Filtering

January 31, 2026

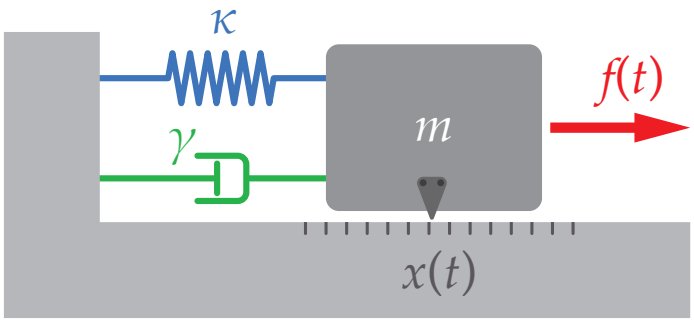
Every couple of months I find myself needing to smooth a noisy signal. These notes are a practical guide for doing this. Organizing this material has given me a welcome excuse to recall some beautiful ideas and constructions, and also to learn something new (never having worked with Butterworth filters before). My goal is for these notes to be useful & fun for me and hopefully you as well.

1 Mechanical Motivation

A spring-mass-damper system has dynamics

$$m\ddot{x} + \gamma\dot{x} + \kappa x = f,$$

and we think of this as a filter that transforms an input signal $f(t)$ into an output signal $x(t)$. As we apply a force $f(t)$ to the mass m, it moves with motion $x(t)$ that depends on $f(t)$, on m, on the stiffness of the spring κ, and on the damping coefficient γ. We have a lot of intuition for these kinds of systems because they are all around us.



2 Generalization

We generalize the spring-mass-damper system to create a filter that doesn't necessarily have a physical analog, but that is useful for modifying signals.

$$x^{[n]} + a_{n-1}x^{[n-1]} + \cdots + a_2x^{[2]} + a_1x^{[1]} + a_0x = b_0f,$$

with

$$x^{[k]} = \frac{d^kx}{dt^k}.$$

We make each time derivative the component of a point in $\mathbb{R}^n$. As an example, with $n = 4$,

$$x^{[4]} + a_3 x^{[3]} + a_2 x^{[2]} + a_1 x^{[1]} + a_0 x = b_0 f, \tag{1}$$

and we define $\mathbf{x} \in \mathbb{R}^4$ as

$$\mathbf{x} = \begin{bmatrix} x \\ x^{[1]} \\ x^{[2]} \\ x^{[3]} \end{bmatrix}.$$

We note that

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + b_0 \mathbf{f}, \tag{2}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 \end{bmatrix}, \text{ and } \mathbf{f} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ f \end{bmatrix}.$$

Solutions to (2) are given by

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}(t-s)}\mathbf{f}(s)ds \tag{3}$$

where the initial condition $\mathbf{x}_0$ is $\mathbf{x}$ at time $t = 0$, and where, for the operator $\mathbf{B} \in \mathbb{R}^{n\times n}$, we define the operator $e^{\mathbf{B}} \in \mathbb{R}^{n\times n}$ as

$$e^{\mathbf{B}} = \mathbf{I} + \mathbf{B} + \frac{1}{2!}\mathbf{B}^2 + \frac{1}{3!}\mathbf{B}^3 + \frac{1}{4!}\mathbf{B}^4 + \cdots$$

If $\mathbf{B}\mathbf{u} = \lambda\mathbf{u}$, we see that $e^{\mathbf{B}}\mathbf{u} = \mathbf{u}e^{\lambda}$. Likewise if $\mathbf{A}\mathbf{u} = \lambda\mathbf{u}$, then

$$e^{\mathbf{A}t}\mathbf{u} = \mathbf{u}e^{\lambda t}.$$

It follows that (3) is unstable if any of the eigenvalues of $\mathbf{A}$ has positive real part. These eigenvalues λ are the roots of

