Lab Specialization Advanced Velocity Estimation

Yves Martin, *MSc in Robotics, EPFL,* Luca Rezzonico, *MSc in Robotics, EPFL,*

***Abstract*—During this project, several methods to estimate the velocity of the Haptic Paddle with a position sensor are implemented and compared. More advanced methods are compared to the basic Euler differentiation method with a low-pass filter, like First Order Adaptive Windowing, Levant's differentiation and the Kalman filter. Two experiments are conducted to compare these methods: the first one involves moving the paddle by the hand, the other on relies on observing the behavior of the Haptic Paddle against a virtual wall.**

***Index Terms*—Haptic Paddle, Velocity Estimation, Euler differentiation, FOAW, Levant's differentiation, Kalman filter, K-B Plot**

I. Introduction

PRECISE velocity estimation of a one degree of freedom joint is a challenge in robotics. The velocity is needed for many applications, such as velocity tracking, next position prediction, viscous friction compensation and so on. There is at the moment no silver bullet for velocity estimation. For this purpose several sensors can be used, the speed can for example be integrated from an accelerometer, measured directly from a gyroscope, or derived from a position sensor. This lab is about estimating the velocity out of a position sensor and comparing different methods. The position sensor used is a 4096 steps per revolution encoder. During previous experiments, velocity was estimated from the position of the haptic paddle using Euler differentiation and filtering. Since angular velocity is a key variable and no sensor is available to measure it directly, advanced velocity estimation techniques are introduced.

II. Euler Differentiation

This method computes the slope between two measured positions y_k and y_{k-1} over one sampling period T. The velocity is estimated using the following formula:

$$\hat{\dot{x}}_k = \frac{y_k - y_{k-1}}{T} \tag{1}$$

Since the sampling frequency of the Haptic Paddle is high (sampling period of $T = 350\mu s$), the noise is amplified. However it can be reduced using a low-pass filter. A cutoff frequency of $f_c = 50Hz$ is chosen to compromise between performance and delay.

III. First Order Adaptive Windowing

The first-order adaptive windowing method is shown to be optimal in the sense that it minimizes the velocity error variance while maximizing the accuracy of the estimates. In order to do so, the number of fitted past sensed positions to compute the velocity is chosen at each computation. This value, called windowing length n, is chosen in function of the last sensed positions variance around a linear fit. The higher the variance is, the smaller n will be. Due to this adaptability, when there is a sudden velocity change, FOAW shows reactive behavior, and when the joint is on a constant velocity, FOAW shows very precise results due to a large number of position points fitting. The implemented proposal is "end-fit first-order adaptive windowing"(end-fit-FOAW)$^{(2)}$.

$$\hat{v}_k = \frac{1}{n}\sum_{j=0}^{n-1} \hat{v}_{k-j} = \frac{y_k - y_{k-n}}{nT} \tag{2}$$

As one can see on equation 2, the velocity estimation is a moving average over n Euler differentiations. The value of n is called the length of the window, and $n = max\{1, 2, 3, ...\}$ such that

$$|y_{k-i} - {}^{L}y_{k-i}| \leq d, \forall i \in \{1, 2, ..., n\} \tag{3}$$

Where d is the maximum noise on the position estimation, T is the sampling period and ${}^{L}y_{k-i} = a_n + b_n(k-i)T$ the linear regression, where the regression factors are given by:

$$a_n = \frac{ky_{k-n} + (n-k)y_k}{n}, \text{ and } \hat{v}_k = b_n = \frac{y_k - y_{k-n}}{nT} \tag{4}$$

The algorithm to implement end-fit-FOAW estimator works as follows:

1) Set $i = 1$.
2) Set y_k as the last sample and y_{k-i} as the i-th before y_k.
3) Calculate b_n, slope of the line passing through y_k and y_{k-i} from equation 4.
4) Check whether the line passes through all points inside the window within the uncertainty band of each point.
5) If so, set $i = i+1$ and go to step 3), else return the last estimate for $\hat{v}_k$.

