

<Filtering Method Comparison in MATLAB>

2017272043 이성진

Filtering method comparison (in matlab)

1. filter

- 1차원 디지털 필터
- $y = \text{filter}(b,a,x)$: 분자 및 분모 계수인 b 와 a 로 정의되는 유리 전달 함수로 입력 데이터 x 를 필터링

- 유리 전달 함수 (Rational Transfer Function)

$$Y(z) = \frac{b(1) + b(2)z^{-1} + \dots + b(n_b + 1)z^{-n_b}}{1 + a(2)z^{-1} + \dots + a(n_a + 1)z^{-n_a}} X(z), \quad \rightarrow \text{FIR 필터, IIR 필터 모두 처리 가능}$$

(n_a : 피드백 필터 차수 n_b : 피드포워드 필터 차수)

다음 차분 방정식으로도 표현 가능

$$a(1)y(n) = b(1)x(n) + b(2)x(n-1) + \dots + b(n_b + 1)x(n - n_b) \\ - a(2)y(n-1) - \dots - a(n_a + 1)y(n - n_a).$$

분자 및 분모 계수 b 와 a 를 어떻게 설정하느냐에 따라 필터 특성 결정

Filtering method comparison (in matlab)

1. filter

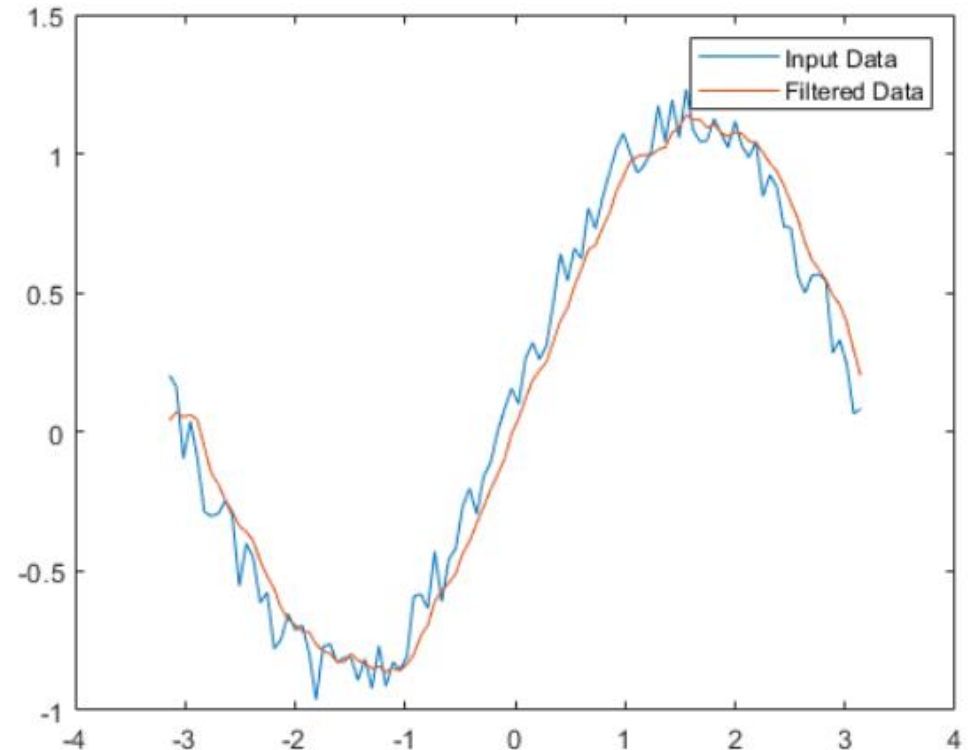
Ex) Moving-Averaged Filter : Noise가 있는 데이터의 평활화에 사용되는 일반적인 방법

벡터 x 의 Moving-Averaged Filter를 정의함

$$y(n) = \frac{1}{windowSize} (x(n) + x(n-1) + \dots + x(n - (windowSize - 1))).$$

Window 크기를 5로 정하고 Moving-Averaged Filter에 해당하는 유리 전달 함수의 분자 및 분모 계수 계산 후 플로팅

```
windowSize = 5;  
b = (1/windowSize)*ones(1,windowSize);  
a = 1;
```



Filtering method comparison (in matlab)

2. filtfilt

- 영위상(Zero-Phase) 디지털 필터링
- $y = \text{filtfilt}(b,a,x)$: 입력 데이터 x 를 순방향과 역방향 모두로 처리하여 영위상 디지털 필터링 수행
- 따라서, 위상 응답에 따라 결과가 달라지는 필터는 사용을 지양 ex) Hilbert FIR 필터, Differential 필터
- 원래 필터 전달 함수 크기의 제곱과 같은 필터 전달 함수
- b 와 a 로 지정된 필터 차수의 2배에 해당하는 필터 차수

