Algorithm for Velocity Estimation in a Multivariable Motion Sensor

Federico Mazzoli[1](✉), Davide Alghisi[2], and Vittorio Ferrari[1]

[1] Department of Information Engineering, University of Brescia, Via Branze 38, Brescia, Italy
f.mazzoli002@unibs.it
[2] Gefran SpA, Via Cave 11, Provaglio d'Iseo, BS, Italy

Abstract. This paper proposes an algorithm for velocity estimation using the position and acceleration signals obtained respectively from a resistive potentiometric displacement sensor and a MEMS accelerometer. The algorithm is composed of two processing chains that independently estimate velocity starting from position and acceleration signals. Velocity estimation from position is obtained through an adaptive windowing differentiator while the estimation from acceleration is based on a leaky integrator low-pass filter. Such two estimations are fused together by means of a tailored weighted average. The proposed algorithm is first simulated in MATLAB and then experimentally implemented and tested. Both simulations and experimental results show that velocity estimation given by the fusion of the outputs of the two processing chains has a lower estimation error compared to the output of each single chain.

Keywords: Real-time velocity estimation · Multivariable sensor · Motion control · Embedded systems

1 Introduction

Velocity estimation obtained from signals provided by motion sensors, without specifically using velocity sensors, plays a crucial role within the industrial field e.g., for servomotor controllers in closed-loop control systems and motion control of mechanical machines, including robot manipulators [1]. A basic approach involves differentiating the signal from a position sensor to obtain a velocity signal. Nevertheless, the high-frequency noise included in such position signal is amplified by the differentiation [2]. The width of the differentiation window rules the trade-off between the estimation noise and its delay. Several methods have been developed to estimate velocity from position data while reducing estimation errors and minimizing delay. For example, adaptive algorithms can be exploited to dynamically adjust the differentiation window [3].

Within the scopes of multivariable sensor and data fusion, there are techniques and methods that allow to estimate velocity by combining position and acceleration data coming from different sensing elements [4]. In this context, this work investigates a new technique for velocity estimation designed for a custom multivariable motion sensor, suitable for implementation in embedded systems based on simple microcontrollers

G. Di Francia and C. Di Natale (Eds.): AISEM 2022, LNEE 999, pp. 160–166, 2023.
https://doi.org/10.1007/978-3-031-25706-3_26

thanks to its low computational complexity. This paper is organized as follows: Sect. 2 presents the working principle and the block diagram of the proposed algorithm; Sect. 3 discusses the blocks and the simulation results, experimental results are shown in Sect. 4, while conclusions are reported in Sect. 5.

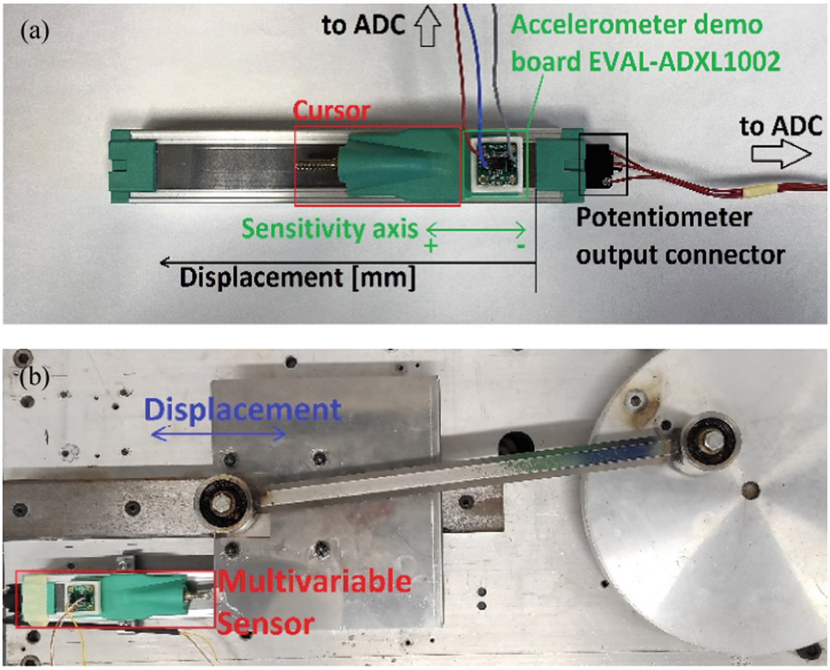

Fig. 1. (a) Multivariable motion sensor: resistive potentiometric displacement sensor (GEFRAN PK) equipped with an accelerometer (Analog Devices ADXL1002) mounted on the cursor. (b) Multivariable sensor linked to the rod-crank mechanism.

