

Explaining Convolution Using MATLAB

Thomas Murphy¹

Abstract

Students often have a difficult time understanding what convolution is. Students can often evaluate the convolution integral (continuous time case), convolution sum (discrete-time case), or perform graphical convolution but may not have a good grasp of what is happening. In other words, students can solve the formula but often do not understand the result or why they get that result. Most engineering texts explain convolution by giving the convolution integral (and/or convolution sum) and doing some mathematical and graphical examples. They often do not attempt to explain how convolution corresponds with what is happening between the system and the input to give the output response. In this paper, a more intuitive explanation of convolution is given and MATLAB and SIMULINK simulations of physical systems are used to give a more intuitive approach to understanding convolution from a systems perspective.

Introduction

Students are often introduced to convolution before they see the use for it. In most Electrical Engineering curriculums, convolution is introduced in sophomore or junior level signals and systems courses. Convolution is often performed numerically and students have a tendency to blindly accept the results their calculator or computer provides. Thus it is important for students to understand the use, along with the theory of convolution, so they can better evaluate the results they get from convolution. If convolution is explained from a systems perspective with good examples, they will see the use, and hopefully understand the theory and be able to determine if their results make sense.

I have found several interactive web lectures/tutorials on convolution [1][2][3]. These web pages give a good definition of convolution and a good explanation of how to compute the convolution of two signals or functions. This paper proposes explaining convolution from a systems perspective using simulations of familiar systems. It is hoped that this will give students insight into what is happening with convolution.

The system simulation examples are performed using MATLAB and SIMULINK. MATLAB and SIMULINK are used extensively in Electrical Engineering programs for signals and systems courses, control systems courses, and signal processing courses to name a few. MATLAB and SIMULINK work well for simulating most electrical, mechanical, and chemical systems. Basically any system that can be modeled in transfer function form, by differential equations, or state variable equations can be simulated using MATLAB and SIMULINK.

The paper is organized as follows: the definition of convolution along with some system theory is provided, methods of computing convolution are discussed, a method of explaining convolution using familiar systems is proposed, and suggestions are provided for how to incorporate this in the classroom.

¹ Assistant Professor of Computer Science, Georgia Southern University, P.O. Box 8093, Statesboro, GA, 30460 - 8093.

Convolution Definition and Systems Background

We begin with the definition of convolution and a brief discussion of some system theory.

Definition of Convolution

Convolution is usually defined via the convolution integral for continuous time functions or the convolution sum for discrete-time functions. The convolution of two continuous time functions $h(t)$ and $x(t)$ written $h(t) * x(t)$ and is given by the convolution integral

$$h(t) * x(t) = \int_{-\infty}^{\infty} h(t)x(t-t)dt = \int_{-\infty}^{\infty} h(t-t)x(t)dt$$

The convolution integral exists if $h(t)$ and $x(t)$ are absolutely integrable.

The convolution of two discrete-time functions $h[n]$ and $x[n]$ written $h[n] * x[n]$ can be found using the convolution sum

$$h[n] * x[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} h[n-k]x[k]$$

Note that convolution is both commutative and distributive.

Systems Background

Since in this paper we will look at convolution from a systems perspective, some systems background will be helpful. A system is an interconnection of components with terminals or ports where inputs can be applied and outputs extracted. Systems can be represented through relationships between the system variables, usually between the input and output variables of the system. Figure 1 illustrates a black box approach of a system representation.

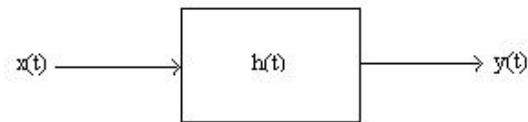


Figure 1

A system can be represented by its system function or impulse response $h(t)$. The output $y(t)$ is the response of the system to the input $x(t)$. Of course usually a mathematical model (approximation) of the system is used as opposed to an exact representation of the physical system.