<filtfilt 함수의 작동 원리 계산>

$x(n)$: input, $h(n)$: Filter's impulse response

$X(e^{j\omega})$: Fourier transform of $x(n)$

$H(e^{j\omega})$: Fourier transform of $h(n)$

순방향 필터 적용 : $X(e^{j\omega})H(e^{j\omega}) \rightarrow$ Time reverse $\rightarrow X(e^{-j\omega})H(e^{-j\omega})$

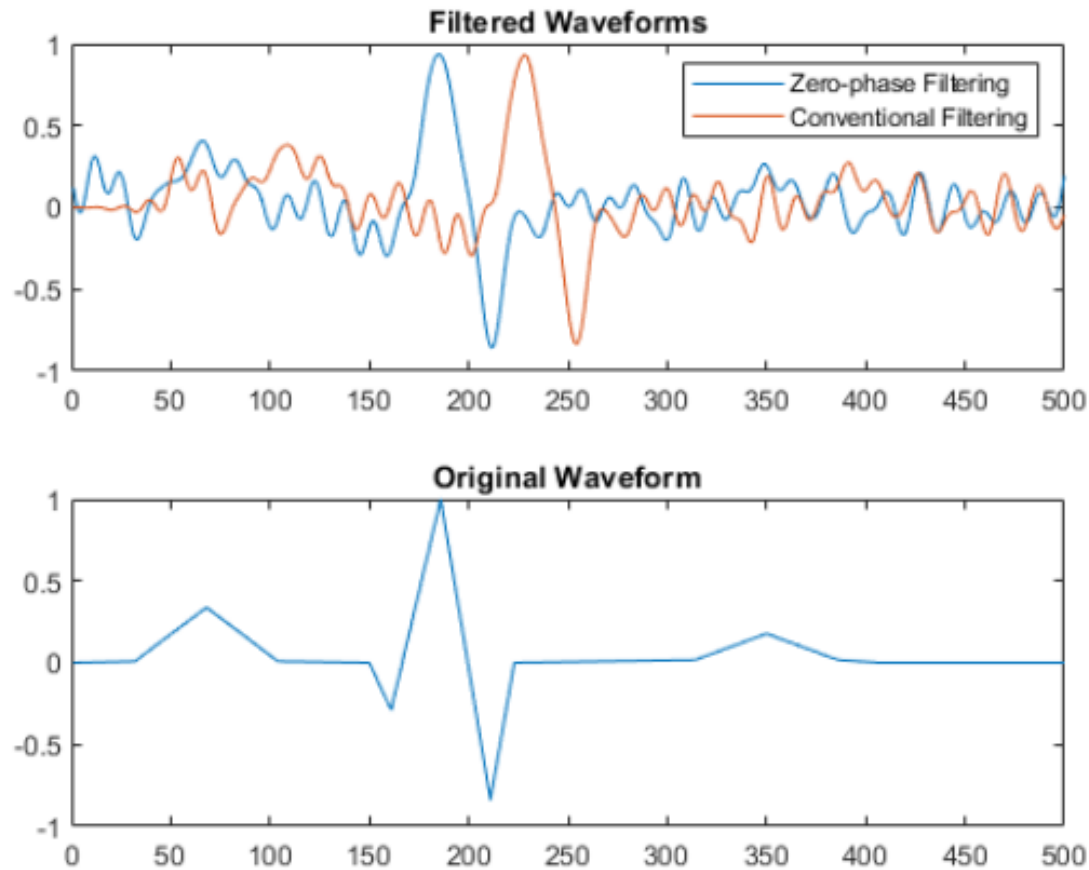
역방향 필터 적용 : $X(e^{-j\omega})H(e^{-j\omega})H(e^{j\omega}) \rightarrow$ Time reverse $\rightarrow X(e^{j\omega})H(e^{j\omega})H(e^{-j\omega})$

최종 출력 값 = $Y(e^{j\omega}) = X(e^{j\omega})H(e^{-j\omega})H(e^{j\omega}) = X(e^{j\omega})|H(e^{j\omega})|^2 \rightarrow \text{Zero-Phase}$

Filtering method comparison (in matlab)

