

Graphical Convolution In Java

by

Erik S. Wheeler

EE 4012 -- Senior Design Project
Department of Electrical and Computer Engineering
Mississippi State University
wheeler@ISIP.MsState.Edu

ABSTRACT

Convolution is a concept that escapes many undergraduate engineering students. For years we have been forced to try and visualize the process with only limited success, but, with the advent of the World Wide Web and programming languages like Java, a tool for performing graphical convolution for anyone with an internet connection is now possible. Imagine wondering what the convolution of Pi and Triangle function is and going to your computer to see it performed right before your eyes. This tool will not solve all the problems of learning or teaching convolution but it should help some.



CONVOLUTION

Definition of convolution:

A linear system is often described by the output obtained when an impulse is placed at the input. This is called the impulse response of that system, denoted $h(t)$. Using this impulse response, the expected output of the system to any input may be determined by convolving that input with the system's impulse response.

Convolution integral:

$$y(t) = \int_{-\infty}^{\infty} x(\lambda)h(t - \lambda)d\lambda$$

Graphical convolution approach:

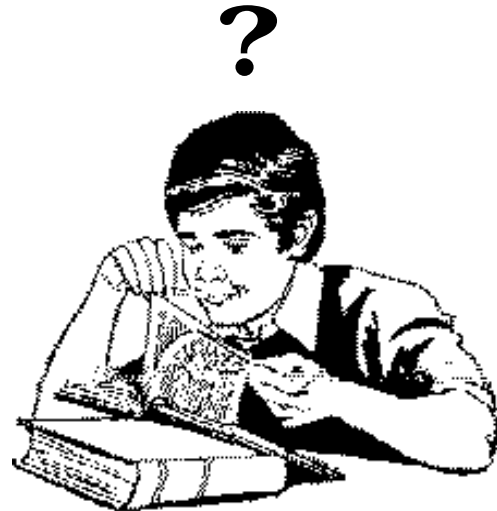
1. Flip the impulse response.
2. Slide it to the left.
3. Move to the right and take the area of the product of the overlapping curves.



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NEED FOR VISUAL TOOL

This sliding and summing of curves is often hard for students to visualize, so a tool that can perform this animation would probably help in understanding the process.

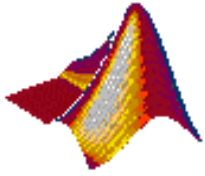


This tool would have to allow the user to draw the input and impulse response curves, then show the result of the convolution as the impulse response is moved across the input, or visa versa.



HOW TO IMPLEMENT THE CONVOLUTION TOOL

Possible Approaches:



Matlab

- ✓ Very convenient
- ✗ Limited accessibility



C or other standard language

- ✓ I already know C
- ✗ Multiple copies required
- ✗ Not platform independent



Java

- ✓ Platform independent
- ✓ Accessible to anyone with a connection to the internet
- ✓ Only requires one copy of executable
- ✗ I didn't know Java



OVERVIEW OF JAVA

According to "The Java Language: A White Paper", Java is
"A simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, and dynamic language."

Which means:

Simple: (this point can be argued)

- Based on C++
- Small Interpreter and class support ~ 40 Kbytes

Distributed:

- Library of Routines allows access across the net via URLs.

Architecture Neutral (this is the biggie):

- Compiler generates bytecode instructions, independent of computer architecture.

Portable:

- No implementation dependant aspects of the specification.

High Performance

- Comparable to native C or C++.

Multithreaded:

- Allows multiple processes to run at once.

Dynamic (another important one):

- Any changes in the code do not require upgrades for the user.



JAVA PROGRAMMING ENVIRONMENT

Java Development Kit (JDK v1.0):

- Java language compiler
- class libraries
- debugger
- other development tools

Availability:

- Windows 95
- Solaris
- Mac
- Linux
- More everyday

Additional Sun resources:

- Java language tutorial
- Java API documentation

Other resources:

- Teach Yourself Java in 21 Days* by
Lemay & Perkins



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

DESIGN PROCESS

- Discrete convolution considerations
 - 1) Scaling
 - 2) Time limits
- Convolution animation
- Drawing tool
- Labeling of drawing tool
- Link drawing tool to convolution
- Add Graphical User Interface (GUI's)
 - 1) Predefined curves
 - 2) Clear buttons
 - 3) Pause button



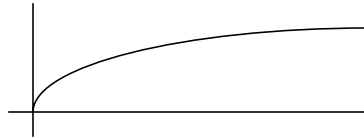
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

PROGRAM FEATURES

Drawing Tool

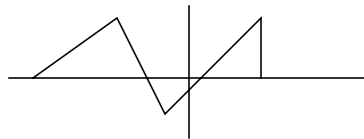
Predefined Curves

- 1) $\text{Cos}(t)$
- 2) $\text{Exp}(-t)$
- 3) $1 - \text{Exp}(-t)$
- 4) Approximation of Impulse
- 5) $\text{Sinc}(t)$



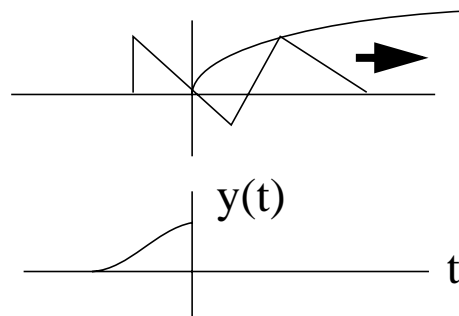
Freehand Drawing

- 1) Connects to last point on curve
- 2) Does not allow overlapping lines or infinite slope
- 3) Clear drawing and start over at any time
- 4) Allows drawing outside of borders



