Fast-Settling Filters

Martin Vicanek

January 1, 2025

1 Introduction

When measuring or tracking some slowly varying quantity in a noisy environment, one usually applies a lowpass filter to suppress the noise, thereby achieving a steady, more accurate reading. A desirable feature of the lowpass filter is that it should not ring nor creep, but settle quickly. In this note I present a class of IIR filters with optimal step response.

2 Fast-Settling Filters in Continuous Time

Consider a second order continuous-time lowpass filter. Its response function in Laplace space is given by [1]

$$H(s) = \frac{\omega_0^2}{\omega_0^2 + s\omega_0/Q + s^2}. \tag{1}$$

Here, s denotes the Laplace variable, ω_0 is the cutoff frequency and Q is the quality factor.

The step response corresponding to the transfer function in eq.(1) is shown in figure 1 for three values of Q. In order to facilitate a fair comparison, ω_0 has been chosen in each case such that the curves reach half the step size at the same time. We may call this time the *(effective) response time* of a given filter. In other words, the time axis in figure 1 represents the time normalized to each filter's response time.

For $Q > \frac{1}{2}$ the step response overshoots and possibly rings for some time before it settles, whereas for $Q < \frac{1}{2}$ it creeps more slowly than necessary towards steady state. Obviously there is a trade-off between reaching the final value fast and accepting a certain amount of overshoot.

This concept may be generalized for higher order lowpass filters. A lowpass of order $2K$ has the following response in Laplace space:

$$H(s) = \prod_{k=1}^{K} \frac{\omega_k^2}{\omega_k^2 + s\omega_k/Q_k + s^2}. \tag{2}$$

Eq.(2) represents a number K second order lowpass filters in series, with cutoff frequencies ω_k and Q-factors Q_k. The shape of the step response is determined by the values of the Q_k

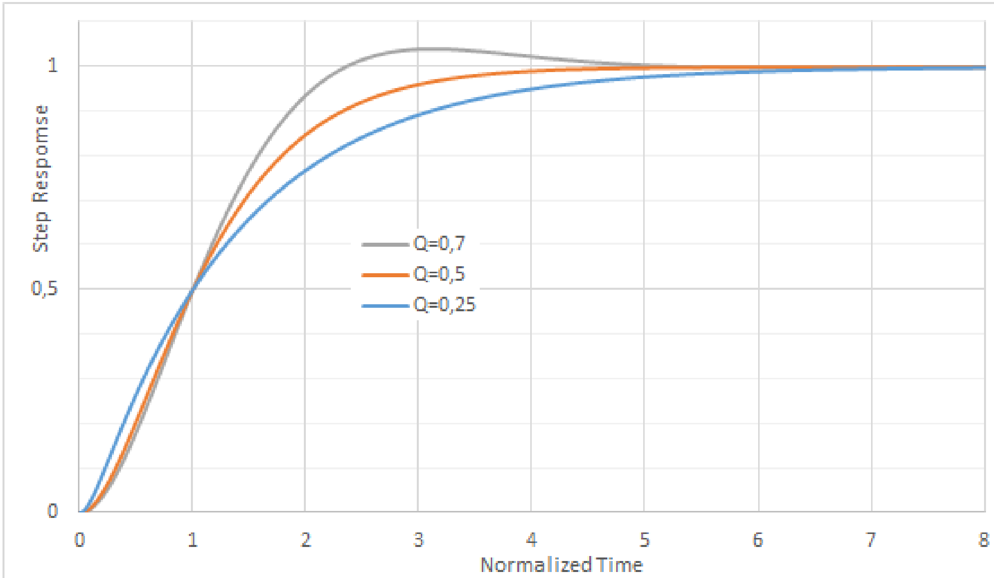

Figure 1: Step response of a second order lowpass filter, eq.(1), for various values of the Q parameter. Time is normalized to the instant where the curves reach half the step size.

and the ratios of the cutoff frequencies ω_k/ω_1. Hence, apart from an overall scaling, there are $2K-1$ parameters which we may adjust to obtain a fast-settling filter. If we tolerate an overshoot or ringing amplitude δ in units of the step height, then the step response with fastest settling will exhibit $2K-1$ half oscillations after the first overshoot with exactly that amplitude and then quickly reach steady state. This equiripple settling condition may be used to determine the filter coefficients for a given ripple tolerance δ and filter order. The following examples illustrate the point.