System analysis and convolution are important for many reasons. For feasibility studies, mathematical analysis is often the only available method. Analytic studies are also usually cheaper than experimental studies. Certain cases are too dangerous to study experimentally (e.g., nuclear power plants at unsafe fission rates or aircraft control under severe turbulence). Convolution allows one to work in the time domain when the system function $h(t)$ and input function $x(t)$ are known or known only via experimental data.

Convolution will allow us to look at the impulse response of a system and predict how close the output will be to the input. Convolution is also important for many applications. Convolution is commonly used to implement linear operations on signals and images such as frequency filtering of signals, smoothing of images, and enhancement of images [4][5].

In this paper, some knowledge of signal and system theory is assumed. Familiarity with the notions of continuous and discrete-time signals and continuous and discrete-time systems and their properties is helpful. Information on these topics can be found in most signals and systems texts such as Kamen and Heck [6].

There are two basic types of system models: input-output models and state or internal models. An input-output model gives the relationship between the inputs and the outputs of a system. A state model gives the relationship between the inputs, internal states, and the outputs of a system. In this paper we will look at the convolution model of a system. The convolution model is a time-domain input-output model of a system. Other system models include: input-output differential equations (a time domain model) and transfer function representations (a frequency domain model). For the discrete time case, difference equations are used instead of differential equations. Transfer function representations are often given in terms of Fourier transforms, Laplace transforms, or Z transforms.

In the continuous time case, the convolution integral gives the relationship between the input $x(t)$ of a linear, time-invariant (LTI) system with impulse response $h(t)$ and the output response $y(t)$. The output response is then given by

$$y(t) = h(t) * x(t) = \int_0^t h(t-\tau)x(\tau)d\tau = \int_0^t h(\tau)x(t-\tau)d\tau ; t \geq 0$$

This assumes that the input is zero for $t < 0$ and that the system has no initial energy (zero initial conditions). The lower limit of zero is due to $x(t) = 0$ for $t < 0$, and the upper limit of t is due to the system being causal ($h(t) = 0$ for $t < 0$, no weight is given to future inputs).

In the discrete-time case, an LTI system represented by the impulse response $h[n]$ with input $x[n]$ has an output response $y[n]$ given by the convolution sum

$$y[n] = h[n] * x[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} h[n-k]x[k]$$

Methods of Evaluating the Convolution Integral and Sum

Evaluating the convolution integral is possible when mathematical expressions for $h(t)$ and $x(t)$ are available. However, in practice the convolution integral is often fairly complicated even for relatively simple $h(t)$ and $x(t)$. The convolution integral is often evaluated graphically or using transforms. To graphically compute the output $y(t)$, plot the impulse response of the system $h(t)$ and the flipped and shifted input $x(t-t)$. To find $x(t-t)$, first flip the input about the y-axis to form $x(-t)$ and then shift $x(-t)$ by t to the left. Multiply the two signals $h(t)$ and $x(t-t)$ and find the area of the resulting function. This gives the output $y(t)$ for a particular value of t . This can be repeated for all values of t of interest. [3]

The convolution sum (discrete-time case) is fairly easy to evaluate in practice. Consider the direct form of the convolution sum

$$y[n] = \sum_{k=-\infty}^{\infty} h[k] x[n-k]$$

Assuming the impulse response $h[n]$ and the input sequence $x[n]$ are available, flip and shift the input sequence to form $x[n-k]$ and make a substitution of variables to form $h[k]$. Then for each value n of interest, evaluate the convolution sum over the index k . For systems with finite length impulse responses and input sequences of finite length, both the lower and upper limits of the summation will be finite values that can be determined. Students often forget that the convolution sum must be evaluated in its entirety to obtain one value of $y[n]$ and thus to obtain the output sequence, the convolution sum must be evaluated for all n of interest.

