97	1.44	blue, 8(g)-8(i)
2^{nd} Int + PD	1.7	12.12	0.45	17953.67	13	1.31	959689.94	7.24	1.52	blue, 8(j)-8(ℓ)
$\hat{y}$+PLL	20.33	25.08	0.82	6542.59	7.31	0.78	287468.49	5.47	1	red, 8(j)-8(ℓ)