Figure 2 shows the "fastest" step response for $\delta = 0.01$ and various filter orders. The time when the step response first crosses the step height (equal to 1 in figure 2) may be viewed as the settling time. After the first crossing, the curves stay within the tolerance band specified by δ with regard to the step height. Figure 3 zooms in on the settling stage. Obviously, the settling time decreases with increasing filter order.

Figure 4 shows optimum step responses of an 8th order lowpass filter for different tolerance levels. The curves look quite similar at this resolution. There is a mild increase in the settling time when the tolerance bounds get tighter. Figure 5 Reveals more details of the settling stage.

Figure 6 shows the settling time for various filter orders and tolerances. This summarizes the results exemplified in figures 2 to 5: The settling time is considerably lower for higher filter orders, especially so at small tolerance levels. For a given filter order, the settling time increases with decreasing tolerance, as one would expect.

Table 1 lists filter parameters obtained for some selected values of δ and filter orders 2,4,6, and 8 used in figures 2 to 6. The data has been obtained by a nonlinear optimization method devised for that purpose. Values for ω_k are normalized such that the filter response time is

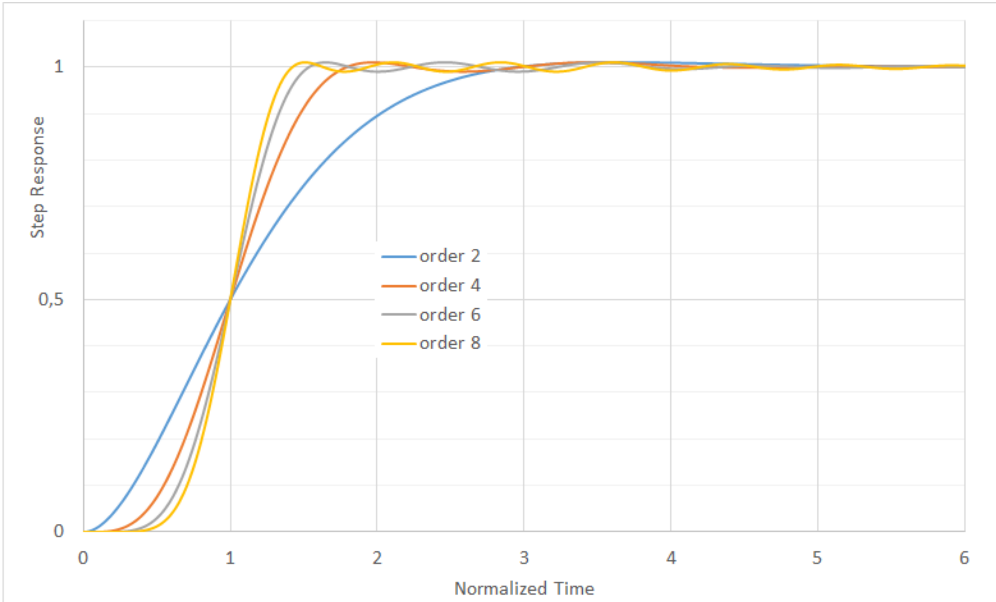

Figure 2: Step response of optimum lowpass filters for 0.01 overshoot tolerance and orders 2,4,6, and 8.

equal to 1.

3 Digital Implementation

The continuous-time filter may be implemented digitally using the bilinear transformation (BLT)[2], preferrably using the SVF TPT topology [3]. The step response will be similar provided that the cutoff frequencies are well below Nyquist. However, it is possible to design a digital filter with exactly the same step response as the continuous-time filter. The method may be called step invariance, similar to the familiar impulse invariance.

