Time Series/Data Processing and Analysis (MATH 587/GEOP 505)

Brian Borchers and Rick Aster

November 8, 2013

Notes on Deconvolution

We have seen how to perform convolution of discrete and continuous signals in both the time domain and with the help of the Fourier transform. In these lectures, we'll consider the problem of reversing convolution or deconvolving an input signal, given an output signal and the impulse response of a linear time invariant system.

We begin with the equation

$$d(t) = g(t) * m(t) \tag{1}$$

where $d(t)$ and $g(t)$ are known. Our goal is to solve for the unknown $m(t)$.

Although there's no obvious way to use the convolution integral to solve this equation, the equation becomes much easier to solve in the frequency domain. By the convolution theorem,

$$D(f) = G(f)M(f). \tag{2}$$

Thus

$$M(f) = \frac{D(f)}{G(f)}. \tag{3}$$

Once we have $M(f)$, we can invert the Fourier transform to obtain $m(t)$. Similarly, if we have discrete time signals and

$$d_n = (g_n * m_n)\Delta t \tag{4}$$

then

$$D_k = G_k M_k \Delta t \tag{5}$$

for $k = 0, 1, \ldots, N-1$. Solving for M_k, we get

$$M_k = \frac{D_k}{G_k \Delta t}. \tag{6}$$

Once we have the vector M we can invert the discrete Fourier transform to obtain m_n. This simple approach to solving the deconvolution problem is called **spectral division**.

Unfortunately, this method seldom works in practice. The first problem is that denominator in (3) might be zero, at least for some frequencies. In that case, $M(f)$ is undefined, and we can't invert the Fourier transform to obtain $m(t)$. Another way of looking at this is to consider what output the system will produce for sine waves at different frequencies. If the system produces zero output for a sine wave at a particular frequency f_0, then it's clear that we can't solve the deconvolution problem for any input signal that contains a sine wave at frequency f_0 because there's no evidence of this sine wave in the output!

What about noise? First, suppose that noise $n(t)$ is mixed with the true signal before the convolution. In that case we have

$$d(t) = g(t) * (m(t) + n(t)) \tag{7}$$

or

$$D(f) = G(f)(M(f) + N(f)). \tag{8}$$

If we perform spectral division, we obtain

$$M(f) + N(f) = \frac{D(f)}{G(f)}. \tag{9}$$

In this situation, the deconvolution hasn't made the noise any worse than it was before the deconvolution. Later in the course we'll discuss approaches to removing noise with a known frequency spectrum from such a signal.

Things get trickier if the noise is added after the convolution with $g(t)$. In that case, we have

$$d(t) = g(t) * m(t) + n(t) \tag{10}$$

or

$$D(f) = G(f)M(f) + N(f). \tag{11}$$

If we try to perform spectral division, we end up with

$$M(f) + \frac{N(f)}{G(f)} = \frac{D(f)}{G(f)}. \tag{12}$$

The $N(f)/G(f)$ term will introduce noise into the recovered signal. At frequencies where $G(f)$ is small but nonzero, the deconvolution process can greatly increase the magnitude of the noise.

Various techniques have been developed to deal with this noise. The basic idea is to avoid division by zero by somehow modifying the denominator in (6). This **regularizes** the deconvolution problem. In performing the regularization, we want to do as little as possible to frequencies where the noise is insignificant, while damping out the noise at frequencies where it is larger than the signal. Because the DFT of a real input signal is always Hermitian (i.e. $M_k = M^*_{N-k}$)

it is important that we perform the regularization in a way that produces a Hermitian M sequence and a real signal m_n.

For example, we might try

$$M_k = \frac{D_k}{(G_k + \lambda)\Delta t} \tag{13}$$

where λ is a small positive real number. When G_k is much larger than λ, then this will have little effect on M_k. However, when G_k is very small compared to λ, this will effectively zero out the response at frequency k. One problem with this scheme is that if $G_k = -\lambda$, we can still get division by zero. It would obviously be better to work with the absolute value of G_k.