$$\lambda^4 + \lambda^3 a_3 + \lambda^2 a_2 + \lambda a_1 + a_0 = 0. \tag{4}$$

We can see the same thing by applying the Laplace transform

$$X(s) = \int_{-\infty}^{\infty} x(t)e^{-st}dt$$

to (1), obtaining

$$X(s) = H(s) * F(s)$$

where $H : \mathbb{C} \longrightarrow \mathbb{C}$ is the *transfer function* given by

$$H(s) = \frac{b_0}{s^4 + a_3s^3 + a_2s^2 + a_1s + a_0} \tag{5}$$

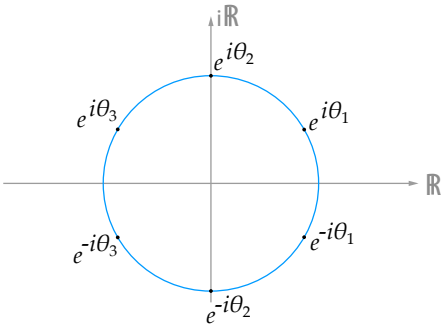
We can go back and forth between $H(s)$ and (1), each containing everything needed to construct the other. The frequency response of the filter is given by $|H(i\omega)|$, and the stability of the filter is revealed by the poles of $H(s)$ which are the roots of (4).

3 Butterworth

Consider the polynomial $p(x) = x^{2n} + 1$, where n is a positive integer. The fundamental theorem of algebra guarantees that $p(x)$ has exactly $2n$ roots. From the geometry of complex numbers, it is straightforward to work out that these roots are given by $e^{i\theta_k}$ and $e^{-i\theta_k}$ for $k = 1, 2, \ldots, n$, and with

$$\theta_k = \frac{\pi}{n}\left(k - \frac{1}{2}\right).$$

Here's the case $n = 3$.



The roots of $x^{2n} + 1$ being $e^{\pm i\theta_k}$ means that for any $x \in \mathbb{C}$,

$$x^{2n} + 1 = ab$$

where

$$a = (x - e^{i\theta_1})(x - e^{i\theta_2}) \cdots (x - e^{i\theta_n}),$$

and

$$b = (x - e^{-i\theta_1})(x - e^{-i\theta_2}) \cdots (x - e^{-i\theta_n}).$$

When $x \in \mathbb{R}$, a and b are conjugates. For any $a, b \in \mathbb{C}$, observe that $|ab| = |a||b|$, $\overline{ab} = \bar{a}\bar{b}$, and $|\bar{a}| = |a|$, and so $x \in \mathbb{R}$ implies

$$|x^{2n} + 1| = |(x - e^{i\theta_1})(x - e^{i\theta_2}) \cdots (x - e^{i\theta_n})|^2 \tag{6}$$

Setting $x = \omega/\omega_c$, and noting that $|i^m p| = |p|$ for any integer m and any $p \in \mathbb{C}$, we obtain

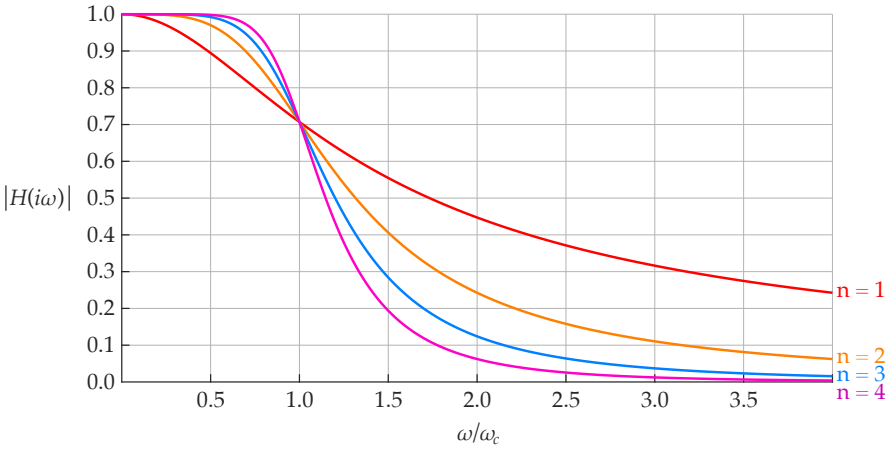
$$\left|\left(\frac{\omega}{\omega_c}\right)^{2n} + 1\right| = \frac{1}{\omega_c^{2n}}\left|(i\omega - \omega_c e^{i\tilde{\theta}_1})(i\omega - \omega_c e^{i\tilde{\theta}_2}) \cdots (i\omega - \omega_c e^{i\tilde{\theta}_n})\right|^2 \tag{7}$$