To this end, expand the response function in eq.(2) into partial fractions,

$$H(s) = \sum_k \left(\frac{A_k}{s - s_k} + \frac{\overline{A_k}}{s - \overline{s_k}} \right), \tag{3}$$

where s_k and $\overline{s_k}$ are the complex conjugate poles,

$$s_k = -\frac{\omega_k}{2Q_k} + i\omega_k \sqrt{1 - \frac{1}{4Q_k^2}}, \tag{4}$$

and A_k and $\overline{A_k}$ denote the respective residues,

$$A_k = \frac{P}{(s_k - \overline{s_k}) \prod_{j \neq k}(s_k - s_j)(s_k - \overline{s_j})}, \quad P = \prod_k \omega_k^2. \tag{5}$$

The step response of $H(s)$ in eq.(3) is the sum of step responses of each partial fraction. In the appendix, we show the step-invariant transformation in detail. The resulting digital filter

Order 2

δ	ω_0	Q
10^{-2}	1.525667	0.605265
10^{-3}	1.596930	0.549280
10^{-4}	1.629077	0.528286
10^{-5}	1.645663	0.518281
10^{-6}	1.655190	0.512764
10^{-7}	1.661123	0.509409

Order 4

δ	ω_1	Q_1	ω_2	Q_2
10^{-2}	2.087833	0.583993	3.643149	1.527235
10^{-3}	2.459946	0.547924	3.677486	0.959346
10^{-4}	2.756932	0.529326	3.681537	0.762557
10^{-5}	2.974557	0.519367	3.680945	0.669984
10^{-6}	3.131173	0.513604	3.679869	0.618881
10^{-7}	3.244468	0.510028	3.678854	0.587697

Order 6

δ	ω_1	Q_1	ω_2	Q_2	ω_3	Q_3
10^{-2}	2.490852	0.560085	3.873728	1.114950	6.002095	3.954038
10^{-3}	2.890925	0.541045	4.112063	0.887498	6.096995	1.819061
10^{-4}	3.327161	0.527289	4.345274	0.747157	6.060397	1.224078
10^{-5}	3.716766	0.518842	4.553588	0.667584	6.002232	0.964593
10^{-6}	4.043417	0.513570	4.731019	0.619855	5.947760	0.825152
10^{-7}	4.310154	0.510145	4.878360	0.589433	5.902072	0.740695

Order 8

δ	ω_1	Q_1	ω_2	Q_2	ω_3	Q_3	ω_4	Q_4
10^{-2}	2.957235	0.542298	4.210168	0.896090	6.106563	1.844445	8.232505	10.51959
10^{-3}	3.237222	0.534594	4.426414	0.815398	6.309273	1.469757	8.566109	3.388531
10^{-4}	3.700966	0.524745	4.735882	0.721816	6.453059	1.138351	8.578797	1.956007
10^{-5}	4.177915	0.517768	5.060828	0.657944	6.579305	0.939250	8.504146	1.405201
10^{-6}	4.619034	0.513119	5.369818	0.616196	6.695730	0.817743	8.409878	1.126162
10^{-7}	5.009863	0.509971	5.649021	0.588246	6.801813	0.739562	8.318633	0.962260

Table 1: Fast-settling lowpass filter coefficients.

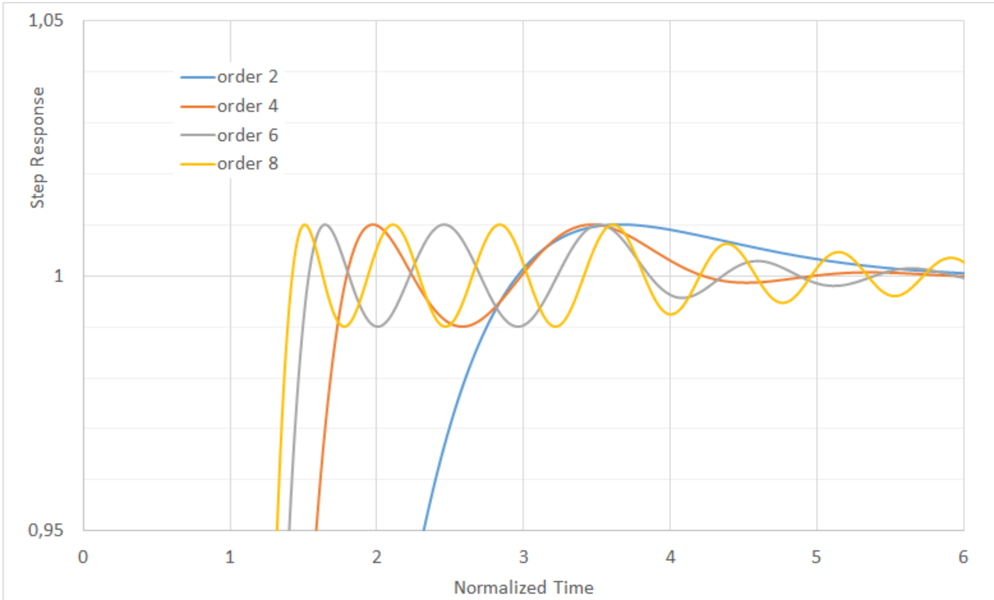

Figure 3: Step response as in figure 2, zoomed in on the settling stage.