A scheme called **water level regularization** is widely used in geophysics. Since problems only occur at frequencies where $|G(f)|$ is small, we pick a critical level w and adjust $G(f)$ only when $|G(f)| \leq w$. At frequencies where $|G(f)| > w$ we simply perform spectral division. This has the advantage of not altering the spectral division method at good frequencies. At frequencies where $|G(f)|$ is small, we need to replace $G(f)$ with something that isn't too small. We could simply use w, but it is slightly better to use a complex number that at least has the same phase as $G(f)$. So we, use

$$\hat{G}(f) = w\frac{G(f)}{|G(f)|}. \tag{14}$$

If $G(f)$ is exactly zero this still causes problems! In that case, we'll use $\hat{G}(f) = w$. In discrete time, the water level deconvolution scheme can be written as

$$\hat{M}_k = \frac{D_k}{\hat{G}_k \Delta t} \tag{15}$$

where

$$\hat{G}_k = \begin{cases} G_k & |G_k| > w \\ \frac{wG_k}{|G_k|} & 0 < |G_k| \leq w \\ w & G_k = 0. \end{cases} \tag{16}$$

Note that $\hat{M}_k$ will be a Hermitian sequence. When we invert the transform to obtain m_n, we'll get back a real signal.

In order for the water level regularization to work we need to make sure that $w\Delta t$ is somewhat larger than $|N_k|$. If w is too large, then we simply get back d_n scaled down by a factor of w. If w is too small, than the result will be overly noisy, often at higher frequencies where $|G(f)|$ is smaller.

In the following example, A small amount of noise in the data makes spectral division unstable, but water level regularization produces very good results.

The input signal is $m(t) = te^{-t}$ and the impulse response is $g(t) = e^{-5t}\sin(10t)$. See Figures 1 and 2. This signal was sampled at intervals of $\Delta t = 0.01$ seconds. The convolved signal $d(t)$ is 10 seconds long, so there are 1001 samples.

Random noise was added to the signal with a normal distribution with mean 0 and standard deviation 0.0001. Figure 4 shows the noisy data. Figure 5 shows

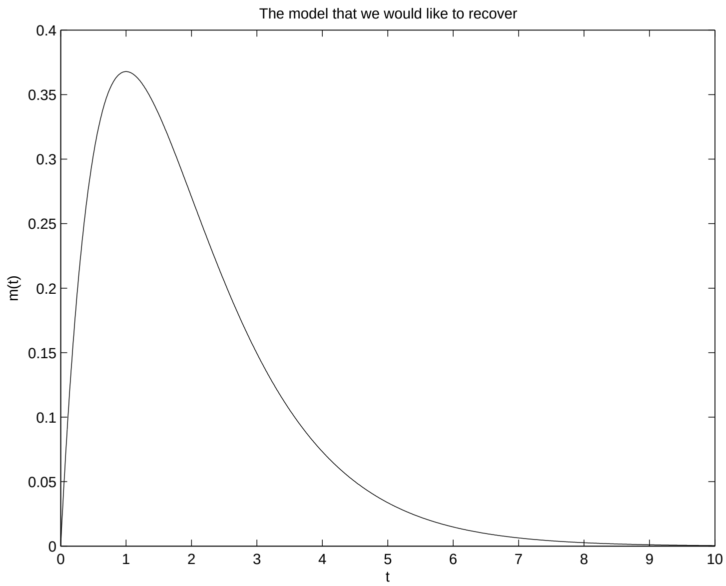

Figure 1: The input signal.

the unfortunate result of simple spectral division- the high frequency noise is greatly increased in amplitude.

The white noise that we added to the signal has approximately equal energy at all frequencies. Recall Parseval's theorem for the DFT,

$$\sum_{j=0}^{N-1} |x_j|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X_k|^2. \tag{17}$$

Since $N = 1001$, we expect the $|N_k|$ values of our noise to be about $\sqrt{1001}$ or roughly 30 times larger than the $|n_n$ values. Thus a typical value of $|N_k|$ should be about 0.003. In order to make the values of $w\Delta t$ larger than 0.003, we'd like to have $w \geq 1$.

Figures 6 through 8 shows the results obtained with $w = 0.1$, $w = 1$, and $w = 10$. Although the solution is under regularized at $w = 0.1$, the solution is quite good by the time we get to $w = 10$.

