

REAL TIME ANGULAR VELOCITY MEASURE USING MEDIAN FILTER

ALLAN DE MEDEIROS MARTINS, SAMAHERNI MORAIS DIAS, KURIOS QUEIROZ

*Dep. of Electrical Engineering – Technology Center
Federal University of Rio Grande do Norte ZIP 59.072-970 – Natal - RN
E-mails: {allan, sama, kurios}@dee.ufrn.br*

Abstract— We present a simple approach to reduce the noise present in velocity measurements when using encoders. The approach is to use the median filter together with a simple first order linear filter and show that this combination avoids the errors due to warp of the encoder (when the angle goes from 2π back to 0 of vice versa). We test this new method and compare with different kinds of filters and algorithms to avoid the warp problem. We show that the proposed method can be implemented only with simple arithmetic and is perfectly suitable for embedded and hardware implementation.

Keywords— Denoising, Median Filter, Velocity Measure

1 Introduction

The need for measuring velocity in motors does not require justification. There are countless applications where one needs to measure the speed of a motor. Most of the velocity measurements made are noisy. Noise can come from all sorts of sources, from thermal noise to numerical error noise[1]. That fact justify the need for better algorithms to filter the measured value aiming to get as close as possible to the true signal, one that carry the desired information, in this case the velocity of a motor. There are several works about signal denoising[12-15] but most of them uses sophisticated techniques that some times are hard to implement in real time. The most common approach for denoising in real time (or some times designated as on-line) is to use linear filters designed based on some knowledge of the signal (for instance the band of the signal)[8].

In this paper we propose the use of the median filter[17] applied to angular velocity measurement. The use of median filter in signal processing has already been proposed[5] and some theoretical analysis has been made[7-9]. The median filter has a simple structure that is easily implemented, allowing it to be assembled using embedded hardware like (FPGAs, DSPs, Microcontrollers, etc.)[3,5].

Our goal is to use a median filter together with a simple low pass filter to reduce the noise in the measurement. The advantage is, as we will see, that this new approach only require simple arithmetic and avoids the use of transcendental functions or special algorithms to denoise. We will start by showing the problem of measuring velocity using an encoder and how can this be solved using a simple trigonometric manipulation. Then we compare the filtering with and without this manipulation, showing that without it, the filtering problem is hard to solve. Since this solution uses trigonometric functions, it is not suitable for most embedded applications. Finally we will show the proposed method and compare the results.

2 Angular Velocity measurement

2.1 Measurement using an encoder

When one uses an encoder to measure velocity of the axis of a motor, one approximates the velocity as

$$\omega[k] = \frac{\theta[k] - \theta[k-1]}{h} \quad (1)$$

The problem is that this computation (and even most elaborated ones as second order numerical derivation) induces a special kind of noise due to the warp of the encoder (when θ goes from 2π back to 0 or vice versa). This abrupt change, appears in the velocity as a huge peak that require a strong low pass filter to eliminate. The problem is that the power spectrum of the peak contaminates all the signal spectrum[xx]. Because of this, the required band pass of the filter have to be too narrow to eliminate most of the peak influence in the signal. Being narrow, the filter ends up filtering part of the signal.

Some work was been done to improve numerical differentiation in signals[6,10,11] including the angular velocity[10]. Our goal is to propose an alternative method that allows the implementation to be simple and that can be implemented in real-time.

2.2 Un-Warping the angle

Due to the warp of the angle in the encoder, one must use some technique to overcome the avoid the creation of the peak in the derivative. One simple trick is to write the equation (1) as

$$\sin(h\omega[k]) = \sin(\theta[k] - \theta[k-1]) \quad (2)$$

Now one can write the right side as the sum of sines form and take the inverse sin in the left side, resulting in

$$\omega[k] = \frac{1}{h} \sin^{-1}(\sin(\theta[k])\cos(\theta[k-1]) - \sin(\theta[k-1])\cos(\theta[k])) \quad (3)$$

Apparently nothing changed between equation (1) and (3), but notice that in equation (3) all the angles are inside sines and cossines. Now, the warp of the encoder angle has no abrupt effect on the angular velocity since $\sin(0) = \sin(2\pi)$ and $\cos(0) = \cos(2\pi)$.