2 Multivariable Motion Sensor and Proposed Estimation Algorithm

A multivariable motion sensor prototype has been realized by combining a resistive rod-less potentiometric displacement sensor (Gefran PK) with electrical stroke length of 100 mm and a single-axis MEMS accelerometer (Analog Devices ADXL1002) with measurement range of $\pm$ 50 g coupled to the respective evaluation board EVAL-ADXL1002 fixed on the cursor. A mechanical adapter ensures the alignment of the accelerometer sensitive axis with the direction of the cursor displacement. A picture of the assembled multivariable sensor and the motion testing equipment is shown in Fig. 1.

The proposed estimation algorithm is composed of two processing chains acting on the position and acceleration signals coming from the two sensors, respectively, as shown in Fig. 2. The first chain is composed of a preprocessing section, consisting of the series of a median filter and a low-pass filter to reduce noise on the position signal *pos*. In this chain, velocity estimation is based on the differentiation of two position samples whose relative distance is dynamically adjusted according to the trend of the position signal. The block *End-Fit FOAW* implements an adaptive windowing differentiation to estimate velocity *vel* as shown is Eq. (1). The term T indicates the sampling period; y_k *and* $y_{k\text{-}N}$ represent two generic position samples while N indicates the number of samples in the differentiation window. The regulation criterion of N, as reported in [3], is based on searching the maximum value assumed by N such that the straight line ${}^{L}y_{k\text{-}i}$ passing through y_k and $y_{k\text{-}N}$ intersects each of the uncertainty bands d associated to the position samples inside the window as shown in Eq. (2). The straight line ${}^{L}y_{k\text{-}i}$ is defined in Eq. (3) while its parameters are reported in Eqs. (4) and (5).

$$vel = (y_k - y_{k-N})/NT \tag{1}$$

$$\left| y_{k-i} -^{L} y_{k-i} \right| \leq d \forall i \in \{1, \ldots, N\} \tag{2}$$

$${}^{L}y_{k-i} = a_n + b_n(k \ - \ i)T \tag{3}$$

$$a_n = (ky_{k-N} + (N-k)y_k)/N \tag{4}$$

$$b_n = (y_k - y_{k-N})/NT \tag{5}$$

The estimated velocity *vel* is median- and low-pass filtered in the postprocessing section to output velocity estimation vel_{p}.

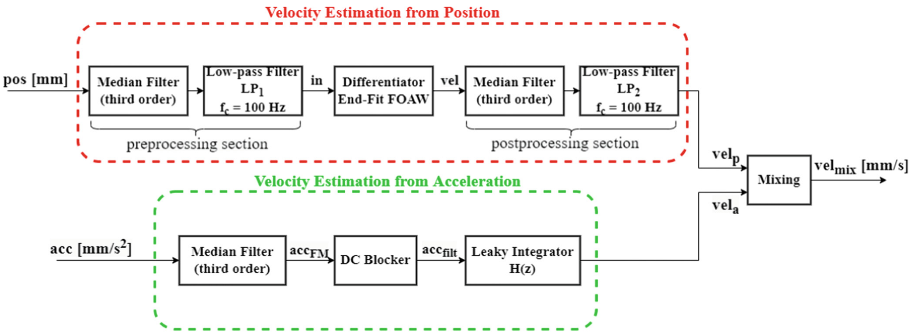

Fig. 2. Block diagram of the velocity estimation algorithm.

The second processing chain acts on the acceleration signal *acc*. First, the acceleration is preprocessed by a median filter to remove spikes. Then, the *DC Blocker* block removes the DC component of acceleration due to the 0-*g* offset of the accelerometer and to any residual misalignment between its sensitive axis and the cursor direction.