response is

$$H(z) = \sum_k \left(\frac{b_k z^{-1}}{1 + a_k z^{-1}} + \frac{\overline{b_k} z^{-1}}{1 + \overline{a_k} z^{-1}} \right), \tag{6}$$

where the coefficients a_k and b_k (and their c.c. counterparts $\overline{a_k}$ and $\overline{b_k}$) are given in terms of the complex poles s_k,

$$a_k = -e^{s_k T}, \quad b_k = -A_k \frac{1 + a_k}{s_k T} \tag{7}$$

T denotes the sampling interval (inverse of the sample rate).

Equation (6) represents a sum of parallel filters, each with a single complex pole. A filter with a single pole may be implemented recursively as

$$y_{k,n+1} = b_k x_n - a_k y_{k,n}, \tag{8}$$

where x_n denotes the input stream of (real valued) samples for $n = 1, 2, 3, \ldots$, and $y_{k,n}$ the (complex valued) output stream of the kth filter. Splitting real and imaginary parts $a_k = a'_k + ia''_k$ and likewise for b_k and $y_{k,n}$ yields two coupled equations

$$\begin{aligned} y'_{k,n+1} &= b'_k x_n - a'_k y'_{k,n} + a''_k y''_{k,n} \\ y''_{k,n+1} &= b''_k x_n - a'_k y''_{k,n} - a''_k y'_{k,n}. \end{aligned} \tag{9}$$

The single-pole filters come in c.c. pairs, so imaginary parts cancel and real parts double for each c.c. pair, hence it is sufficient to consider only one of each c.c. poles and take twice the sum of the real filter outputs $2 \sum_k y'_{k,n}$.

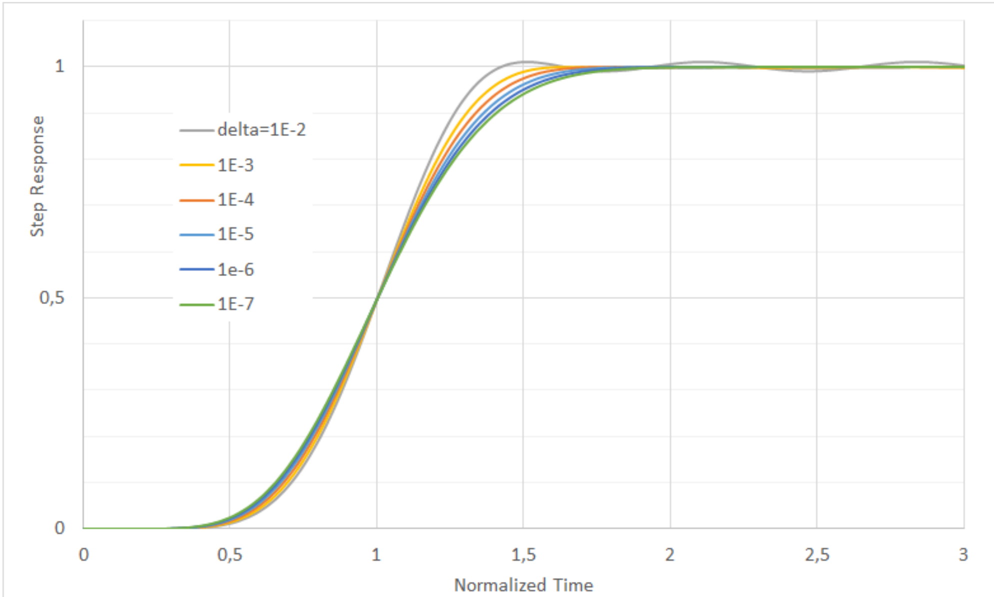

Figure 4: Step response of optimum lowpass filters of order 8 for various overshoot tolerance levels.