3 Median Filter

The median filter is a filter that is normally used in image processing[2] but was been used in signal processing[7] specially for denoising tasks. It consists in using a window of size L (normally odd in size) to go through the signal (as if it were convolving the signal). For each sample n of the signal, one places the center of the window at that sample and take the median of the L samples of the signal within the window. The median is taken by sorting the samples in, say, ascending order of magnitude and taking the one that falls in the middle of the sorted sequence.

The use of the median filter is motivated by its power to remove outliers in the signal. For instance, consider the signal

$$x = \{..., 0; \overbrace{0.1; 0.2; 5.8; 0.3; -0.1}^{L=5}; 0.3; ...\}$$

Say we want to find the median of the highlighted samples. One can notice that there is an outlier (value 5.8) within the set. Applying the median, one sorts the set yielding to

$$x = \{..., 0; \overbrace{0.1; 0.1; 0.2; 0.3; 5.8}^{L=5}; 0.3; ...\}$$

Now the outlier is replaced by the value 0.2, and the signal becomes

$$x = \{..., 0; 0.1; 0.2; 0.2; 0.3; -0.1; 0.3; ...\}$$

It is important to notice that the outlier is replaced by a value that gets repeated. This has to be taken into account in the implementation as we will see latter on.

Because of this outlier avoidance, this is exact the kind of filter one looks for to avoid the 2π warp when measuring velocity with an encoder.

4 Algorithm

We will compare three algorithms; The first one will consist of a simple low pass filter in the signal without the trigonometric un-warp. The second will consist of the same filter but with the un-warp. Finally

the last one will consists in of the proposed algorithm using the median filter together with a low pass filter. We aim to focus on the implementation, so we will present the algorithms as pseudo-codes. In all algorithms, the current sample has index k , the current velocity is represented by $\omega[k]$ (in radians per second), $\theta[k]$ is the current encoder reading (in radians) and the sampling period is h . The variable p represents the cut-off of the filter (the lower the value of p the lower is the band of the filter).

4.1 Low pass without un-warping

This algorithm is the fastest one and the simplest. It is summarized by the pseudo-code 1.

1. $\omega[k] = \frac{\theta[k] - \theta[k-1]}{h}$
2. $\omega[k] = p\omega[k] + (1-p)\omega[k-1]$

4.2 Low pass with un-warping

This algorithm basically does the same thing the first one does but the velocity is computed with equation 3. the algorithm then becomes

1. $\omega[k] = \frac{1}{h} \sin^{-1}(\sin(\theta[k])\cos(\theta[k-1]) - \sin(\theta[k-1])\cos(\theta[k]))$
2. $\omega[k] = p\omega[k] + (1-p)\omega[k-1]$

4.3 Proposed median filter

The proposed algorithm consists in applying the median filter to the velocity and then apply a low pass filter as described by the algorithm

1. $\omega_d[k] = \frac{\theta[k] - \theta[k-1]}{h}$
2. $\omega[k] = \text{median}(\omega_d[k-L], \omega_d[k-L+1], \dots, \omega_d[k])$
3. $\omega[k] = p\omega[k] + (1-p)\omega[k-1]$

One can notice that we introduced another variable ($\omega_d[k]$). This set of L variables are used to store the latest values of the "raw" velocity (velocity that was not processed and still has the warp angle). In fact we detailed this simple algorithms to emphasize that the proposed algorithm uses a little bit more memory (to store the L components of the signal $\omega_d[k]$). This is necessary because at each moment, one needs the values of the velocity that were not filtered, otherwise when the next sample arrive, the set to be fil-

tered by the median filter will contain repeated values of the median previously computed (as illustrated in the example of section 3).

Comparing the second the first algorithms it is important to notice that the main computational difference is the presence of a trigonometric and trigonometric inverse function in one and the sorting computation in the second. The first algorithm, as we will see is unusable and was presented only for consistency purpose.