The flip and slide method of discrete-time convolution is similar to graphical convolution. To convolve a system's impulse response $h[n]$ with an input sequence $x[n]$, the input sequence will be flipped, shifted to the left by n units, and slid through the stationary impulse response. In other words, first obtain $x[-k]$ by flipping the input sequence about the y -axis. Form $x[n-k]$ by shifting the $x[-k]$ sequence by n samples to the left. Finally, hold $h[k]$ stationary and slide $x[n-k]$ through. Start with an n large enough so that $h[k]$ and $x[n-k]$ do not overlap, calculate $y[n]$ by taking the dot product of $h[k]$ and $x[n-k]$ for that particular n . Continue by incrementing n by one, calculating $y[n]$ and repeating until all nonzero $y[n]$ have been calculated. There will be a finite number of nonzero terms for sequences that have finite length.

Convolution is often performed using the property that convolution in the time-domain corresponds with multiplication in the frequency domain. Thus using the Fourier (or Laplace) Transform in the continuous time case, the convolution of $y(t) = h(t) * x(t)$ can be found by

$$Y(\omega) = H(\omega)X(\omega), \text{ where } \mathcal{F}\{h(t)\} = H(\omega), \mathcal{F}\{x(t)\} = X(\omega), \mathcal{F}\{y(t)\} = Y(\omega) \text{ and } y(t) = \mathcal{F}^{-1}\{Y(\omega)\}$$

Using the Z transform in the discrete-time case, the convolution $y[n] = h[n] * x[n]$ can be found by

$$Y(z) = H(z)X(z), \text{ where } \mathcal{Z}\{h(t)\} = H(z), \mathcal{Z}\{x(t)\} = X(z), \mathcal{Z}\{y(t)\} = Y(z) \text{ and } y(t) = \mathcal{Z}^{-1}\{Y(z)\}$$

Explaining Convolution Using a RC Circuit Example

Two of the more intuitive explanations of convolution I have found are in the introductory circuits textbook by Nilsson [7] and from an email conversation between Dr. Richard Hathaway of the Georgia Southern Department of Mathematics and Computer Science and other members of the same department [8] (the most intuitive response of the conversation came from Dr. Donald W. Fausett, Professor and Chair of the Department of Mathematics and Computer Science).

The convolution integral can be looked at as $h(t)$ being a weighting function and the convolution integral as being a weighted average of the input $x(t)$ over the interval of integration. We can think of the convolution at each point t as a compromise of the different memories of the input x in the past (or at different locations,

depending on whether the variable is time or space). The values of $h(\tau)$ tell how well we remember the input $x(t-\tau)$ [8].

Looking at graphical convolution, when the input is folded, the vertical axis represents present time. Past values of the input are to the right of the vertical axis and future values of the input are to the left of the vertical axis. The impulse response of the system weights the inputs according to present and past values, giving more weight to some values than others. [7, p. 700]

So by multiplying the flipped shifted input by the impulse response, the impulse response serves as a weighting function and gives insight into system's memory. The more time, over which the impulse response is nonzero, the more memory the system has. The impulse response provides a certain weight depending upon when the value of the input occurs/occurred past or present. If the system has no memory, past values of the input are given no weight. The more memory a system has, the more distortion or difference between the system response and the input signal.

Although convolution is often computationally easier to do in the frequency domain, I believe that it is easier to see what is happening in the time domain. Let us look at using these interpretations of convolution to explain what is happening in a RC circuit (Figure 2). In other words, look at the system response of a RC circuit and relate convolution to what students would have learned in a circuits course.

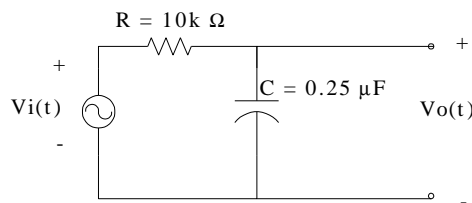


Figure 2, RC circuit of convolution example

The transfer function of the system (you may need to derive this using circuit analysis depending upon the level of the students) is

$$H(s) = \frac{V_o(s)}{V_i(s)} = \frac{1/RC}{s + 1/RC}$$

The time constant of the circuit is $RC = 2.5 \times 10^{-3}$ sec and the impulse response of the system is

$h(t) = \frac{1}{RC} e^{-\frac{1}{RC}t} = 400e^{-400t}$. The input to the system is a pulse waveform with an amplitude of 4 Volts, a period of 0.02 seconds, and a duty cycle of 50%. A plot of the input signal is shown in Figure 3. When the input signal $x(t)$ is folded or flipped, the vertical axis represents the present time and future input values are the values to the left of the vertical axis. The flipped signal $x(-t)$ is shown in Figure 4.