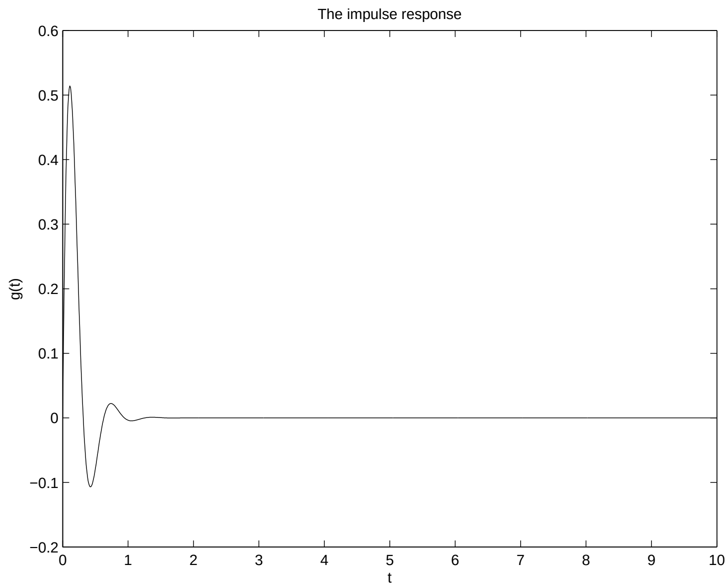

Figure 2: The impulse response.

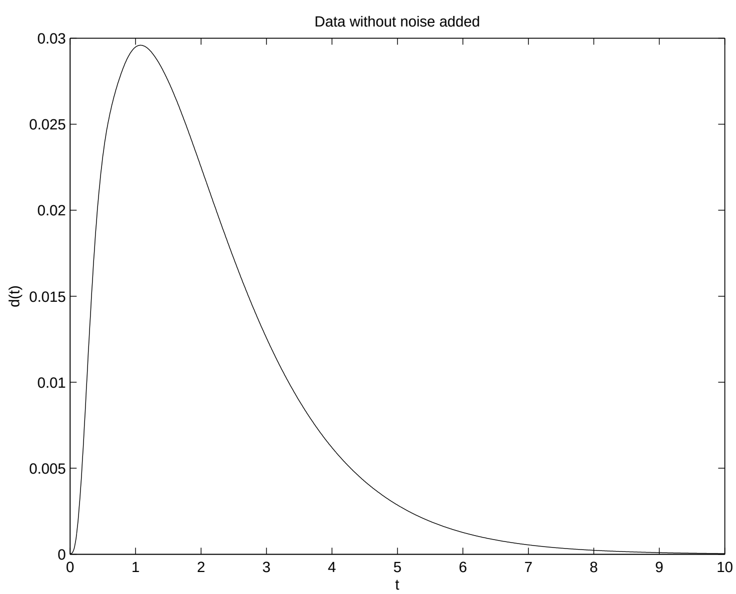

Figure 3: Clean data.

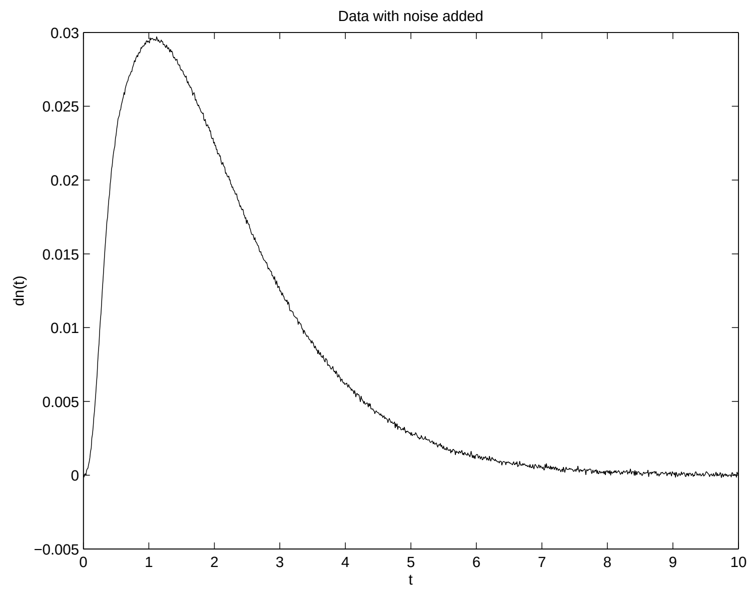

Figure 4: The data with a small amount of noise added.