5 Results

To illustrate the proposed algorithm we acquired the position of the rotor of an induction motor running with a velocity control. We used a 13bit absolute encoder to read the rotor angle. The data is instantaneously differentiated using (1) or (3). Figure 1 shows the whole data after differentiation using (1). The average velocity is about 180 rad/s. One can notice big peaks due to the warp of the angle (from 2π to 0). Figure 2 shows the same variable but with the warping procedure applied. It is still possible to notice some big peaks due to bit reading problems. To compare the result of the proposed filter we used two first order low pass filters given by

$$x[k+1] = bx[k] + (1-b)x[k-1] \quad (4)$$

where b is the pass-band in normalized units. To illustrate the performance of the proposed filter we used two filters denominated in this paper as *low-pass 1* ($b=0.1$) and *low-pass 2* ($b=0.01$). We chose those bands to show that in both cases the result is not good (first one keeps the peaks and the second one smoothes out the signal too much).

As can be seen in the figures 3 to 6, a low pass is needed to denoise the signal, but no compromise between smoothing and peak erasure can be archive using conventional low pass filters (that's

why we only tried first order simple filters, since the point can be made about the smoothing versus peak compromise).

Figure 3 is the first comparison of the filters. The figure is zoomed out to show the presence of the large peaks in both low pass filters if the signal is not un-warped. In fact without the un-warp procedure, the use of a simple low-pass filter is prohibitive. For the next results we will show always the result of the proposed filter *without* the un-warp procedure but we will use the un-warp procedure to the low-pass filters. This makes the comparison unfair to the proposed filter, but the goal is to show that even in this situation, the median filter can overcome the use of conventional low pass filters. In all of the tests we use the median with a window with 7 samples in length.

Figure 4 shows the first second of the signal. We can notice the angular velocity increase with small "bumps". Those oscillations are the result of vibration of the sensor. We will use this fact to highlight even more the performance of the filters. Now we aim to have a filter that, at the same time be insensitive to the peaks, and do not smooth out the oscillations that are part of the signal. In figure 4 to 7 the gray signal in the background is the original signal (too noisy). Figure 5 shows the result of the proposed filter. One can see that no peak is present and the oscillations are quite evident and denoised. In figure 6 we show a zoom in the signal (to detail the oscillations) and the results of the two low pass filters. It is clear that no compromise between smoothing and peak avoidance is possible. While the first filter one keeps the oscillations (does not smoothes it out) it keeps also the undesirable peaks, the second smoothes out the peaks (not all) but loose the oscillations (that are part of the signal). Figure 7 finally shows the same zoomed result but for the proposed filter. As one can notice, no peaks are present and no over-smoothing was done.

Next we will analyze the performance of the spectrum of the filters in the frequency domain.

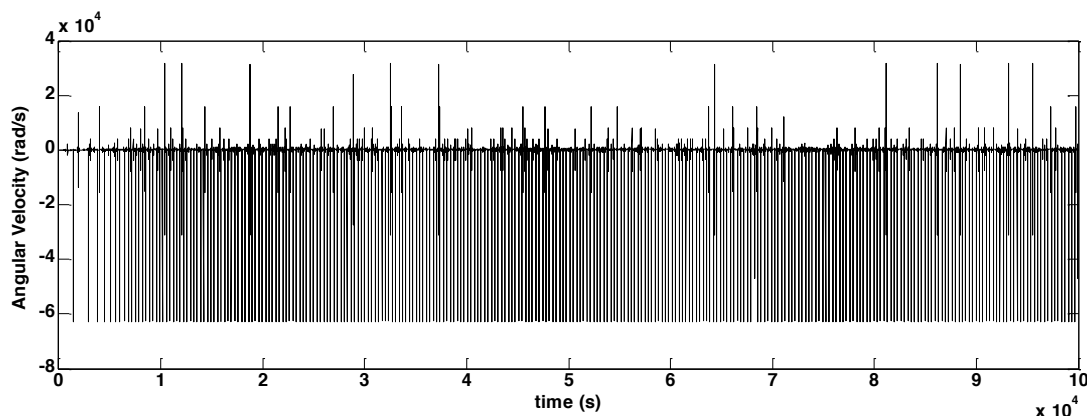


Figure 1 – Data acquired from a 13bit encoder and numerically differentiated using (1). The scale was preserved to show the large peaks due to the warp from 2π back to 0.