The impulse response $h(t)$ weights the input according to present and past values, giving more weight to some values than others. When convolving $h(t)$ and $x(t)$, multiplying $h(t)$ by $x(t-t)$ applies certain weights depending on when the input value occurs (past versus present). Looking at the impulse response $h(t)$ of the RC circuit (Figure 5) the present input value is weighted much more strongly than past input

values as seen by the high amplitude of $h(t)$ near $t=0$ (present input) and smaller amplitudes of $h(t)$ for larger t .

If the system had no memory, then past input values would be given no weight and the output response would be a replica of the input.

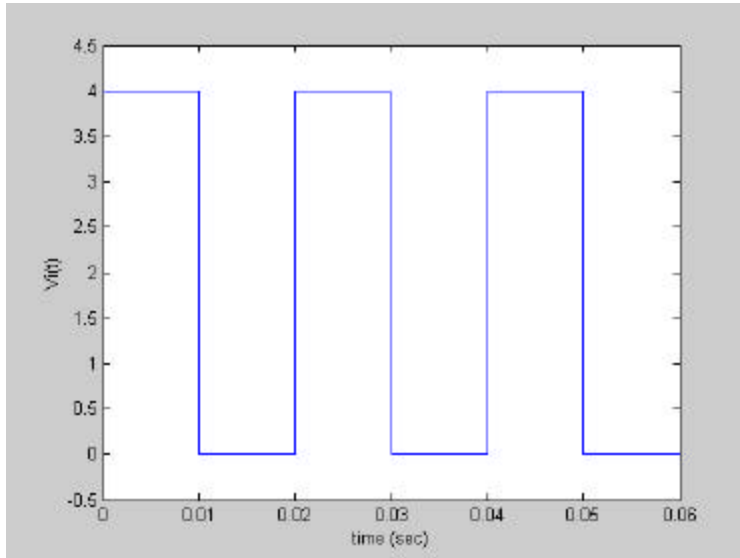


Figure 3, input $x(t)$

For the RC circuit, the impulse response decays rapidly to zero. Recall the time constant of the system was $RC = 2.5 \times 10^{-3}$ sec. Thus compared to mechanical and chemical systems, which usually have much longer time constants, this system does not have much memory. This will show up in the output response by the output rapidly decaying to zero when the input voltage is taken away. A system with more memory would have a longer nonzero output after the input was taken away.

After investigating the input and impulse response, discuss the system from a perspective they would have encountered earlier in their coursework. For the RC circuit example, this could mean explaining the system response from the capacitor charging and discharging.

Initially assume the voltage across the capacitor is zero. When the pulse is applied to the RC circuit, the capacitor will start charging up. This corresponds with the flipped input starting to overlap the impulse response and as more of the input pulse overlaps the impulse response, the output voltage will go up. When the nonzero portion of the input pulse is fully overlapping the impulse response, the output voltage should be at its maximum. As the input pulse is taken away (less of the nonzero part of the pulse overlaps with the impulse response), the capacitor should start discharging and the output voltage will go down. Once the input pulse is no longer overlapping the impulse response, the output voltage will be zero.