The DC component is recursively estimated by a third-order elliptical low-pass filter. Finally, the leaky integrator with transfer function *H(z)* reported in Eq. (6) integrates the acceleration acc_{filt} to obtain velocity estimation vel_{a}. The pole in *H(z)* is determined by the stability factor α. In order to obtain asymptotical stability, it must be 0< α <1. The term *T* stands for the sampling period of the acceleration signal *acc*.

$$H(z) = zT/(z - \alpha) \tag{6}$$

The overall velocity estimation vel_{mix} is obtained by the weighted average of vel_{p} and vel_{a}, calculated according to Eq. (7). To minimize the estimation error and its delay, two different pairs of weights are designed: the first pair is named *exponential weights* and is indicated as [w_{1e}, w_{2e}] in Eq. (8), while the second pair is named *linear weights* and is indicated as [w_{1l}, w_{2l}] in Eq. (9). From the rod-crank mechanism kinematics of Fig. 1b, the motion law is sinusoidal with fundamental frequency f_{rd} which can be set. Weights are calculated for a given frequency f_{rd}, depending on the tuning parameters *A, B, m, q*.

$$vel_{mix} = (w_1 vel_p + w_2 vel_a)/2 \tag{7}$$

$$[w_{1e} = 2e^{-frd+A}/(e^{-frd+A} + e^{frd-B}), w_{2e} = 2e^{frd-B}/(e^{-frd+A} + e^{frd-B})] \tag{8}$$

$$[w_{1l} = 2(-mf_{rd} + q)/q, w_{2l} = 2mf_{rd}/q] \tag{9}$$

3 Simulation Results

The proposed algorithm has been simulated in MATLAB by supplying position pos_{ref} and acceleration acc_{ref} input signals, both generated numerically according to the kinematic model of the rod-crank mechanism, as reported in Eqs. (10), (11), (12).

$$pos_{ref} = r(1 - cos(2\pi f_{rd} t) + \mu - (\mu^2 - sin^2(2\pi f_{rd} t))^{1/2} \tag{10}$$

$$vel_{ref} = 2\pi f_{rd} r(sin(2\pi f_{rd} t) + sin(4\pi f_{rd} t)/2(\mu^2 - sin^2(2\pi f_{rd} t))^{1/2})^{1/2} \tag{11}$$

$$acc_{ref} = (r(2\pi f_{rd})^2)(cos(2\pi f_{rd} t) + cos(4\pi f_{rd} t)/\mu) \tag{12}$$

The block parameters adopted in the algorithm are listed below. The cut-off frequency of the low-pass filters LP_1 and LP_2 is f_{c}=100 Hz; the uncertainty band of *End-Fit FOAW* differentiator is set to $d = 30\ \mu m$; the third-order elliptic low-pass filter relative to *DC Blocker* has a normalized bandwidth equal to 10^{-4} and the leaky integrator stability factor is α = 0.999. For the two weighted averaging methods, Fig. 3 compares the maximum relative errors of velocity estimations for different signal frequencies f_{rd} in the range [1, 10] Hz. The estimation errors of vel_p and vel_a respectively increase and decrease with f_{rd} because of the frequency responses of the differentiator *End-Fit FOAW* and the leaky integrator block. For each value of the test frequency f_{rd}, the error of the velocity estimation vel_{mix} is the lowest between those of vel_{p} and vel_{a}. In particular, the weighted average with *linear weights* typically produces the lower estimation error.

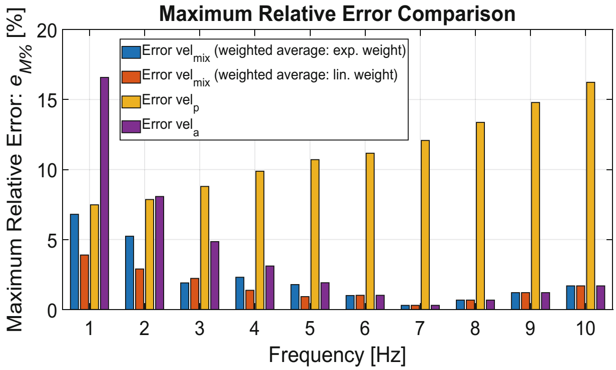

Fig. 3. Maximum relative error for different velocity estimations versus test frequency f_{rd} in the range [1, 10] Hz.

4 Experimental Results

Experimental results have been obtained by fixing the multivariable motion sensor to the rod-crank mechanism as shown in Fig. 1b. The instrumentation setup is reported in Fig. 4. Sensors are powered at 5 V by PL303-P Power Supply and the position and acceleration signals are sampled by means of a NI USB-6212 DAQ with sampling frequency $f_s =$ 1 kHz and a resolution of 16 bits. The algorithm has been implemented in MATLAB, using the same block parameters used in simulation.