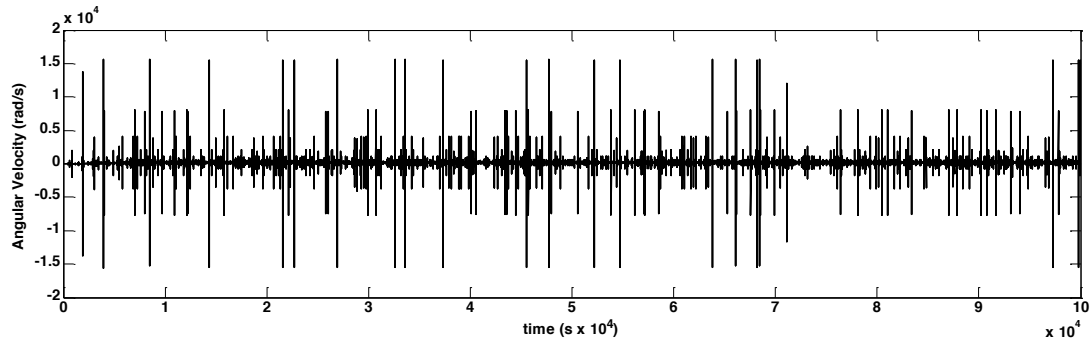


Figure 2 – Same data acquired from a 13bit encoder but numerically differentiated using the un-warping technique (3). The scale also was preserved to show the large peaks now due to faults in the encoder reader (bit faults).

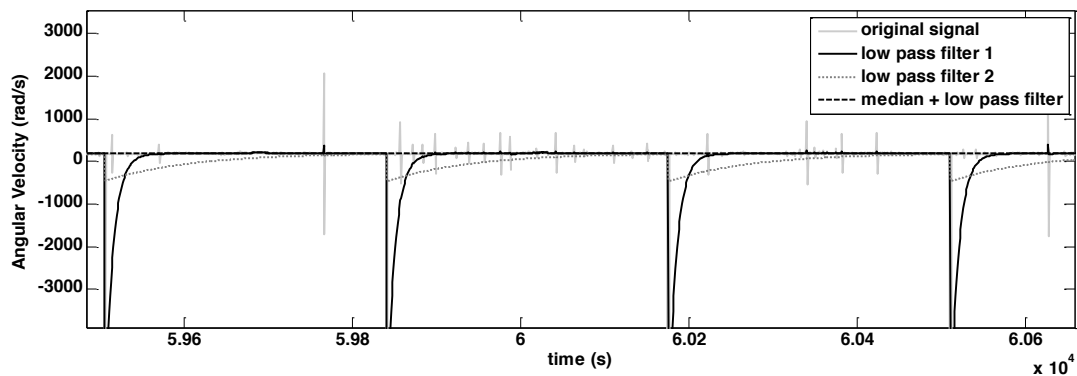


Figure 3 – Magnification of part of the signal filtered by the three filters (without the un-warp). We can see the effect of the warp peak using a filter that has low pass characteristics.

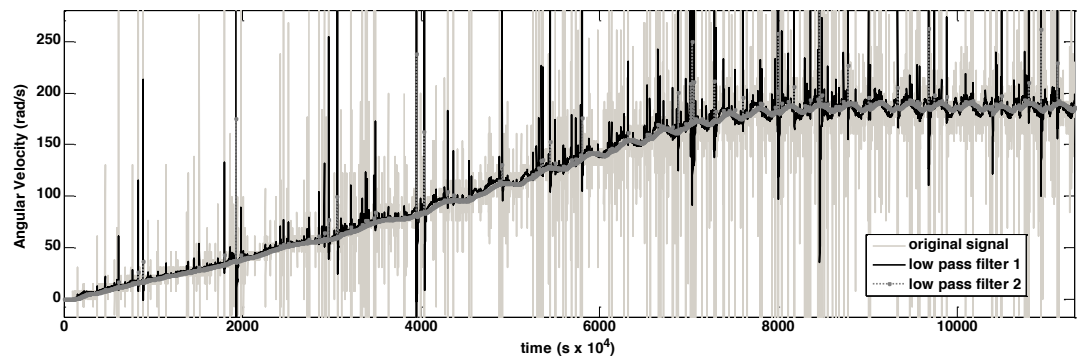


Figure 4 – Data filtered with low pass filters. Notice the presence of oscillation (due to vibration of the sensor). The filters are able to extract that oscillation but the peaks are still present. As we increase the cutoff frequency to reduce the peaks, the oscillation is also smoothen out.