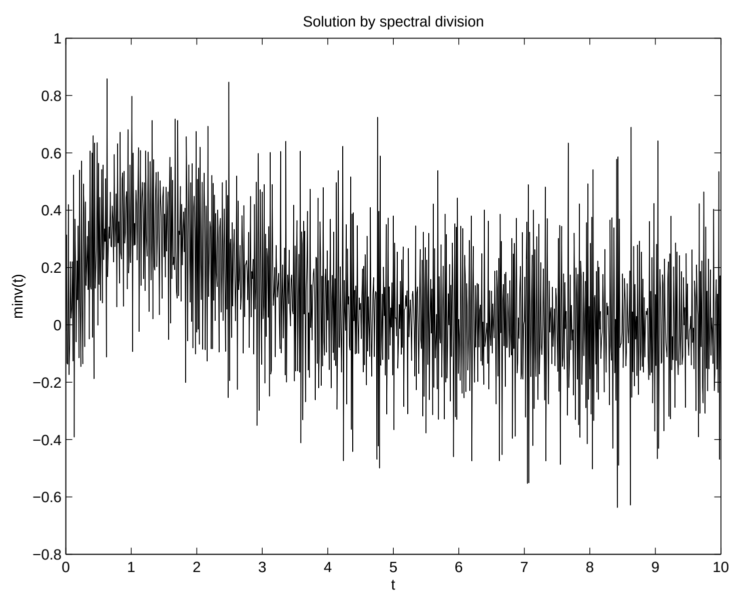

Figure 5: Deconvolution by spectral division, no regularization.

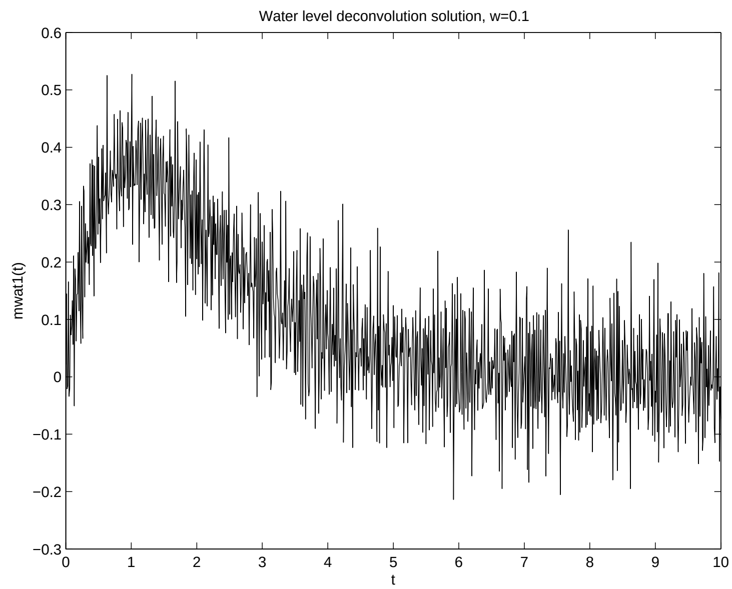

Figure 6: Water level solution, $w = 0.1$.

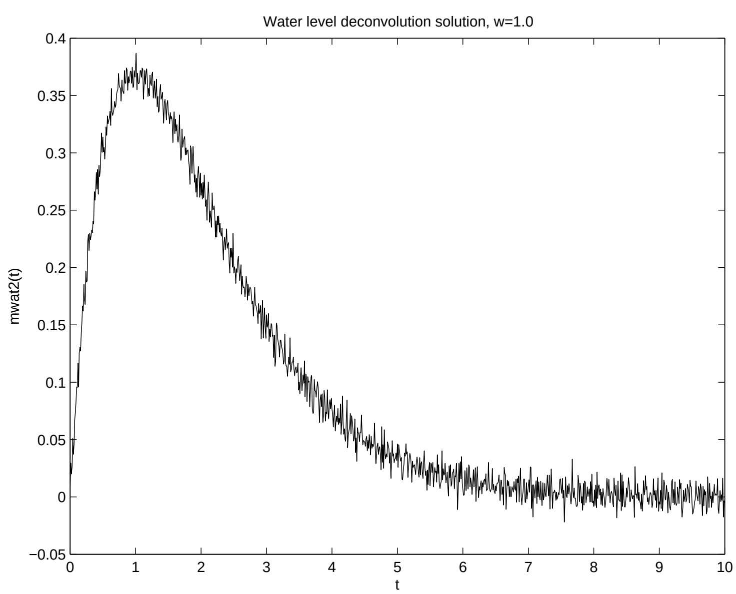

Figure 7: Water level solution, $w = 1.0$.

