TRIPLE F (F³) FILTERING OF KINEMATIC DATA

Manfred M. Vieten
Sport Science, University of Konstanz, Konstanz, Germany

Raw data in biomechanical studies usually require filtering. Depending on the used filter there exist some drawbacks as “signal shifted relative to the raw data”, “instability or degradation”, “endpoint problem”, “oscillations”. A filter called triple F or F³ based on the Fourier transformation is presented that is not crippled by these drawbacks. It consist of a transformation of the original data into the frequency spectrum followed by eliminating the unwanted frequencies (windowing) and an inverse Fourier transformation back to the data as a function of time. This procedure is stable and does not shift the data. It is shown how to dampen additional oscillation on the filtered data and how to avoid the endpoint problem completely. A comparison with a Butterworth filter and an application completes the presentation.

KEYWORDS: Filtering, Fourier transformation, Kinematic data

INTRODUCTION: When using 3D kinematic data from a digitizing system as input for calculations in inverse dynamics, a noisy signal prevents successful computation. Because the numerical deviations amplify the noise disproportionately compared with the low frequency signal (Winter 1979, pp 29), filtering is unavoidable. One of the most widely used digital filters is the Butterworth filter (BWF) (Winter 1979, pp 35). The main reasons for choosing the BWF is its short mathematical form, hence its short computation time and especially its smooth transfer function. This ensures removing most of the unwanted higher frequencies. A comparison with the “moving average” (Haykin 1990, p 73), “FIR low pass” (Ifeachor and Jervis 1993, pp 278) and “exponential smoothing” (Hartung, Elpelt et al. 1989, pp 672) filters shows the superiority of the BWF. However, the BWF still allows substantial portions of higher frequency noise to pass. In addition, the output signal is shifted relative to the raw data and data points are lost. Data shifting can be avoided by a second filtering in the reverse direction of time. But in this way the filter’s order is doubled, as is the loss of data. In addition IIR filters, and the Butterworth is one, can for no apparent reason become unstable or degraded (Ifeachor and Jervis 1993, p 375). For the use of three-dimensional kinematic data for our simulation system, we need signals which display very few noisy artifacts, even in the first and second derivative with respect to time. Furthermore, the filtered output should not display a data shift relative to the original raw data. Therefore, we have developed a filtering method based on the fast Fourier transformation which we call “Fast Fourier Filtering” or simply “triple F (F³)”.

METHOD: F³ consists of five steps.

1. Extension of the original raw data as an extrapolation of the raw data.
2. Fast Fourier transformation of the extended raw data.
3. Windowing of the frequency spectrum by multiplying with a transfer function.
4. Inverse Fourier transformation back into the coordinate space.
5. Removing the extended data.

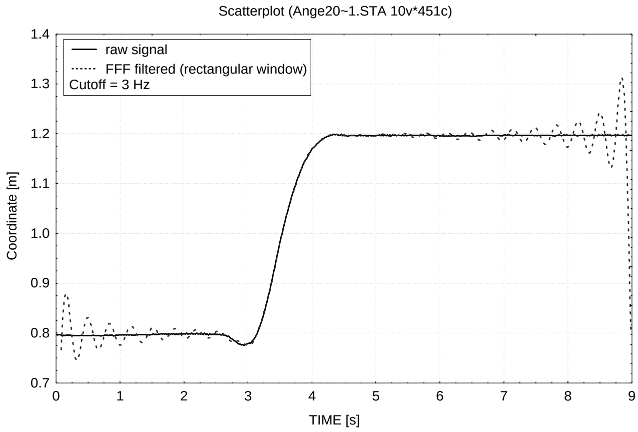
Step one consists of an extension that does not alter the spectral quality of the raw data. In addition, the extension needs to be smooth, and that means the numerical derivative must be identical with the raw data. This can be achieved by using the following equations:

$$x_{-i} = -x_i + 2x_0 \tag{1.1}$$

for the extension before the start of the raw data and

$$x_{N+i} = -x_{N-i} + 2x_N \tag{1.2}$$

for the extension at the end of the data. Here x_i is the original data, with x_0 being the first and x_N the last member of the raw data set. Our data sets contain an average of between 50 and 2000 data. We extrapolate at least 450 before and 450 after the original data. The extension is done by iterating the equations (1.1) and (1.2) simultaneously and, therefore, the extension is never undefined, even for very small data sets.



Step two is a fast Fourier filtering of the extended data. We use the Sande-Tukey algorithm (Ramirez 1985), which calculates for N data of a set 2^n spectral data. Here n is the smallest natural number for which the relation $N \le 2^n$ holds true. For our extended data sets we always choose the relation as $N = 2^n$. N is the sum of the original raw data set plus 900 plus the rest up to 2^n.