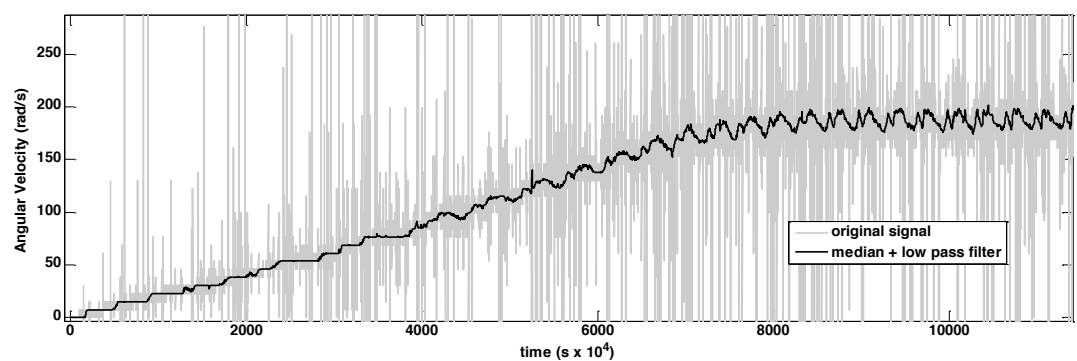


Figure 5 – Same result as in figure 4, but this time with the proposed filter. Notice that the oscillations are not smoothen out and no peaks are present.

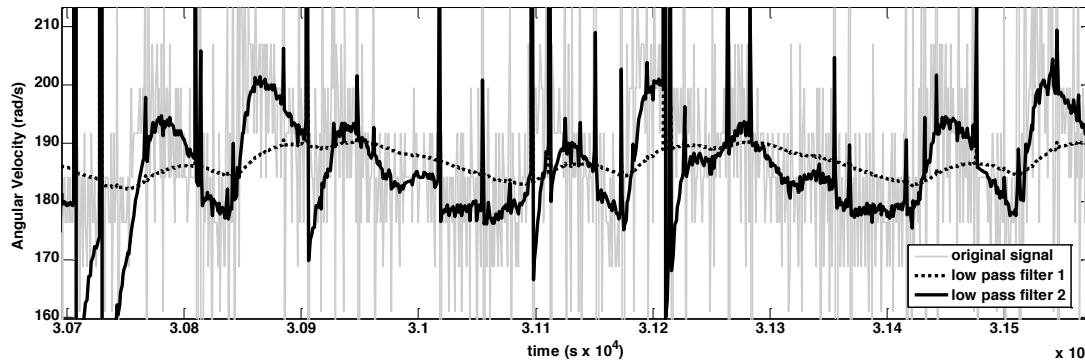


Figure 6 – Zoom-in on the result showed in figure 4. The smoothing of the oscillations is evident in the filter with low cutoff frequency. On the other hand, the peaks are present in the filter with higher cutoff frequency.