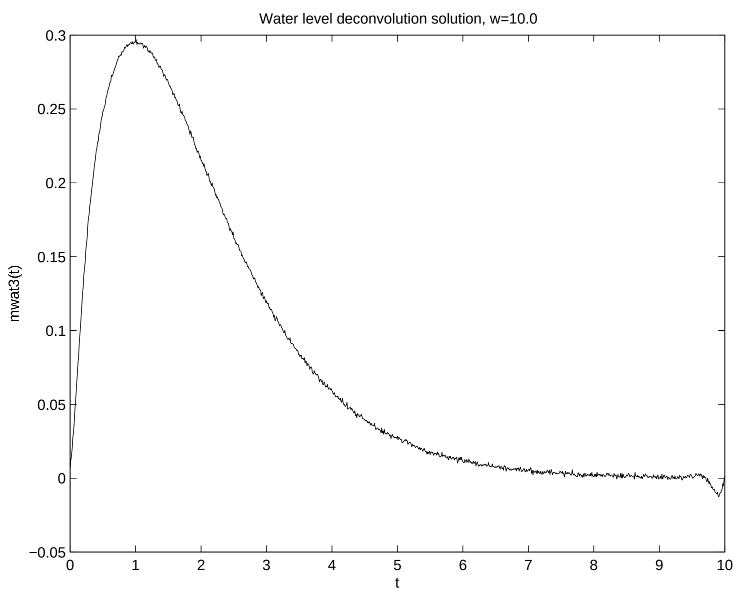

Figure 8: Water level solution, $w = 10.0$.

In Tikhonov regularization, we use

$$\hat{M}_k = \frac{G_k^* D_k}{(G_k^* G_k + \lambda)\Delta t} \tag{18}$$

where λ is a small positive parameter. This is similar to (13), in that when $|G_k|$ is much larger in magnitude than λ, we get essentially (6). However, when $|G_k|$ is much smaller than λ, M_k is reduced in magnitude. It's not hard to show that if M_k is obtained by Tikhonov regularization then M_k will be Hermitian. Furthermore, the denominator in this formula can never be 0.

The size of the factor

$$\frac{G_k^* N_k}{(G_k^* G_k + \lambda)\Delta t} \tag{19}$$

determines whether a noise frequency k will be effectively eliminated from the deconvolved signal. To get rid of the noise, we want

$$|G_k^* N_k| < \lambda \Delta t. \tag{20}$$

This gives us a very simple criteria for picking λ. We'll discuss more sophisticated methods for picking λ in the inverse problems course.

Returning to our earlier example, we know that $|N_k|$ is typically about 0.003, while $|G_k|$ is typically around 3. Thus we need $\lambda \Delta t > 0.01$ or $\lambda > 1$ to cover the noise. Figures 9 through 11 show the effect of different values of the regularization parameter λ.

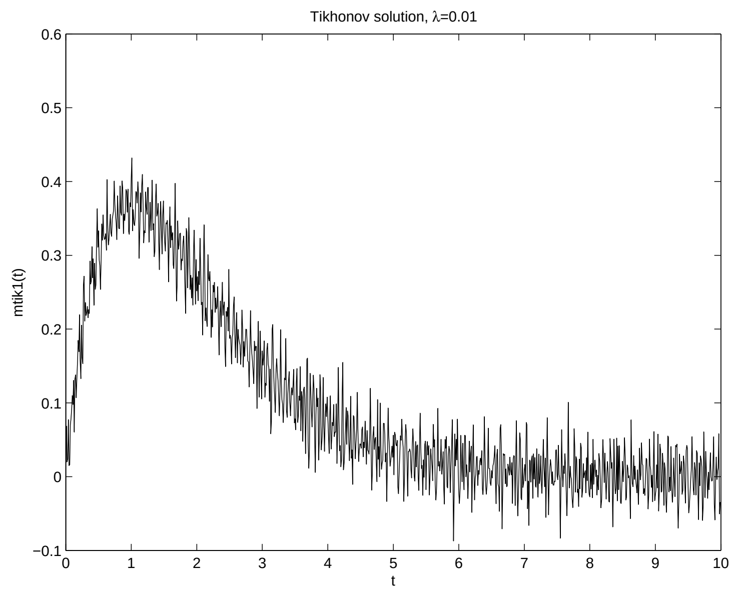

Figure 9: Deconvolution with Tikhonov regularization, $\lambda = 0.01$.

