**Tune-Up Tuesday #4 Deconvolution for October 9, 2025**

Before we talk about deconvolution, let’s define convolution. Then, we’ll derive a deconvolution algorithm and apply it to two examples. You could have worked either example for the Tune-Up.

**Convolution**. For a finite impulse response filter with $M+1$ filter coefficients $b\_{0}, b\_{1}, …, b\_{M}$, the output signal $y\left[n\right]$ for an input signal $x\left[n\right]$ is computed according to

$$y\left[n\right]=b\_{0} x\left[n\right]+b\_{1} x\left[n-1\right]+\cdots +b\_{M} x\left[n-M\right]$$

If we input the discrete-time impulse signal $δ[n]$, which has value of 1 at $n=0$ and 0 otherwise, the output is called the *impulse response* (response is synonymous with output):

$$h\left[n\right]=b\_{0} δ\left[n\right]+b\_{1} δ\left[n-1\right]+\cdots +b\_{M} δ\left[n-M\right]$$

Hence,

$$h\left[k\right]=b\_{k} for k= 0, 1,…, M $$

Otherwise, $h\left[k\right]=0$. Because $h\left[k\right]=b\_{k} for k= 0, 1,…, M$,

$$y\left[n\right]=h\left[0\right]x\left[n\right]+h\left[1\right]x\left[n-1\right]+\cdots +h\left[M+1\right]x\left[n-\left(M+1\right)\right]=\sum\_{k=0}^{M}h\left[k\right] x[n-k]$$

This computation is known as convolution of $h\left[n\right]$ and $x\left[n\right]$:

$$y\left[n\right]=h\left[n\right]\*x[n]$$

Given input signal $x\left[n\right]$ and impulse response $h\left[n\right]$, we can compute output signal $y\left[n\right]$. If $x\left[n\right]$ is also finite in length, then the length of $y\left[n\right]$ will be the length of $h\left[n\right]$ plus the length of $x\left[n\right]$ minus 1.

**Deconvolution.** Whereas convolution computes the output signal $y\left[n\right]$ from input signal $x\left[n\right]$ and an impulse response $h\left[n\right]$ of a FIR filter, deconvolution seeks to find impulse response $h\left[n\right]$ given input signal $x\left[n\right]$ and output signal $y\left[n\right]$. We can choose the input signal $x\left[n\right]$, also known as a test signal, and observe the output signal.

**Practical scenario**. We would start the test signal and the observation at a particular point in time, which we’ll say is at $n=0$ without loss of generality. Further, we will assume that $x\left[n\right]=0 for n<0$; i.e., $x\left[n\right]$ is a causal signal.

**Deconvolution Algorithm.** We’ll work backwards in the time domain to compute the FIR filter coefficients. We derive a time-domain deconvolution algorithm by first evaluating the output at $n=0$:

$$y\left[0\right]=h[0] x\left[0\right]+h[1] x\left[-1\right]+h[2] x\left[-2\right]+\cdots +h[M] x\left[-M\right]$$

As mentioned above, we’ll assume $x\left[n\right]$ is a causal signal; i.e.,$x\left[n\right]=0 for n<0$*.* Since we know *x*[*n*] and *y*[*n*], we have one equation and one unknown at *n* = 0:

$$y\left[0\right]=h[0] x\left[0\right]$$

and we can compute$$h[0]=\frac{y[0]}{x[0]}$$

For this calculation to be valid, the first value of the test signal, *x*[0], cannot be zero.

Second output: $y\left[1\right]=h[0] x\left[1\right]+h[1] x\left[0\right]$, and therefore, $h[1]=\frac{y\left[1\right]- h\left[0\right] x[1]}{x[0]}$.

Third output: $y\left[2\right]=h[0] x\left[2\right]+h[1] x\left[1\right]+h[2] x\left[0\right]$ and $h[2]=\frac{y\left[2\right]- h\left[0\right] x\left[2\right]-h\left[1\right] x[1]}{x[0]}$

In general, for the *N*th output, $h[N]=\frac{y\left[N\right] - \sum\_{i=0}^{N-1}h\left[i\right] x[N-i]}{x[0]}$.

The MATLAB script [utdeconvolve.m](http://users.ece.utexas.edu/~bevans/courses/signals/tuneups/fall2024/utdeconvolve.m) implements this algorithm.

**Example**. *Problem 4.3(b).* In this problem, we’re given

* causal input signal *x*[*n*] with non-zero values [ 1 2 3 4 5 ]
* causal output signal *y*[*n*] with non-zero values [ 1 1 1 1 1 -5 ]

We can compute the FIR filter coefficients using the above deconvolution algorithm:

$$h\left[0\right]=\frac{y\left[0\right]}{x\left[0\right]}=\frac{1}{1}=1$$

$$h\left[1\right]=\frac{y\left[1\right]- h\left[0\right] x\left[1\right]}{x\left[0\right]}=\frac{1-1∙2}{1}=-1$$

$$h\left[2\right]=\frac{y\left[2\right]- h\left[0\right] x\left[2\right]-h\left[1\right] x\left[1\right]}{x\left[0\right]}=\frac{1-1∙3-(-1)∙2}{1}=0$$

The values of $h[n]$ for $n>2 $are zero. The MATLAB script [utdeconvolve.m](http://users.ece.utexas.edu/~bevans/courses/signals/tuneups/fall2024/utdeconvolve.m) will give the same answer for $h[n]$. We can validate the answer by convolving $h[n]$ and $x[n]$. We can use the Matlab command conv to do this:

y = conv( [1 -1], [ 1 2 3 4 5 ] )

y =

 1 1 1 1 1 -5

Alternately, we could use the filter command. Keeping in mind that the filter command produces as many output samples as there are input samples,

y = filter( [1, -1], 1, [ 1 2 3 4 5 0 ] )

y =

 1 1 1 1 1 -5

When convolving two finite-length signals $x\left[n\right]$ and $h\left[n\right]$, the result $y\left[n\right] $has finite length. The length of $y\left[n\right]$ is the length of $x\left[n\right]$ plus the number of filter coefficients minus 1. Since the length of $y\left[n\right]$ is 6 and the length of $x\left[n\right] $is 5, there are 2 filter coefficients.