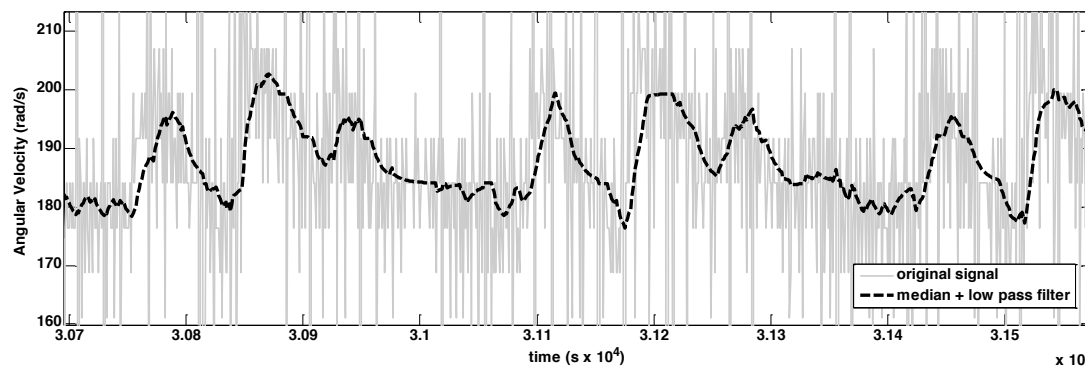


Figure 7 – Same result as in figure 6 but with the proposed filter. No peaks and no over-smoothing is present.

5.1 Spectral analysis and interpretation

In order to quantify the denoising performance of the proposed filter we will examine the results in terms of its spectrum. To do that we will consider only the stable part of the angular velocity (from the first second on) and with the mean removed. This procedure ensure that we will have only the periodic oscillations due to the vibration of the motor.

Figure 8 shows the Fast Fourier Transform (FFT) of the signal (un-warped). As can be seeing, due to the impulsive nature of the noise, there are a massive presence of high frequencies. Figure 8 also shows the first frequencies of the spectrum of the signal. There we can see the peaks that compose the oscillation in the signal. With an off-line filter (erasing the frequencies that are in those peaks) one can easily filter out the whole noise and get the oscillations (to a certain precision given by the frequencies that were evident in the spectrum). Of course this procedure is not viable to be used in an on-line situation since one needs the whole signal to perform the FFT erase the frequencies and go back to the time domain. This off-line filtered signal is a good estimate of the noise-free signal that is composed only by the vibrations and can be used to quantify the result of a filter.

Figure 9 shows the result of the three filters compared with the off-line result. We can see that the proposed filter is the closest to the off-line result. In terms of error, table 1 shows the comparison. The errors are measure as the rms value given by

$$e = \sqrt{\frac{1}{N} \sum_{k=1}^N (x[k] - x_f[k])^2} \quad (5)$$

for each situation (each filter). Where N is the number of samples, $x[k]$ is the off-line filtered signal and $x_f[k]$ is the signal filtered by the corresponding filter.

Table 1 - Errors for the filters result against the off-line filter (RMS value)

Filter	median	LP 1	LP 2
Error (with un-warping)	4.64	33.43	6.27
Error (without un-warping)	4.64	727.22	158.65

In the first row, the error for the conventional filters were computed using the filters plus the un-warping procedure. The second row shows the same result without the un-warping procedure (for the conventional low pass filters).

It is worth to emphasize that the errors for the proposed filter were measured in both cases with the original signal (without un-warping).

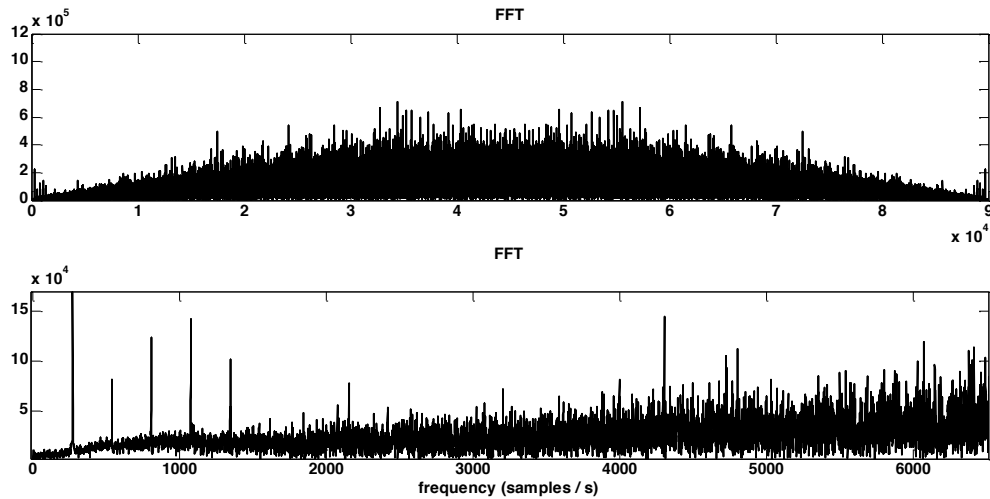


Figure 8 - Fast Fourier Transform of the original signal (un-wrapped). Up, whole series. Down, zoom in the first 6500 samples.