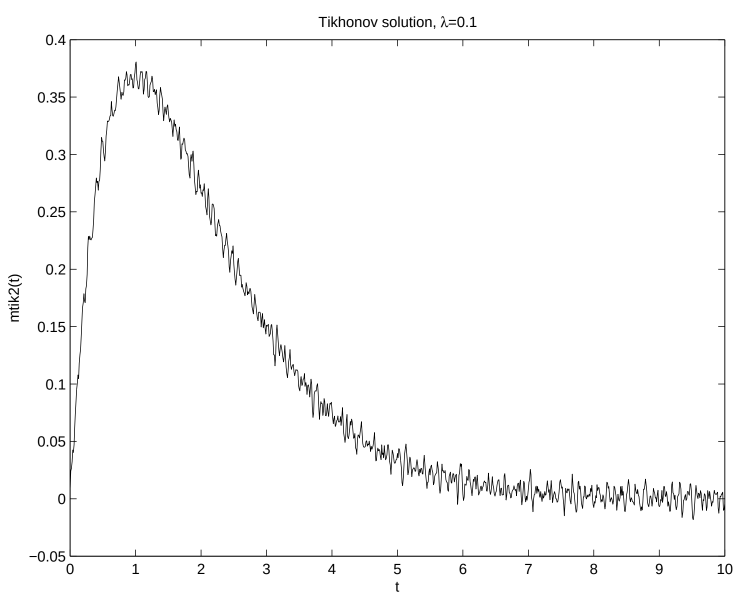

Figure 10: Deconvolution with Tikhonov regularization, $\lambda = 0.1$.

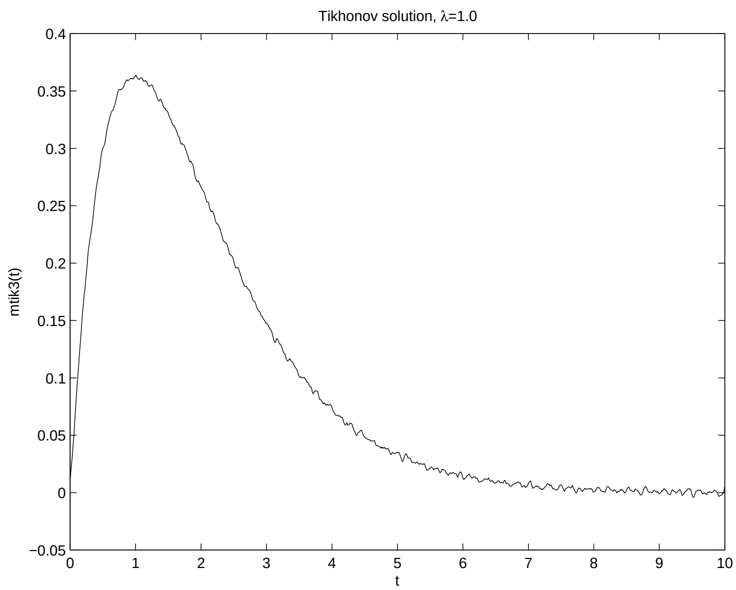

Figure 11: Deconvolution with Tikhonov regularization, $\lambda = 1.0$.

It can be shown that Tikhonov regularization minimizes

$$\min \|G \cdot M - D\|_2^2 + \lambda \|M\|_2^2. \tag{21}$$

By Parseval's theorem, this is equivalent to minimizing

$$\min \|g * m - d\|_2^2 + \lambda \|m\|_2^2. \tag{22}$$

The objective function is a weighted sum of a term that measures how well the model m fits the data d and a term that measures the energy of the model m. Tikhonov regularization is effectively picking the smallest energy signal that fits the data reasonably well, with the relative balance of these two factors controlled by the regularization parameter λ.