IV. Levant's Differentiation

This technique is a robust exact differentiation using Second-Order Sliding Mode (SOSM) for signals with a given upper bound on the Lipschitz's constant of the derivative C:

$$\frac{\left|\dot{f}(t_1) - \dot{f}(t_2)\right|}{|t_1 - t_2|} \leq C = 50'000 \ deg/s^2 \tag{5}$$

The velocity is estimated using the following equations:

$$\begin{aligned} \dot{x}_k &= u_k - \lambda |e_k|^{1/2} \operatorname{sign}(e_k) \\ \dot{u}_k &= -\alpha \operatorname{sign}(e_k) \end{aligned} \tag{6}$$

where $e_k = x_k - y_k$ is the error between the measured position y_k of the encoder and the estimated position x_k

of the Haptic Paddle. The measurement of the encoder is used as y_k and thus $e_k \leq |d|$ is quantization noise with $d = \pm 0.012°$ the resolution of the encoder on the paddle side. α and λ are strictly positive tuning parameters that determine the differentiation accuracy. To ensure finite time convergence of the estimated velocity to the true velocity, these parameters were chosen as follows:

$$\alpha = 1.1C, \;\; \lambda = C^{1/2} \tag{7}$$

V. Kalman Filter

The Kalman filter is a stochastic method and has been implemented with a triple integrator model for better accuracy. The discrete-time linear model of the triple integrator describes how the next states are estimated.

$$\begin{aligned} \boldsymbol{x}_{k+1} &= A\boldsymbol{x}_k + G\boldsymbol{w}_k \\ y_k &= H\boldsymbol{x}_k + e_k \end{aligned} \tag{8}$$

where the state vector at sampling time k contains respectively: the estimated position, the estimated velocity and the estimated acceleration of the Haptic Paddle $\boldsymbol{x}_k = (x_k, \dot{x}_k, \ddot{x}_k)^T$. The measurement noise $e_k = x_k - y_k$ is the error between the measured position y_k of the encoder and the estimated position x_k of the Haptic Paddle. The process noise $\boldsymbol{w}_k$ and the measurement noise e_k are assumed to be Gaussian zero-mean white noise. The dynamics of the triple integrator are modelled in the state transition matrix A. The observation matrix H is constructed such that the only measured state is the position of the Haptic Paddle. The following matrices are used:

$$A = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{9}$$

The algorithm of the Kalman filter consists of a prediction step and an update step, which is implemented using the following equations:

$$\begin{aligned} &\text{Prediction step:} && \hat{\boldsymbol{x}}_{k,k-1} = A\hat{\boldsymbol{x}}_{k-1,k-1} \\ & && P_{k,k-1} = \lambda_k A P_{k-1,k-1} A^T + Q_k \\ &\text{Optimal Gain:} && K_k = P_{k,k-1} H_k^T \left[R + H_k P_{k,k-1} H_k^T\right]^{-1} \\ &\text{Update step:} && \hat{\boldsymbol{x}}_{k,k} = \hat{\boldsymbol{x}}_{k,k-1} + K_k \left[y_k - H_k \hat{\boldsymbol{x}}_{k,k-1}\right] \\ & && P_{k,k} = P_{k,k-1} - K_k H_k P_{k,k-1} \end{aligned} \tag{10}$$

The forgetting factor is set to $\lambda_k = 1$ to have no effect on the recursive algorithm, but could be lowered to reduce the effect of past data. This technique turns the Kalman filter into an Adaptive fading Kalman filter (AKF).

The scalar R represents the variance of the quantization noise.

$$R = \operatorname{var}(e_k) = E\left[e_k^2\right] = \frac{d^2}{3} = 0.000048^{(°)^2} \tag{11}$$

The error covariance matrix P_k is initialized taking into account the quantization noise. The process error covariance matrix Q_k needs to be initialized taking into account the unknown process noise.

$$P_0 = \begin{bmatrix} \frac{d^2}{3} & 0 & 0 \\ 0 & \frac{2d^2}{3T^2} & 0 \\ 0 & 0 & \frac{2d^2}{3T^4} \end{bmatrix}, Q_k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q \end{bmatrix} \tag{12}$$

where d is the resolution of the encoder measured on the paddle side, which constrains the quantization noise, T denotes the sampling period and the factor $q = 100'000$ was tuned experimentally. To initialize the state vector, the first two readings from the encoder are used to approximate the velocity and acceleration using Euler's differentiation method, and the second reading is used as the initial position estimate.

VI. Velocity Estimates Comparison

The various velocity estimation methods require parameter tuning. If these parameters are not well tuned, the estimation suffers from delay or overshoot compared to the true velocity, especially for high frequency oscillations. The approximation of the velocity by the non-filtered Euler differentiation of the Paddle position is shown in yellow in Fig. 1. This velocity was filtered at 50Hz to obtain the magenta signal. The FOAW velocity estimate was further filtered at 50Hz because of the quantization noise that was still present. The velocity calculated with Levant's differentiation method is very noisy due to the factor α, especially around a constant velocity. For this reason it is additionally filtered at 50Hz to reduce the high frequency change of the error sign. Finally, the Kalman velocity estimate did not require further filtering after tuning.