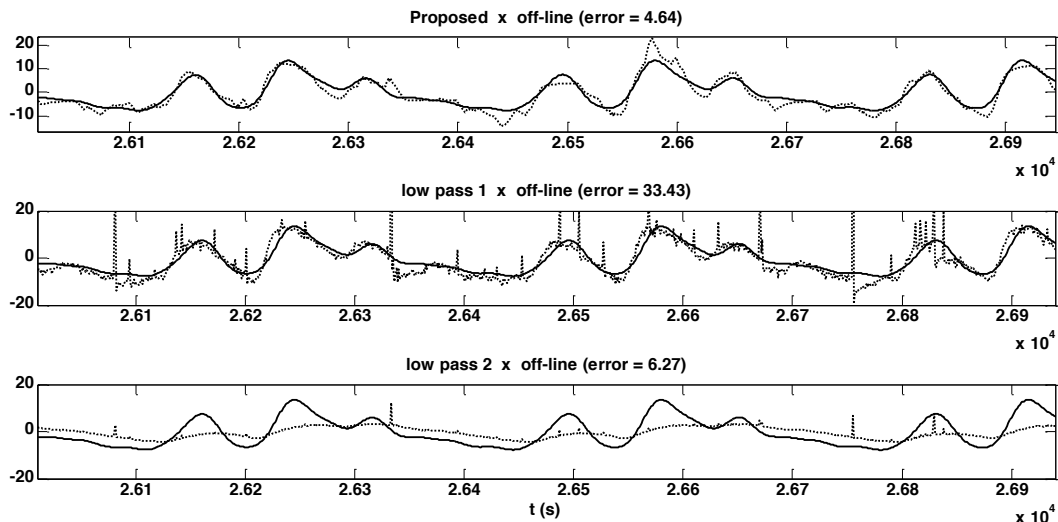


Figure 9 - Comparison of the three filters against the off-line filter (with errors). The continuous lines are the off-line result. and the dashed lines are the filters result in the sequence; The proposed filter, low pass 1 and low pass 2.

6 Conclusions

We presented an algorithm for filtering velocity measurement when one uses an encoder as angular position sensor. We showed that the angle warp problem can be solved basically by two algorithms. One uses direct and inverse trigonometric functions and the second (the proposed algorithm) uses only simple arithmetic operations. The problem with trigonometric functions is that the algorithm limits itself to implementations of those functions on a given hardware. Normally floating point operations and / or large lookup tables are required to implement such functions. This is a huge disadvantage when one is looking for a simple hardware implementation to

measure angular velocity (like DSPs, FPGAs, etc...). Since the proposed method uses only memory manipulations (for the sorting algorithm) and basic numeric operations, it can be implemented using fixed point arithmetic. This is suitable for low level implementations where floating point is not available or one has limited resources to spend in a simple velocity sensor.

The results showed that the new algorithm was able to denoise a real signal measured from an encoder and even small disturbances like mechanical vibrations of the system were evident in the filtered signal. The quantitative results using an off-line filter, asserts that the proposed filter is good even when compared with conventional filters. The use of simple first order filters to compare were sufficient because the process of analyzing the compromise between

peak avoidance and smoothing characteristics failed in both cases (low pass 1 and 2). Moreover, the conventional filters were viable only in the un-warped signals which leads to complex trigonometric procedures. Therefore, although hi-order filters could be implemented, to be able to avoid the peaks, they would smooth out the signal at a point that little information would be present.