where $\tilde{\theta}_k = \theta_k + \frac{\pi}{2}$.

When the poles of $H(s)$ are given by $\omega_c e^{i\tilde{\theta}_k}$, and when $b_0 = \omega_c^n$, it follows from (7) that the frequency response is given by

$$|H(i\omega)| = \frac{1}{\sqrt{\left|(\omega/\omega_c)^{2n} + 1\right|}}. \tag{8}$$

It is surprising and beautiful that all the terms in (5) can boil down to such a simple expression. Note that $(\omega/\omega_c)^{2n}$ increases with ω, with a transition just before ω_c that gets sharper and sharper the greater the value of n. This causes $|H(i\omega)|$ to fall off a cliff just before ω_c that gets sharper as n increases.



Butterworth filter design consists of choosing n and ω_c to get a desired frequency response curve.

4 Discrete Time

The following discrete time system governs the evolution of $x_k \in \mathbb{C}$.

$$a_0 x_{k-4} + a_1 x_{k-3} + a_2 x_{k-2} + a_3 x_{k-1} + x_k = b_0 f_k \tag{9}$$

Similar to continuous time, we write this as $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + b_0 \mathbf{f}_k$ with

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ x_{k-1} \\ x_{k-2} \\ x_{k-3} \end{bmatrix}, \ \mathbf{A} = \begin{bmatrix} -a_3 & -a_2 & -a_1 & -a_0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \text{ and } \mathbf{f}_k = \begin{bmatrix} f_k \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Arithmetic shows that

$$\mathbf{x}_k = \mathbf{A}^k \mathbf{x}_0 + b_0 \sum_{l=0}^{k-1} \mathbf{A}^l \mathbf{f}_{k-l},$$

and so the stability of the system depends on the eigenvalues of $\mathbf{A}$, which are given by the roots λ of

$$\lambda^4 + a_3 \lambda^3 + a_2 \lambda^2 + a_1 \lambda + a_0 = 0.$$

If any of the eigenvalues is outside the unit circle in $\mathbb{C}$, our solutions can explode. We can alternatively find these eigenvalues by applying the z-transform

$$F(z) = \sum_{k=-\infty}^{\infty} f_k z^{-k} \tag{10}$$

to (9), obtaining

$$X(z) = H(z) * F(z),$$

with

$$H(z) = \frac{b_0 z^4}{z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0}.$$

The poles of $H(s)$ are the eigenvalues of $\mathbf{A}$.

5 Discrete Time from Continuous Time

Let $f_k = f(t_k)$ where $f(t)$ is a continuous signal and $t_k = kT$ with sampling period $T > 0$. We can approximate the Laplace transform of $f(t)$ as

$$\begin{aligned} F(s) &= \int_{-\infty}^{\infty} f(t)e^{-st}dt \\ &= \sum_{k=-\infty}^{\infty} \int_{t_k}^{t_{k+1}} f(t)e^{-st}dt \\ &\approx T \sum_{k=-\infty}^{\infty} f_k z^{-k} \\ &= TF(z) \end{aligned}$$

where $z = e^{sT}$ and $F(z)$ is the z-transform (10) of f_k. The key to moving between the s-plane of the Laplace transform and the z-plane of the z-transform would seem to be $z = e^{sT}$. However, for a number of practical reasons (beyond the scope of these notes), this is not how it's done.