4 Discussion

We have seen that IIR filters — both continuous and discrete time — may be designed to settle fast. One might ask why not take FIR filters in the first place, as these obviously settle within a finite time. Moreover, FIR filters may easily be designed linear phase, and with a step response that does not overshoot at all. One disadvantage is that FIR filters only exist in the discrete time (i.e. digital) domain, but not in the continuous time (aka analog) domain. Another drawback is that for low cutoff, which is quite typical in many applications, FIR filters are computationally more expensive with respect to both CPU and and memory requirements than IIR filters.

There is perhaps one exception, the running average and related filters [4, 5], which may be evaluated recursively, hence of comparable efficiency as IIR filters. These filters are only viable for integer kernel sizes, which means that not all cutoffs are possible.

It is interesting to note that the fast setting IIR filters presented here have a flat group delay of approximately the response time over most of the passband. In fact they are similar to Bessel filters, although not exactly the same. Another interesting and related aspect is the (normalized) settling time. For strictly linear phase, it would be exactly 2. Figure 6 shows that for certain parameter choices, the settling time is even less than that. This is somewhat surprising because the step response is, after all, a superposition of exponential decays (with some oscillation) — objects with more or less long tails. The secret is in the subtle interference, as a result of our equiripple settling condition.

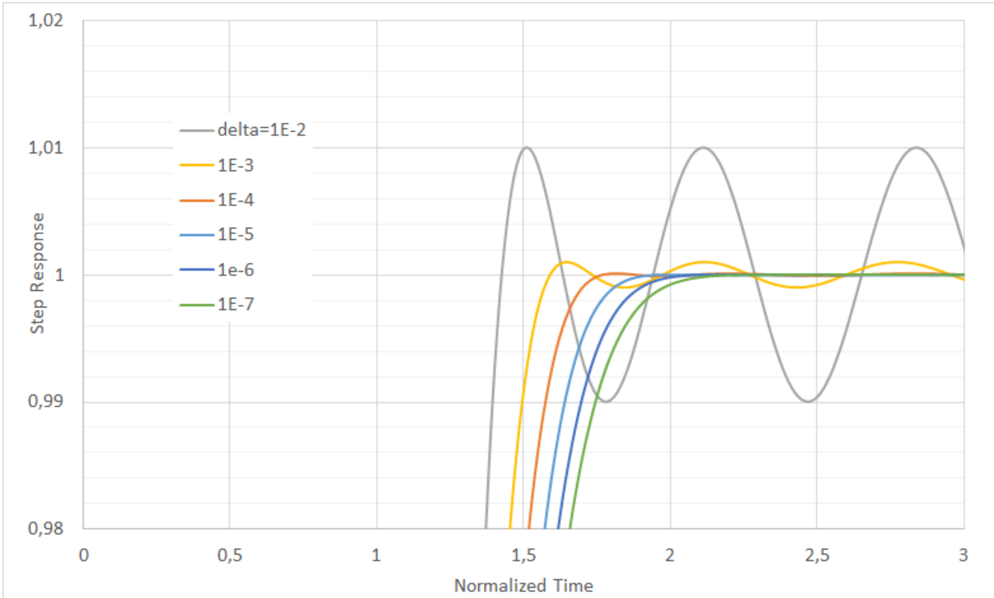

Figure 5: Step response as in figure 4, zoomed in on the settling stage.

5 Conclusion

IIR filters with ideally fast settling properties have been presented for a range of design parameters and choices. Such filters are computationally efficient and easy to implement once the coefficients are known. Possible applications in audio processing include level meters and envelope followers in general.

6 Appendix: Step-Invariant Transform

This section explains how to transform a continuous-time filter to a discrete-time (aka digital) filter with the same step response. Here is the recipe:

1. Start with the filter response in Laplace space $H(s)$. The step response is $H(s)/s$.
2. Transform this to the time domain.
3. Evaluate the result at sampling points nT and perform a z-transform with respect to n.
4. Multiply the result with $(1 - z^{-1})$ to get the filter response in z-space.