References

- [1] Alan V. Oppenheim, Ronald W. Schaffer, John R. Buck, **Discrete - Time Signal Processing**, Prentice-Hall, January 10, 1999
- [2] Rafael C. Gonzalez, Richard E. Woods, **Digital Image Processing**, Prentice Hall, January 15, 2002.
- [3] Miguel A. Vega-Rodríguez, Juan M. Sánchez-Pérez, Juan A. Gómez-Pulido, *An FPGA-based implementation for median filter meeting the real time requirements of automated visual inspection systems*, Proceedings of the 10th Mediterranean Conference on Control and Automation - MED2002 Lisbon, Portugal, July 9-12, 2002.
- [4] Mamadou Mboup, C'edric Join, Michel Fliess, *A revised look at numerical differentiation with an application to nonlinear feedback control*, Proceedings of the 15th Mediterranean Conference on Control & Automation, July 27-29, 2007, Athens – Greece
- [5] Gavin L. Bates & Saeid Nooshabadi, **FPGA Implementation of a Median Filter**, IEEE TENCON - Speech and Image Technologies for Computing and Telecommunications, 1997
- [6] A. A. Hiasat, M. M. Al-Ibrahim and K. M. Gharaibeh, *Design and implementation of a new efficient median filtering algorithm*, IEE Vision, Image and Signal Processing, Vol 146, Issue 5, pp 273-278, October 1999. DOI: [10.1049/ip-vis:19990444](https://doi.org/10.1049/ip-vis:19990444)
- [7] Gallagher Jr., N. and Wise, G., **A Theoretical Analysis of the Properties of Median filters**, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol 29, Issue 6, pp 1136 - 1141, December 1981
- [8] V. V. Bapeswara Rao and K. Sankara Rao, **A New Algorithm for Real-Time Median Filtering**, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol 34, issue 6, pp 1674 - 1675, December 1986
- [9] Arce, G. and McLoughlin, M., **Theoretical Analysis of the Max/Median Filter**, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol 35, issue 1, pp 60 - 69, January 1987. DOI: [10.1109/TASSP.1987.1165036](https://doi.org/10.1109/TASSP.1987.1165036)
- [10] diCenzo, Colin D. and Szabados, Barna and Sinha, Naresh K., **Digital Measurement of Angular Velocity for Instrumentation and Control**, IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol IEIC-23, Issue 1, pp 83 - 86, February 1976. DOI: [10.1109/TIEIC.1976.351354](https://doi.org/10.1109/TIEIC.1976.351354)
- [11] Wu Wenjiang and Sun Zhanpeng and Gao Zhanfeng, *Estimating Algorithm of Angular Velocity on Assisted Motor*, The Ninth International Conference on Electronic Measurement & Instruments ICEMI'2009
- [12] B. K. Natarajan, **Filtering random noise from deterministic signals via data compression**, IEEE Trans. Signal Processing, Vol 43, Issue 11, pp 2595-2605, 1995. DOI: [10.1109/78.482110](https://doi.org/10.1109/78.482110)
- [13] K. Konstantinides, B. Natarajan and G. S. Yovanof, **Noise estimation and filtering using block-based singular value decomposition**, IEEE Trans. Image Processing, Vol 6, Issue 3, pp 479-483, 1997. DOI: [10.1109/83.557359](https://doi.org/10.1109/83.557359)
- [14] H. Hassanpour, **A time-frequency approach for noise reduction**, Digital Signal Processing, Vol 18, pp 728-738, 2008. DOI: [10.1016/j.dsp.2007.09.014](https://doi.org/10.1016/j.dsp.2007.09.014)
- [15] M. Aminghafari, N. Cheze, J. M. Poggi, **Multivariate denoising using wavelets and principal component analysis**, Computational Statistics & Data Analysis, Vol 50, pp 2381 - 2398, 2006. DOI: [10.1016/j.csda.2004.12.010](https://doi.org/10.1016/j.csda.2004.12.010)
- [16] José F. Barraza and Norberto M. Grzywacz, **Measurement of angular velocity in the perception of rotation**, Vision Research, Vol, 42 pp 2457 - 2462, 2002 DOI: [10.1016/S0042-6989\(02\)00259-6](https://doi.org/10.1016/S0042-6989(02)00259-6)
- [17] Astola J. and Haavisto P. and Nuvo Y., **Vector Median Filter**, Proceedings of the IEEE ,Vol 78 ,Issue 4, pp 678 - 689, April 1990. DOI: [10.1109/5.54807](https://doi.org/10.1109/5.54807)