Figure 1: Oscillations of the filtered data using a rectangular window

In step three the frequency spectrum is windowed. The simplest window would of course be a rectangular window, where all but the desired frequencies are multiplied by zero and the rest is left unaltered. But this leads to unwanted oscillations of the retransformed coordinates (Figure 1). This is evident for the calculation of a slowly varying spectral function $y(f)$.

$$x'(t) = \int_{-f_s/2}^{f_s/2} H(f)\, y(f) \exp(i2\pi ft)\, df \rightarrow \int_{-f_s/2}^{f_s/2} H(f) \exp(i2\pi ft)\, df = \frac{\sin(2\pi Ct)}{\pi t} \quad (1.3)$$

Here $x'(t)$ is the filtered signal, f_s the sampling frequency, and $i = \sqrt{-1}$. The damping with the increase in time t is a mere $1/t$. C is the cutoff parameter that separates the unaltered from the altered data in the spectral function. The rectangular window (transfer function) is defined as

$$H(f) = \begin{cases} \theta(f + C) & f < 0 \\ 1 - \theta(f - C) & f \ge 0 \end{cases} \quad (1.4)$$

Here $\theta(f \pm C)$ is the step function, which is zero for the argument being smaller zero and one for the argument being bigger zero. We have obtained much better results by using a cosine window.

$$H(f) = \begin{cases} 0 & for \quad f < -C - \Delta \\ \frac{1}{2}\left(1 - \cos\left(\frac{\pi(f + C + \Delta)}{2\Delta}\right)\right) & for \quad -C - \Delta \le f \le -C + \Delta \\ 1 & for \quad -C + \Delta < f < C - \Delta \\ \frac{1}{2}\left(1 + \cos\left(\frac{\pi(f - C + \Delta)}{2\Delta}\right)\right) & for \quad C - \Delta \le f \le C + \Delta \\ 0 & for \quad C + \Delta < f \end{cases} \quad (1.5)$$

The cutoff C is defined as $H(C) = 1/2$. The cosine flank steepness Δ is half the frequency interval of $H(f)$, changing from one to zero or vice versa. Again as before, for a slowly varying spectral function $y(f)$ we get

$$
\begin{aligned}
x'(t) &= \int_{-f_s/2}^{f_s/2} H(f)\,y(f)\exp(i2\pi ft)\,df \\
&\rightarrow \int_{-f_s/2}^{f_s/2} H(f)\exp(i2\pi ft)\,df \\
&= \int_{-C-\Delta}^{-C+\Delta} \frac{1}{2}\left(1\text{-}\cos\left(\frac{\pi\left(\mathrm{f}+\mathrm{C}+\Delta\right)}{2\Delta}\right)\right)\exp(i2\pi ft)\,df + \int_{-C+\Delta}^{C-\Delta} \exp(i2\pi ft)\,df \\
&+ \int_{C-\Delta}^{C+\Delta} \frac{1}{2}\left(1+\cos\left(\frac{\pi\left(\mathrm{f}\text{-}\mathrm{C}+\Delta\right)}{2\Delta}\right)\right)\exp(i2\pi ft)\,df \\
&= \frac{-\sin(2\pi Ct)\cdot\cos(2\pi\Delta t)}{\pi t\left(16t^2\Delta^2-1\right)}
\end{aligned}
\qquad (1.6)
$$

The result displays a much better damping – proportional to $1/t^3$ – with respect to time.

Steps four and five are straight forward. Four is the inverse Fourier transformation of the extended and windowed raw data. This data set displays an endpoint problem. Step five is the simple truncation of added data in front and at the end of the original data set, which eliminates both the endpoint problem and removes the extended data.

RESULTS AND DISCUSSION: To compare the well known BWF with our F³ filter we used two different data sets. Data set one is rather artificial. It consists of 900 data points at a sampling frequency of 50 Hz that contains a superposition of simple sinus oscillations of 15 different frequencies. Beginning with 0.5 Hz up to 5 Hz, any frequencies in 0.5 Hz steps and 6, 8, 10, 15 and 20 Hz sinus waves are included. Data set two is from a clinical study. It is the vertical component of a head marker of a hemiplegic patient rising from a chair, a so-called Sit To Stand (STS) movement with a sampling frequency of 50 Hz.