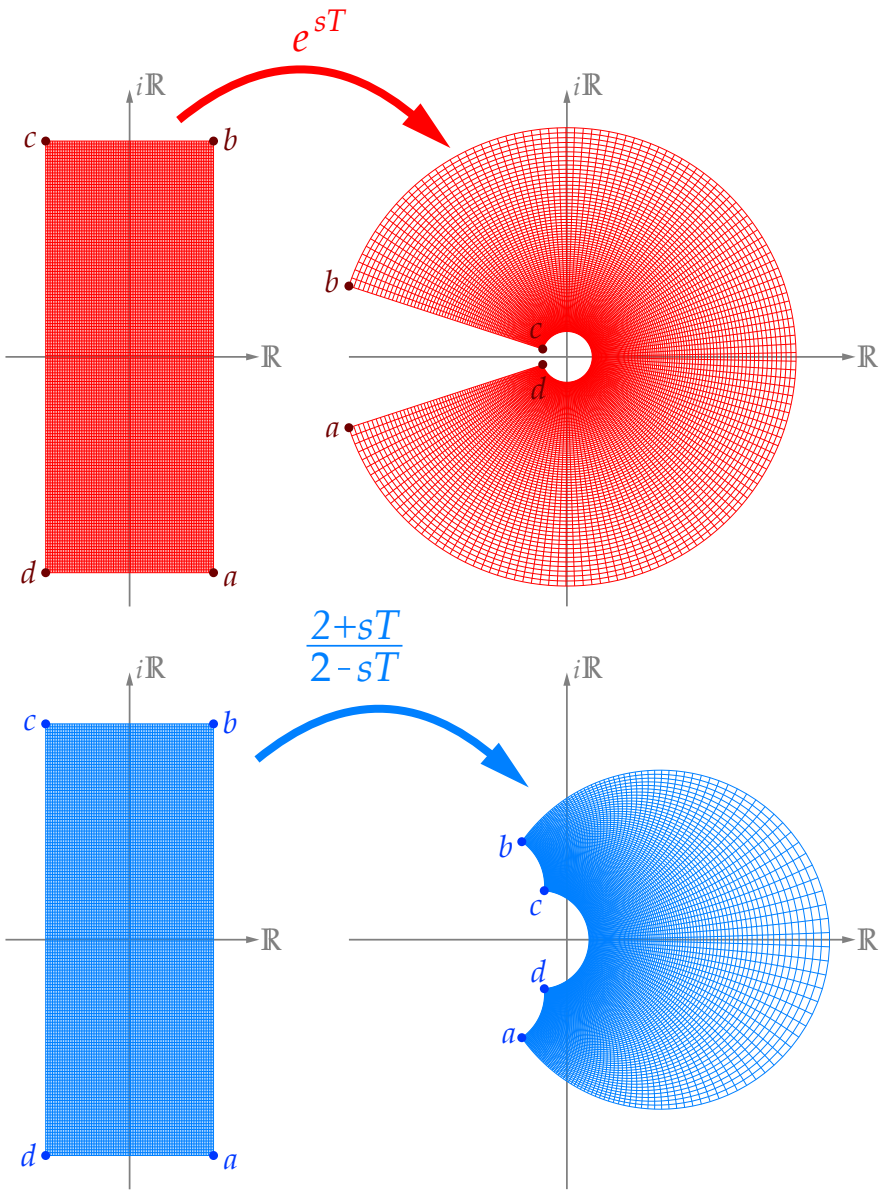
6 Building Discrete Filters

The standard way to build a discrete filter is to map the poles of a continuous filter into the z-plane using the bilinear transform