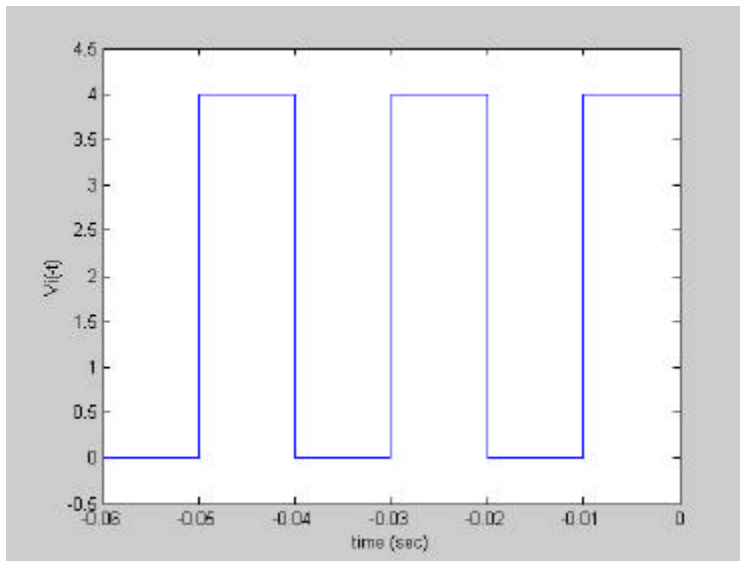


Figure 4, flipped input $x(-t)$

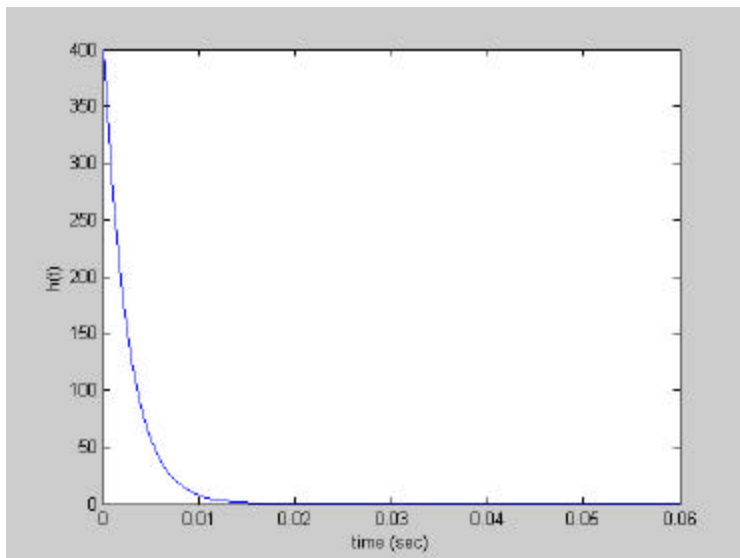


Figure 5, impulse response $h(t)$

The explanation/comparison of the system response from convolution and from a method they learned previously hopefully will make sense to the students. They then should have a good idea of what the output response should look like and why. One could further flesh out the expected output response by discussing the time constant of the system and how this will show up in the response.

Looking at a plot of the output response (figure 5), one can see that it agrees with what we expect from our knowledge of RC circuits and the convolution (this can be explained in more detail). Comments about system stability (relating to finite area under the curves), causality (when output starts), and other properties could be made if desired.