Convolution Area

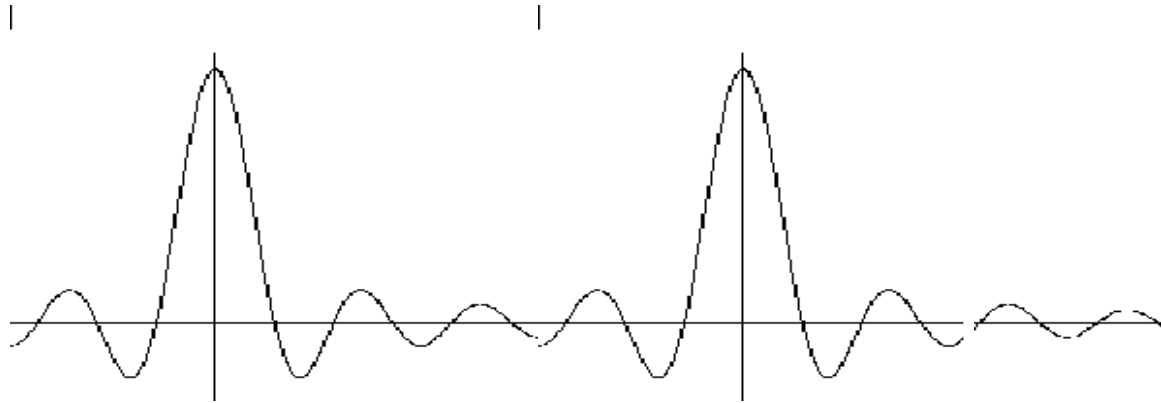
- Separate Applet linked to Drawing Tool
- Displays animation loop of graphical convolution process



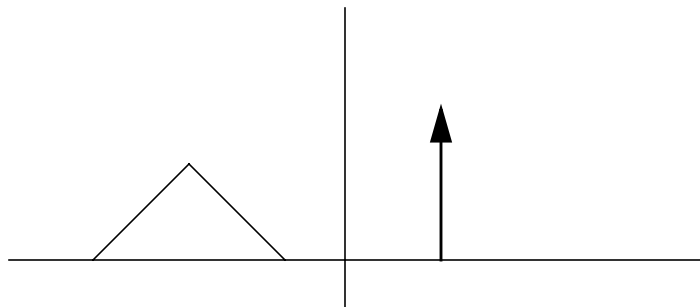
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

DEMO

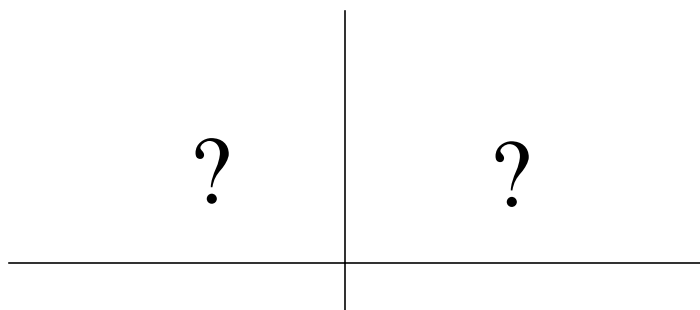
$$\square \quad \text{Sinc}(t) \otimes \text{Sinc}(t) = \frac{1}{2} \text{Sinc}(t)$$



$$\square \quad \Lambda(t-1) \otimes \delta(t) = \Lambda(t-1)$$



\square Special Request?



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CODE

DrawCurves.java

```
void fillArray() {
    double slope = (double)(y1 - ylast) / (double)(xlast - x1);
    for (int i=x1; i<=xlast; i++) {
        if ((i>=0)&&(i<(int)(maxwidth/2))) {
            curveArray[i]=(int)(slope*(i-x1)+(zeroY-y1)); }
        }
    }
}
```

Convolution.java

```
void createY() {
    int i;
    Convolve();

    for (i=0;i<maxwidth;i++) {
        hposy[i] = i+pstart;
        vposy[i] = 20 + 2*maxheight - yarray[i];
    }
}

public void Convolve() {
    int i,j;
    for (i = 0;i < maxwidth;i++) fyarray[i]=0.0;
    for (i = 0;i < (2*quarterwidth);i++) {
        for (j = 0;j < (2*quarterwidth);j++) {
            fyarray[i+j]=fyarray[i+j]+(fxarray[i]*fharray[j]);
        }
    }
    for (i = 0;i<maxwidth;i++) {
        fyarray[i] = fyarray[i] / timedivconst;
        yarray[i] = (int)(fyarray[i] * divconst);
    }
}
```



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

POSSIBLE IMPROVEMENTS:

- Allowing Infinite Slopes
- Better Approximation of the Impulse function

WHAT'S NEXT:

Signals & Systems Toolkit

- Correlation / Autocorrelation
- Graphical Fourier Transform
- Bode Plots

OTHER JAVA APPLETS:

(Available at Gamelan, <http://www.gamelan.com/>):

- Phasor Demonstrations
- Virtual Wind Tunnel
- Many more



SUMMARY

- Not just a demonstration
- Not just for MSU students and teachers
- Java has a great potential for teaching
- Check it out with a Java compatible browser:



<http://isip.MsState.Edu/>

Click on FUN STUFF

- The code is available via anonymous ftp at:
ftp://ftp.isip.msstate.edu/pub/software/java_convolution/



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

REFERENCES

1. M. Campione and K. Walrath, "The Java Language Tutorial, Object-Oriented Programming for the Internet," at <http://java.sun.com/>, Sun Microsystems, updated Jan. 23, 1996.
2. Java API Documentation, at <http://java.sun.com/>, Sun Microsystems, 1996.
3. L. Lemay and C.L. Perkins, *Teach Yourself Java in 21 Days*, Sams.net, Indianapolis, IN 1996.