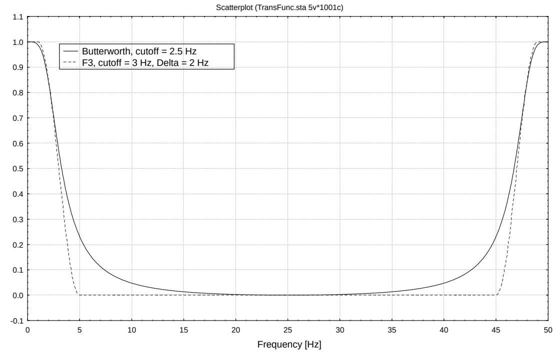


Figure 2: Norm of the transfer functions of Butterworth and F³ filters

First we give the results of the two filters to remove higher frequency noise. The specifications of the second order BWF used are as given in Winter (1979, p 36) – with the correction of two interchanged minus signs – for a cutoff frequency of 2.5 Hz. The absolute of the transfer functions is given in Figure 2. The specifications of the F³ filter are the cutoff frequency $C = 3\,Hz$ and a cosine flank steepness of $\Delta = 2\,Hz$. We should note here that the cutoffs for the two filters are defined differently. Nonetheless, the transfer functions have some similarity, whereby the F³ filter removes much more of the higher frequencies compared with the BWF, even when we increase the BWF order. This is clearly visible in the Fourier spectra of the first example data set, filtered BWF and F³ in comparison with the raw data in Figure 3.

However, the comparison of the two filterings for the STS data does not show many differences for the coordinates. The velocity, the first derivative with respect to time of the filtered data, shows only modest differences. The difference appears for the acceleration (Figure 4), the second derivative with respect to time of the filtered data.

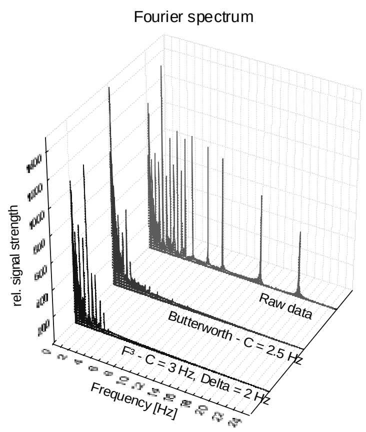


Figure 3: Relative strength of the Fourier spectra for raw data, Butterworth, and F³ filtered sets

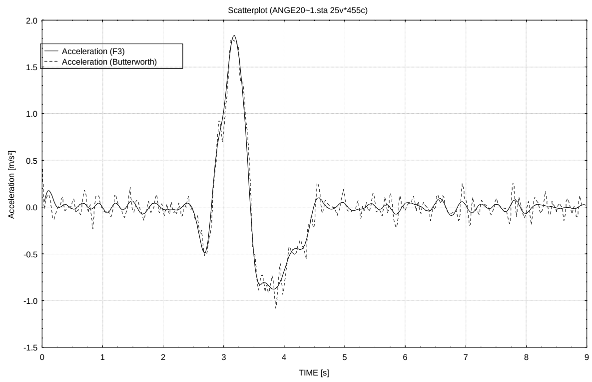


Figure 4: Acceleration of the STS movement calculated from coordinates filtered using a F³ and a BWF

Also here the damping of the higher harmonics is best for the F³ filter. A F³ filter, as we use it, has two control parameters (cutoff and cosine flank steepness) compared with the two parameters of the BWF (cutoff and order). But the control parameter of the F³ gives much more freedom to choose the filter's characteristic in the form of the transfer function. Where the BWF even alters the desired low frequencies of the spectrum, the F³ allows those frequencies to pass unaltered. The BWF cannot remove the higher frequency parts of the spectrum satisfyingly, while the F³ filter can. This is especially important for studies in which it is necessary to calculate higher derivatives of kinematic measurements. F³ filtered data do not have an endpoint problem (which would cause enormous deviations at the beginning and the end of the filtered data set display in comparison with the raw data) and do not show a shift relative to the original raw data as well as they do not lose data points. This makes it very easy to judge the influence of the filter on the results.

CONCLUSION: We have used the F³ filter successfully for clinical as well as for sports-related studies using simulation and inverse dynamics. However, it seems that there is no reason to restrict the method to the filtering of kinematic data. In the past, when computation was expensive and slow, there was a need for short, quick running algorithms, which widely excluded Fourier transformations. But with today's fast and inexpensive PCs this argument is no longer valid. With the described method the "endpoint problem" that cripples a simple fast Fourier-filtered sequence is eliminated. In conclusion, we would like to comment that there is no reason to limit this method to low pass filters. All kinds of filters – low pass, high pass, band pass, or even more exotic ones – are in principle easily constructible.

References

Hartung, J., B. Elpelt, et al. (1989). Statistik - Lehr- und Handbuch der angewandten Statistik. München, Wien, R. Oldenbourg Verlag.

Haykin, S. (1990). Modern filters. New York, MacMillian Publishing Company.

Ifeachor, E. C. and B. W. Jervis (1993). Digital signal processing : a practical approach. Wokingham, England; Reading, Mass., Addison-Wesley.

Ramirez, R. W. (1985). The FFT, Fundamentals and Concepts. London, Prentice-Hall International.

Winter, D. A. (1979). Biomechanics of human movement. New York, Wiley.