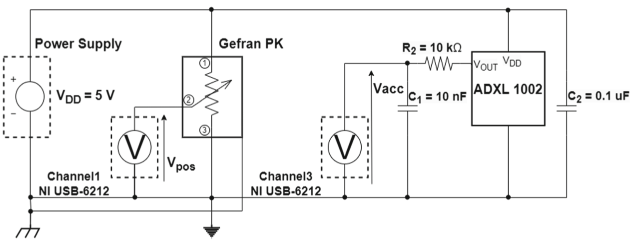

Fig. 4. Schematic instrumental setup, adopted to acquire position and acceleration signals.

Figure 5 shows the position and acceleration signals obtained when the multivariable sensor cursor is linked to the rod-crank mechanism, rotating at $f_{rd} = 2.8$ Hz. From these two signals the algorithm computes the velocity estimations vel_p, vel_a and vel_{mix}. The corresponding estimation errors, calculated as $|vel_{ref} - vel|$, are shown in Fig. 6.

5 Conclusions

An algorithm for velocity estimation from position and acceleration signals composed of two parallel processing chains producing respectively two velocity estimations, vel_{p} and vel_{a} was presented. Such estimations are fused through a weighted average operation to produce a single combined velocity estimation $\mathrm{vel}_{\mathrm{mix}}$. Both simulations and experimental results show the effectiveness of the proposed algorithm whereby the combined estimated velocity $\mathrm{vel}_{\mathrm{mix}}$ has lower errors with respect to both single estimations vel_{p} and vel_{a}.

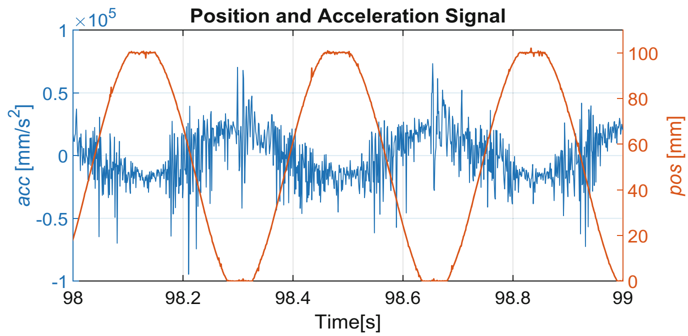

Fig. 5. Measured position and acceleration signals obtained from the multivariable motion sensor linked to the rod-crank mechanism at rotation frequency $f_{\mathrm{rd}} = 2.8$ Hz.

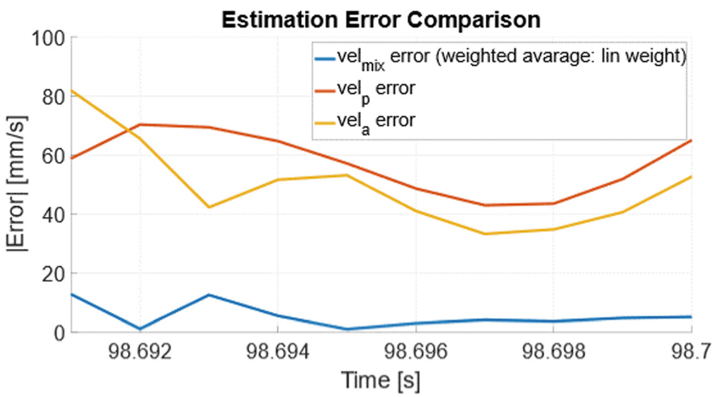

Fig. 6. Experimentally obtained estimation error of $\mathrm{vel}_{\mathrm{mix}}$ compared with those of $\mathrm{vel}_{\mathrm{p}}$ and $\mathrm{vel}_{\mathrm{a}}$.

References

1. Liu, G.: On velocity estimation using position measurements. In: Proceedings of the 2002 American Control Conference, pp. 1115–1120 (2002)
2. Brown, R.H., Schneider, S.C., Mulligan, M.G.: Analysis of algorithms for velocity estimation from discrete position versus time data. In: IEEE Transactions on Industrial Electronics, vol. 39, no.1, pp. 11–19 (1992)
3. Janabi-Sharifi, F., Hayward, V., Chen, C.J.: Discrete-time adaptive windowing for velocity estimation. In: IEEE Transactions on Control Systems Technology, vol. 8, no.6, pp. 1003–1009 (2000)
4. Zhu, W., Lamarche, T.: Velocity estimation by using position and acceleration sensors. In: IEEE Transactions on Industrial Electronics, vol. 54, no.5, pp. 2706–2715 (2007)