An alternative formulation of Tikhonov regularization sets a limit δ on the data misfit and then minimizes $\|m\|_2$.

$$\begin{array}{lrcl} \min & & \|m\|_2 & \\ & \|g * m - d\|_2 & \leq & \delta. \end{array} \tag{23}$$

There are situations in which other kinds of regularization are appropriate. We'll consider an example in which a controlled source (e.g. a vibroseis truck) is used to send a seismic wave down into the earth. The wave bounces back from reflecting layers at various depths and a seismograph of the reflected signal is recorded. We'd like to recover the depths of these reflecting layers.

Here, $g(t)$ is the known source signal, $d(t)$ is the recorded seismograph, and $m(t)$ is the unknown. The reflector should appear in $m(t)$ as scaled delta functions, with a reflect at time "depth" $t_0/2$ appearing as a scaled $\delta(t - t_0)$.

In this case, we want m to be a simple sequence of spikes. Rather than using Tikhonov regularization to minimize $\|m\|_2$, we want to minimize the number of nonzero entries in m. Let $\|m\|_0$ be the number of nonzero entries in m. Than we can formulate our regularization problem as

$$\begin{array}{lrcl} \min & & \|m\|_0 & \\ & \|g * m - d\|_2 & \leq & \delta. \end{array} \tag{24}$$

Unfortunately, these kinds of optimization problems are extremely difficult to solve.

A surprisingly effective alternative is to instead minimize

$$\|m\|_1 = \sum_{j=1}^{n} |m_j|. \tag{25}$$

The regularization problem is then

$$\begin{array}{lrcl} \min & & \|m\|_1 & \\ & \|g * m - d\|_2 & \leq & \delta. \end{array} \tag{26}$$

It turns out that these problems can be effectively solved by convex optimization techniques.

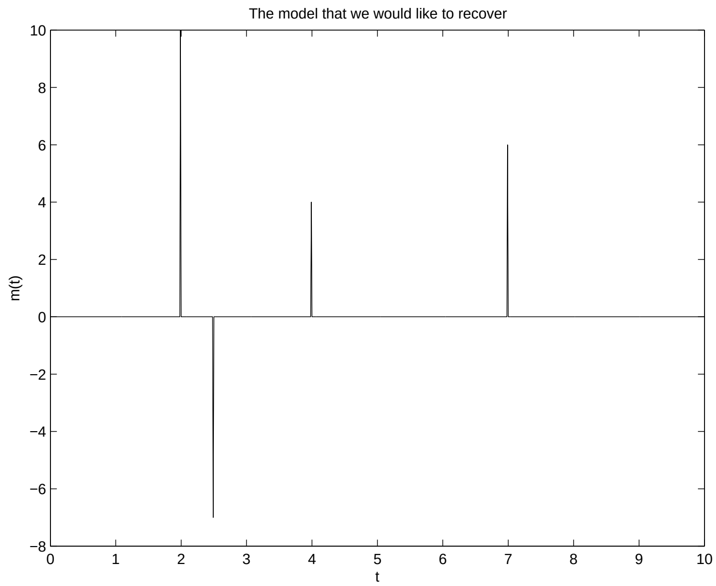

Figure 12: The target model $m(t)$.

For an example, we'll use the same impulse response from our previous example.

$$g(t) = e^{-5t}\sin(10t). \tag{27}$$

This time our target model $m(t)$ will be

$$m(t) = 10\delta(t-2) - 7\delta(t-2.5) + 4\delta(t-4) + 6\delta(t-7). \tag{28}$$

Again we'll add random noise to the convolved signal and then attempt to recover $m(t)$.

Figure 12 shows the target model. Figure 13 shows the data with noise added. It's quite hard to pick out the impulses in this plot. Figures 14 shows the best result that could be obtained with Tikhonov regularization. The impulses are artificially broadened, and the noise is not completely removed from the signal. Figure 15 shows using (26) produces an amazingly good recovery of $m(t)$. Notice that the spikes are correctly placed in time. The amplitude of the spikes is reduced and the spikes are slightly broader than they should be, but the results are vastly better than the results obtained with Tikhonov regularization.