2. filtfilt

Ex) 예제 심전도 파형에 Low-Pass FIR Filter를 filtfilt와 filter 함수를 이용하여 적용 후 비교



- ① Zero-phase Filtering : filtfilt 함수
Original Waveform과 위상 차이 X
- ② Conventional Filtering : filter 함수
Original Waveform과 위상 차이 발생

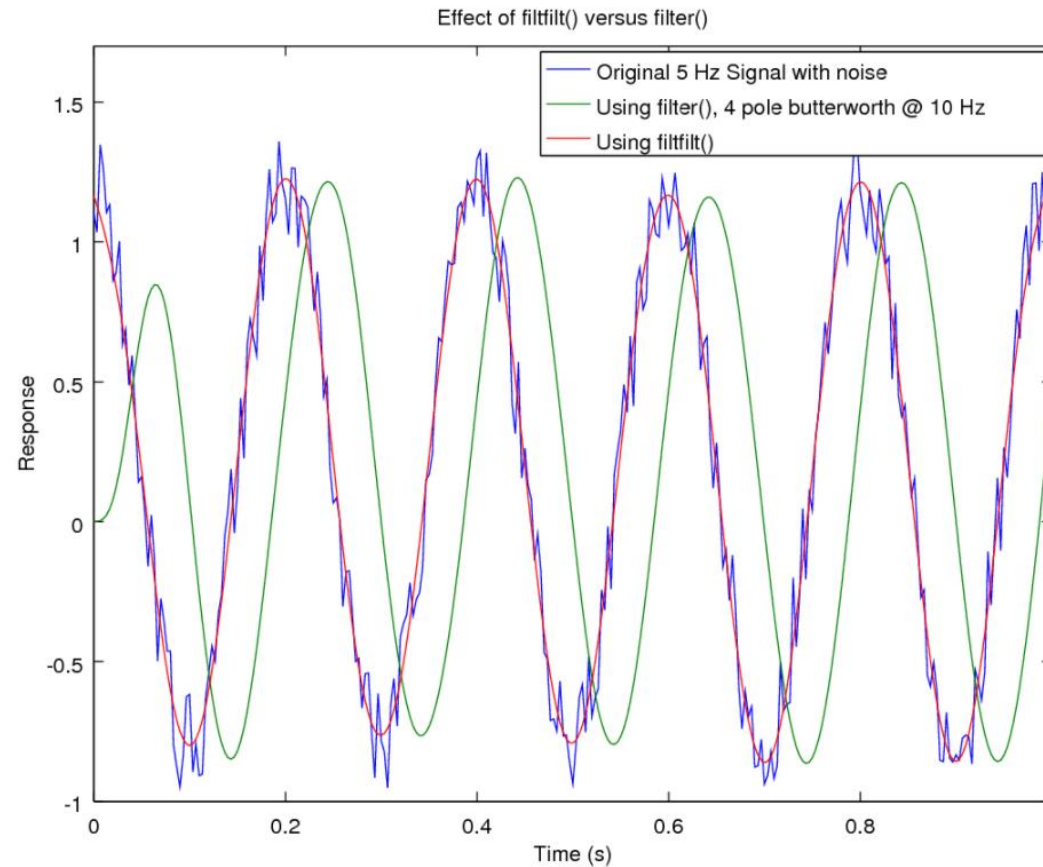
Filtering method comparison (in matlab)

filter vs filtfilt

- 어떤 필터든간에, "*Real-time*" 에서 작동하는 경우 $t = t_0$ 에서 출력은 $t \leq t_0$ 에 의존한다
- 데이터를 기록하고 이후에 처리하는 경우 $t > t_0$ 의 출력을 $t = t_0$ 에서 출력에 사용이 가능하다
→ 이를 "*non-causal filter*" 라고 하며 대표적인 예가 matlab의 filtfilt 함수이다
- filtfilt의 동작 방식은 실시간으로 발생할 수 없으며 모든 실시간 필터에는 약간의 지연이 있다
- filtfilt와 filter함수 모두 고주파 노이즈를 제거하지만 위상 지연에 대한 차이가 있다
- $t=0$ 에서 filter는 0에서 시작하지만 filtfilt는 초기 조건을 유지하여 잘 반영이 된다
- filtfilt는 데이터를 두 번 필터링하기 때문에 더 나은 진폭 응답을 가진다
→ 특히 stop band에서 더 날카로운 cutoff와 더 많은 감쇠를 가진다

Filtering method comparison (in matlab)

filter vs filtfilt



결론 : 거의 모든 일상적인 필터링 상황에서 filtfilt함수를 사용한다
filter함수를 사용하는 경우는 실시간 프로세스를 필터링할 때만 사용한다