We will apply this recipe to a single-pole filter

$$H(s) = \frac{1}{s - s_0}. \tag{10}$$

Here, the pole s_0 may be complex, however its real part must be negative for stability. The step response of the above is

$$\frac{H(s)}{s} = \frac{1}{(s - s_0)s} = \frac{1}{s_0}\left(\frac{1}{s - s_0} - \frac{1}{s}\right), \tag{11}$$

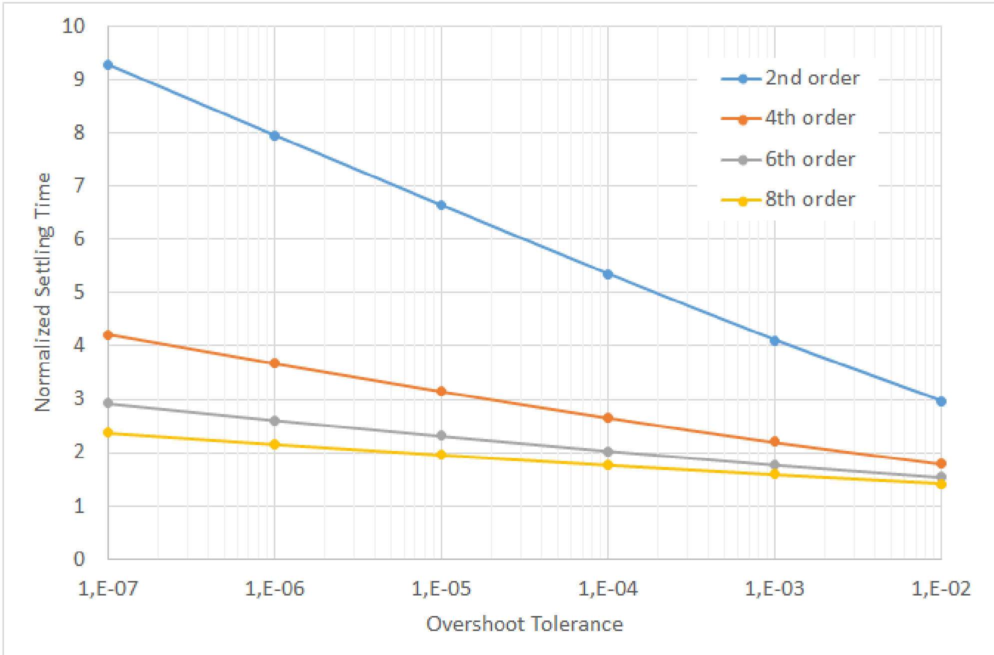

Figure 6: Settling time in units of the effective response time for various filter orders and tolerances.

Taking the inverse Laplace transform we obtain the step response in the time domain:

$$\frac{e^{s_0 t} - 1}{s_0} \tag{12}$$

At sampling points nT this yields:

$$\frac{z_0^n - 1}{s_0}, \quad \text{with} \quad z_0 = e^{s_0 T} \tag{13}$$

Taking the z-transform results in

$$\frac{1}{s_0}\left(\frac{1}{1 - z_0 z^{-1}} - \frac{1}{1 - z^{-1}}\right) \tag{14}$$

Multiplying this expression with $1 - z^{-1}$ yields the filter response in z-space:

$$H(z) = \frac{z_0 - 1}{s_0} \frac{z^{-1}}{1 - z_0 z^{-1}}. \tag{15}$$

The denominator $(1 - z_0 z^{-1})$ represents a single-pole filter, while the numerator z^{-1} ads a delay by one sample. The term $(z_0 - 1)/s_0$ is just a constant factor.

References

[1] U. Tietze and Ch. Schenk, Halbleiter-Schaltungs-Technik, 16th ed. 2019, Chapter 12 Aktive Filter.

[2] Robert Bristow-Johnson, Audio EQ Cookbook. https://www.w3.org/TR/audio-eq-cookbook/

[3] Vadim Zavalishin, The Art of VA Filter Design, rev. 2.1.2, 2020. https://www.native-instruments.com/fileadmin/ni_media/downloads/pdf/VAFilterDesign_2.1.2.pdf

[4] Geraint Luff, Cascaded box-filter smoothing filters, Signalsmith Audio Ltd. 2022. https://signalsmith-audio.co.uk/writing/2022/cascaded-box-filter-smoothing/

[5] M. Vicanek, Fast FIR Filters, in preparation, 2025.