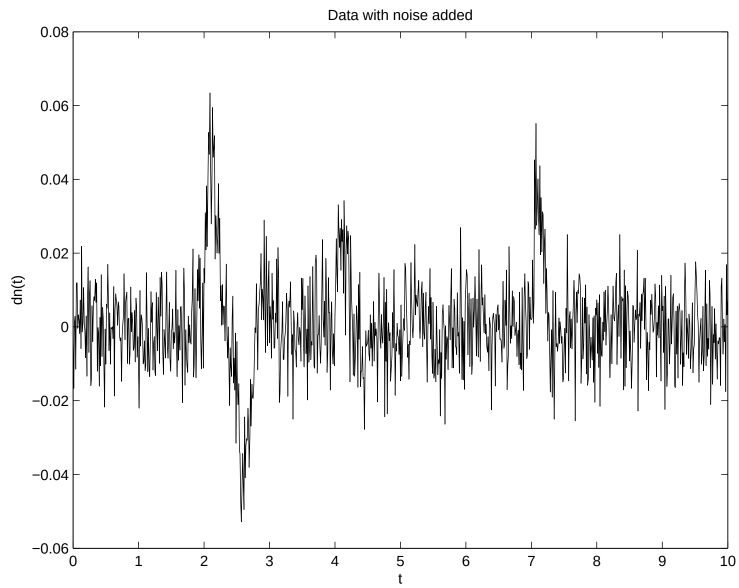

Figure 13: Data with noise added.

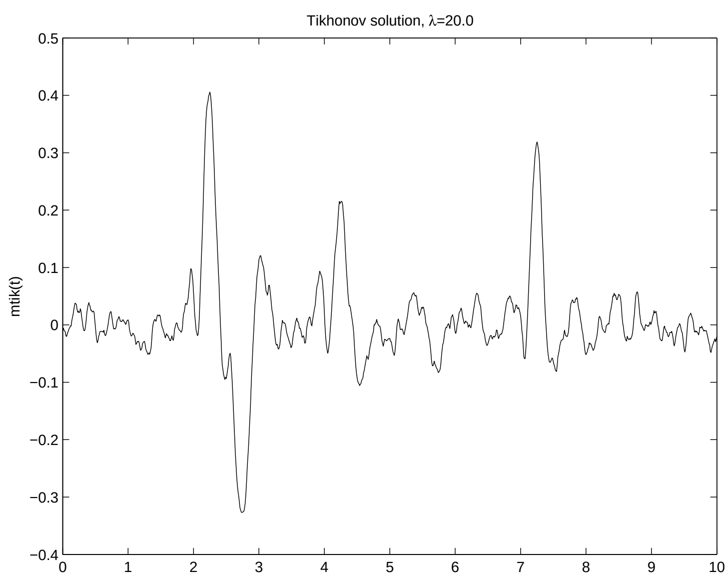

Figure 14: Tikhonov solution.

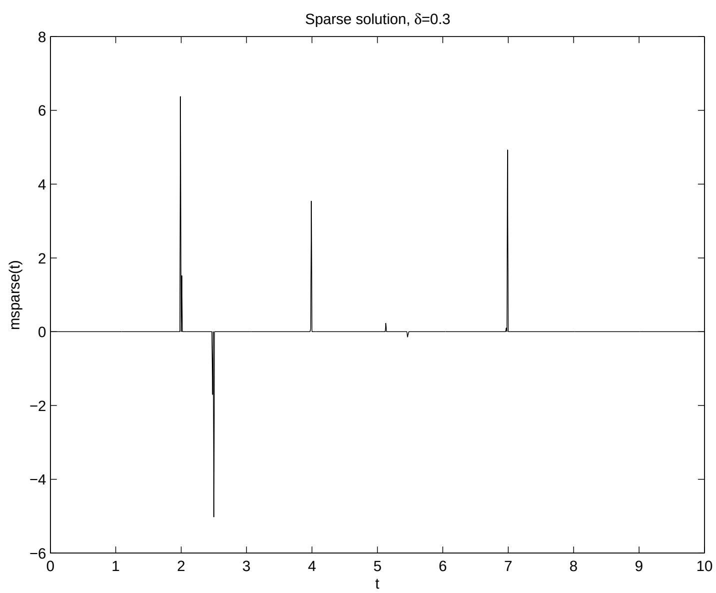

Figure 15: 1-norm regularized solution.