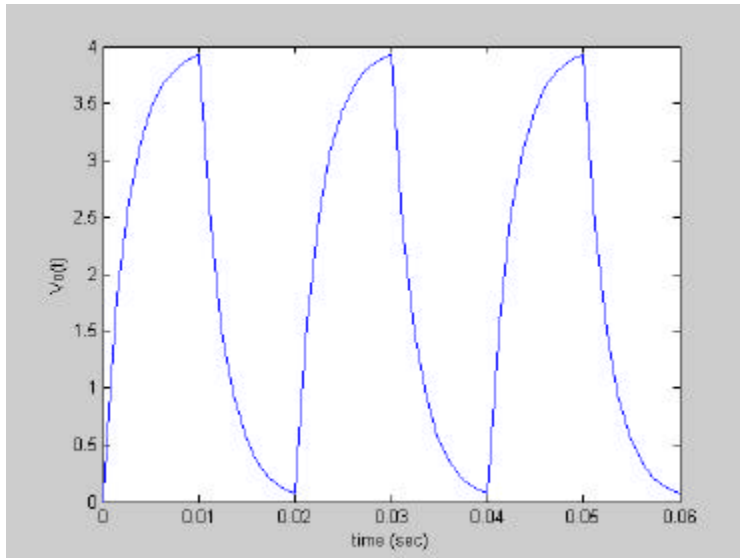


Figure 6, output response $y(t)$

The plots, Figures 3 – 6, were generated using MATLAB and SIMULINK. A SIMULINK block diagram (Figure 7) simulation was used to generate the data values. The data values were then plotted using MATLAB. MATLAB and SIMULINK make it very easy to change the input to the system and quickly see what effect this has on the output. This allows the exploration of convolution and examples illustrating it with minimal effort. It also allows instructors to quickly illustrate “what ifs” by changing system parameters and the input signal on the fly.

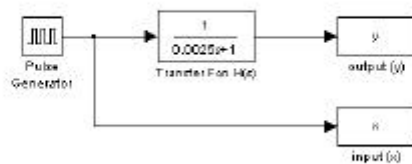


Figure 7, SIMULINK block diagram

Suggestions

When explaining convolution to students, give the definition of convolution, and then quickly move to intuitive examples. Discuss the input, impulse response, and the system's characteristics. Explain how these will contribute to the output response. Ask students to roughly sketch the output response and explain what is going on. Finally show and comment on the output response and answer any questions.

The examples used in class will be dependant on the course that convolution is introduced or discussed in. In Electrical Engineering curriculums, convolution is often explained in sophomore level signals and systems classes. Students will usually have taken a circuits course beforehand. Other possible examples that could be used in Electrical Engineering courses are frequency select filters (make sure to relate the example to something they are familiar with like an audio system), mechanical engineering courses could use automobile, ship, or airplane systems, and chemical engineering courses could use temperature control systems, fluid level systems, or models of chemical reactions.

In my experience, students have a difficult time understanding convolution. This paper proposes that convolution is best explained from a systems perspective using examples that students should be familiar with. In this paper, convolution was explained using a RC circuit example. This also allowed the exploration of system topics such as memory, causality, and stability. MATLAB and SIMULINK make it very easy to simulate systems and quickly see the answer to "what if" questions. There is always a danger that when explaining concepts using technology that students will pay more attention to the technology, in this case MATLAB, than to the subject matter.

References

- [1] <http://www.engin.umich.edu/group/ctm/index.html> (10/9/00, 12:52pm)
- [2] <http://www.engin.umich.edu/group/ctm/extras/polyadd.html> (10/9/00, 12:52pm)
- [3] <http://www.jhu.edu/~signals/> (10/9/00, 12:54pm)
- [4] Mitra, Sanjit K., 1998, *Digital Signal Processing A Computer-Based Approach*, McGraw-Hill, New York.
- [5] Castleman, Kenneth R., 1996, *Digital Image Processing*, Prentice-Hall, Upper Saddle River, New Jersey.
- [6] Kamen, Edward W. and Heck, Bonnie S., 2000, *Fundamentals of Signals and Systems Using the Web and MATLAB*, second edition, Prentice-Hall, Upper Saddle River, New Jersey.
- [7] Nilsson, James W., 1993, *Electric Circuits*, Fourth Edition, Addison Wesley Publishing Company, New York, pp. 694 – 701.
- [8] Email from Dr. Richard Hathaway, Thursday September 14, 2000, 08:41:57.

Thomas Murphy

Dr. Thomas Murphy is an Assistant Professor of Computer Science at Georgia Southern University. He received his BS in electrical engineering from The University of Notre Dame and his ME and Ph.D. in electrical engineering from The University of Florida. His interests lie in the areas of digital design, filtering, and signal processing. He is a member of IEEE, and currently serves as the Chair of the Electrical Engineering Division of the Southeastern Section of the ASEE.