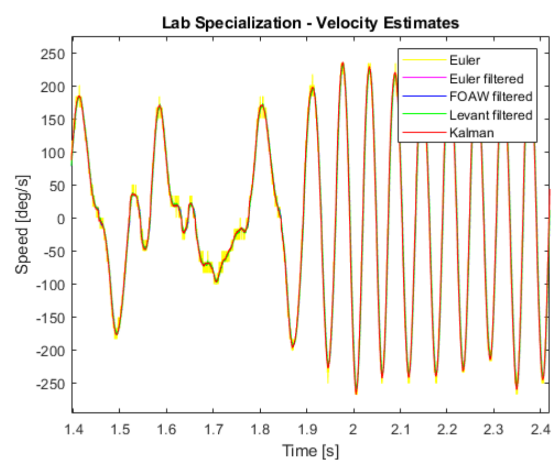

Fig. 1. Estimated speed using different methods, with the Haptic Paddle moved by hand

To visualize the accuracy of the different velocity estimators, a detailed view is shown in Fig. 2. The signals filtered with the 50Hz low-pass filter are smoother, but a delay of about $1ms$ is introduced. The Kalman filter using a triple integrator model is the most accurate method, as no filter needs to be used.

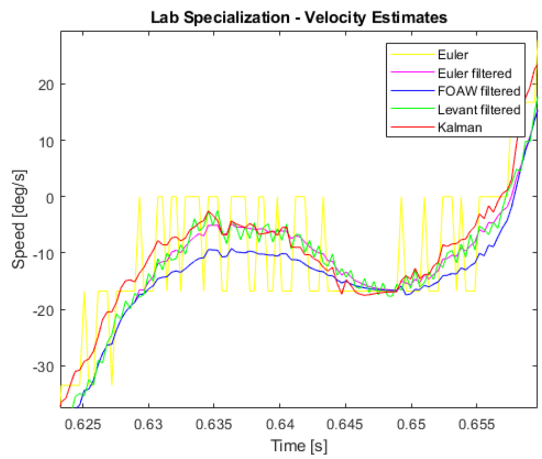

Fig. 2. Detailed view to show the delay in the estimated speed using different methods, with the Haptic Paddle moved by hand

To visualize the effect of high frequency velocity changes, a detailed view is shown in Fig. 3. At 1.61s, a small vibration of the Haptic Paddle has caused a kink in the approximated velocity. The Kalman filter is able to correctly estimate this velocity condition, but the filtered versions do not detect this vibration. One solution would be to increase the cutoff frequency to filter less, but then the estimate would become more noisy. This enlarged view also allows you to see the delay induced by the 50Hz low-pass filter.

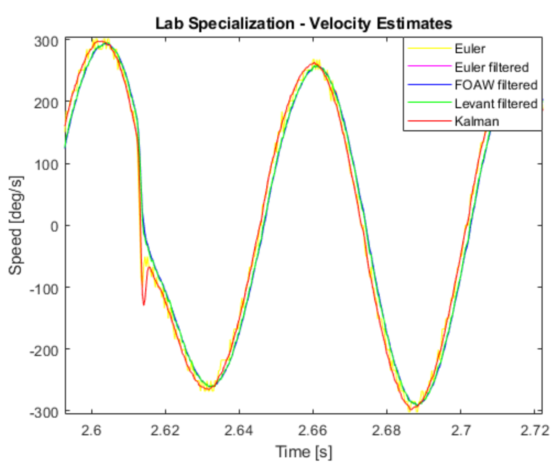

Fig. 3. Detailed view to show the high frequency variations in the estimated speed using different methods, with the Haptic Paddle moved by hand

VII. Virtual Wall Behaviour Comparison

In this section the velocity estimates behaviors against a virtual wall, modeled by a spring with stiffness K and damping factor B, are tested on the Haptic Paddle. The graph represented in Fig. 4 was obtained through manual grid search. For a range of virtual damping coefficients B, the paddle was pressed against the wall and the virtual stiffness coefficient K was gradually increased. The value of K for which the paddle started vibrating every time the paddle touched the wall is denoted with a cross. The interpolated points indicate the stability limit of the virtual wall. This experiment has been conducted separately for the four position and velocity estimation methods discussed in the previous sections. These methods were tuned in the time domain and the parameters remained the same for this virtual wall experiment. They were not specifically tuned to push the stability limit of the virtual wall, as the effect of changing the parameters could not be visualized directly, but only at the end of an experiment. In addition, the estimate was found to perform well in the time domain and did not want to be lost by an arbitrary parameter change.