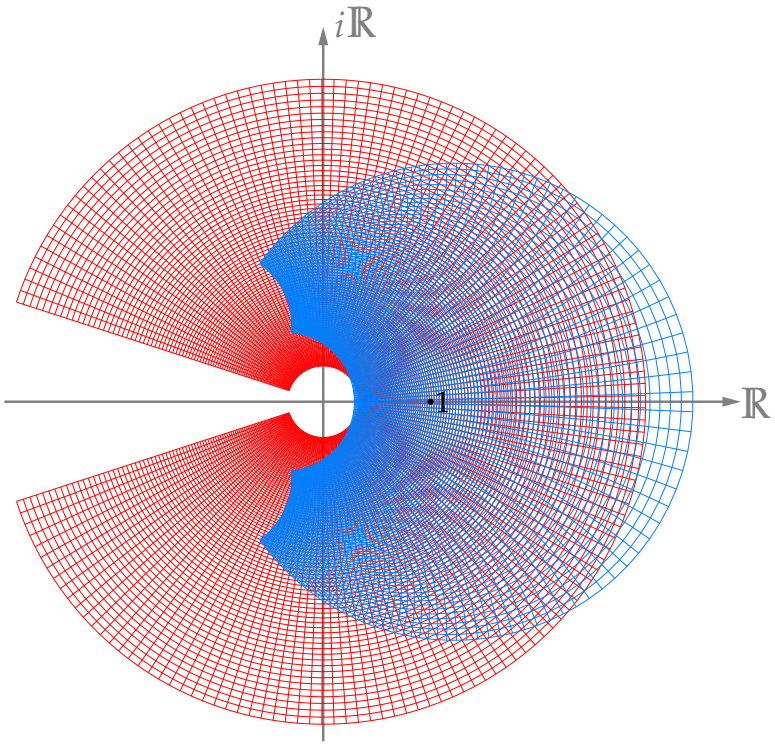
$$z = \frac{2 + Ts}{2 - Ts}. \tag{11}$$

The z-plane poles correspond to a digital filter with the same characteristics as the continuous filter.

Here's an illustration of e^{sT} and the bilinear transformation (11), each applied to the same rectangular region of $\mathbb{C}$. Corresponding corners are identified with the same letter.



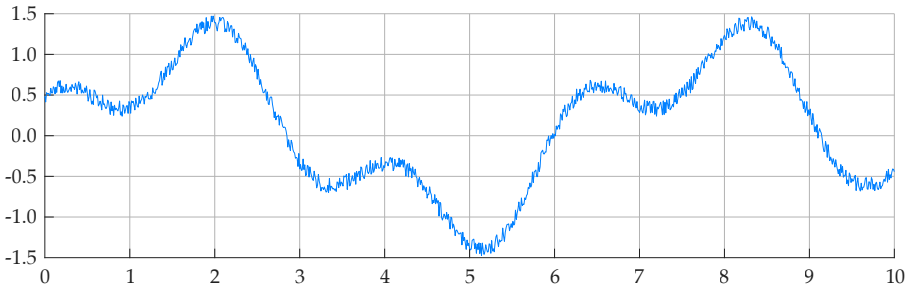
Overlay the grids from the previous page, and notice the interference pattern and the region of alignment around 1. The bilinear transformation (11) is the quotient of two linear functions that best fits $z = e^{sT}$ at this point.



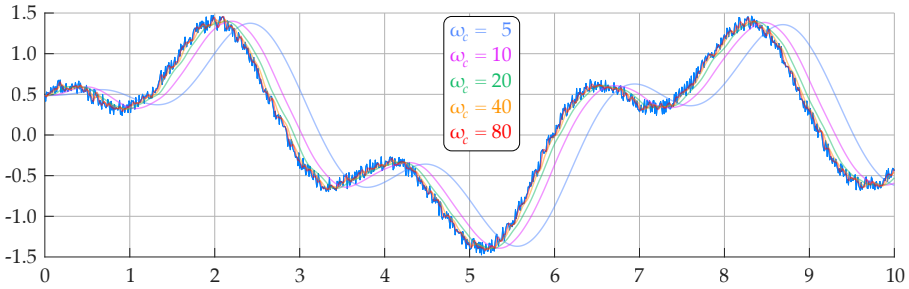
7 Example

Let's filter the following signal in Matlab.

```
rng(2);      % for repeatability
T = 0.01;    % sampling period [seconds]
t = 0:T:10; % 10 second time interval
f = sin(t) + 0.5*cos(3*t) + 0.2*(rand(size(t))-0.5);

figure('color',[1 1 1],'position',[79 514 1539 420]);
line('xdata',t,'ydata',f,'color',[0 0.5 1])
grid on
```



We build order 3 Butterworth filters with cutoff frequencies $\omega_c = 5$, 10, 20, 40, and 80. The results are as follows, with lower ω_c values giving smoother output, but with more latency.

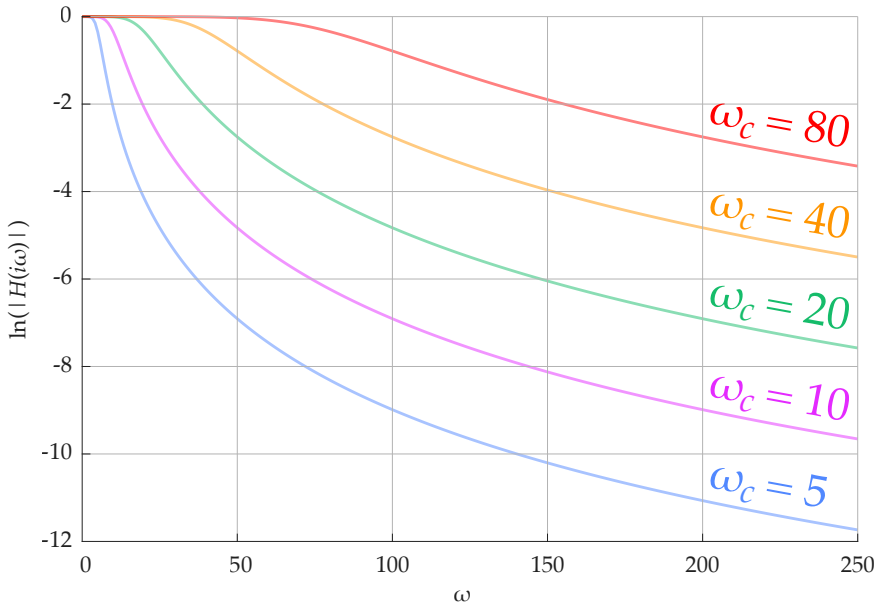


Note that ω_c is in radians per second, with the corresponding frequency in Hertz (cycles per second) given by $\omega_c/2\pi$.

Here is Matlab code for building the previous figures.

```
% Butterworth filter
wc = 5; % cutoff frequency [radians/second]
for wc = [5 10 20 40 80]
    N = 3;   % filter order

    % generate the N poles for the continuous filter
    theta = (pi/N)*(0.5 + (0:N-1)) + 0.5*pi;
    sPoles = wc*exp(1i*theta);

    % map these to the z plane using the bilinear
    % transformation
    zPoles = (2+sPoles*T)./(2-sPoles*T);

    % multiply the terms
    % (1-zPoles(1))*(1-zPoles(2))*...*(1-zPoles(N))
    B = 1;
    for k = 1:N
        B = [B 0]-zPoles(k)*[0 B];
    end
    B = real(B);  % imaginary part should be zero...
    B = B/sum(B); % normalize

    % construct the filtered signal
    g = NaN(size(t));
    gprev = f(1)*ones(1,N); % initialize prior g values
    for k = 1:numel(t)
        g(k) = (f(k) - B(2:end)*gprev')/B(1);
        gprev = [g(k) gprev(1:end-1)];
    end
    line('xdata',t,'ydata',g,'color',[1 0.5 0])
end
```

Here are the frequency response curves of these filters.



And here is the code that made this figure.

```
% visualize the frequency response
figure('color',[1 1 1])
WC = [5 10 20 40 80];
w = linspace(0,250,300);
for wc = WC
    Hmag = 1./sqrt((w/wc).^(2*N)+1);
    line('xdata',w,'ydata',log(Hmag),'color',[1 0.5 0])
end
grid on
```

References

[1] Oppenheim, A.,Willsky, A., Nawab, H., "Signals & Systems," 2nd Edition *Prentice Hall*, 1996.