The stability region of the Haptic Paddle against the virtual wall is the largest when Levant's differentiation is used, followed by FOAW. Even if the low-pass filter introduces a delay, which reduces the area of the stability region, this region is enlarged compared to the area with Euler differentiation. The Kalman filter performed best in the time domain, but the stability region is smaller than when using FOAW or Levant's differentiation. When moving the paddle by hand the limit of stability was felt blurred when using the Kalman filter, but for the other methods the limit was more abrupt. Another observation that was made is that this stability limit greatly depends on the mechanical vibrations of the paddle that occurs when touching the wall. If the paddle is held softly, the stability area will be smaller than when the paddle is held strongly.

Generally, a high sampling frequency or small sampling period results in big errors due to amplification of measurement noise, which will decreases the area of the stability region. FOAW rejects noise and extends the stability region of the wall, compared to the Euler method.

VIII. Conclusion

In this article were presented several methods to differentiate position estimations in order to estimate the velocity. Those methods can be selected depending on the system's requirement. If there is few requirements, Euler differentiation is the easier way to find a derivative. FOAW offers a good trade-off between reactivity and precision. Levant's differentiation performs best on the virtual wall experiment, it's implementation is not too hard and the computational cost is not as high as for the Kalman filter. For these reasons this velocity estimator is a good option for haptic interfaces or robotic installations generally.

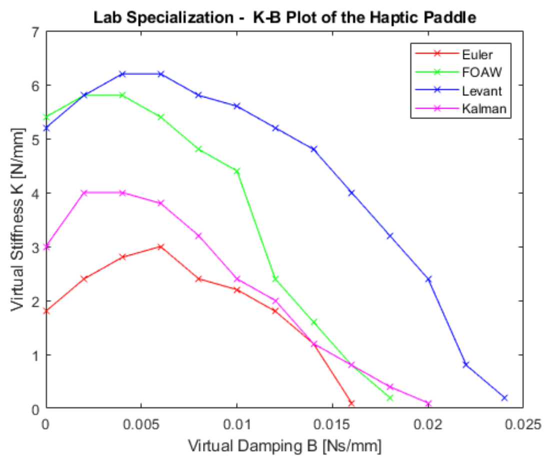

Fig. 4. Plot of the stiffness K - damping B characteristics of the virtual wall implemented on the Haptic Paddle

Several of the proposed methods can be combined in order to get a robust velocity estimation, such as FOAW and Kalman filter and could potentially result in better performances than each of them separately, or even compete with the method using the Levant differentiation. These methods can also be improved individually, for example using best-fit instead of end-fit FOAW.

The Kalman filter performs best when it comes to velocity estimation in time domain. Even if the computation is heavier than for other methods, the performances are worth it, when Newtonian dynamics of the system are predominant. In addition to providing us with the best estimate, this stochastic method also, provides the variance of the estimation error. For this reason Kalman filters are a good option for velocity, and generally for state estimation.

In this work a tripple integrator model was implemented as the Kalman filter, but the Kalman filter could be used with a different structure to combine sensor measurement from the incremental encoder and the absolute hall position sensor. If an IMU was added this stucture could be modified to also incorporate it's measurements and improve the state estimation of the Haptic Paddle. This article is an opening for further researches.

ACKNOWLEDGMENT

The authors would like to thank Professor Mohamed Bouri and Ph.D. students Ali Reza Manzoori, Zeynep Ozge Orhan, and Evgenia Vladimirova Roussinova for their generous guidance and supervision of the course "Haptic Human-Robot Interfaces", as well as for the opportunity to specialize in the very useful topic of velocity estimation.

REFERENCES

[1] V. Chawda, O. Celik and M. K. O'Malley, "Evaluation of Velocity Estimation Methods Based on Their Effect on Haptic Device Performance," in IEEE/ASME Transactions on Mechatronics, vol. 23, no. 2, pp. 604-613, April 2018, doi: 10.1109/TMECH.2018.2805863

[2] F. Janabi-Sharifi, V. Hayward and C. S. J. Chen, "Discrete-time adaptive windowing for velocity estimation", IEEE Trans. Control Syst. Technol., vol. 8, no. 6, pp. 1003-1009, Nov. 2002