

VIP ITS: Discrete Convolution GUI and State Saving

Sean Crowley, Neal Kurande, Camille Bossut

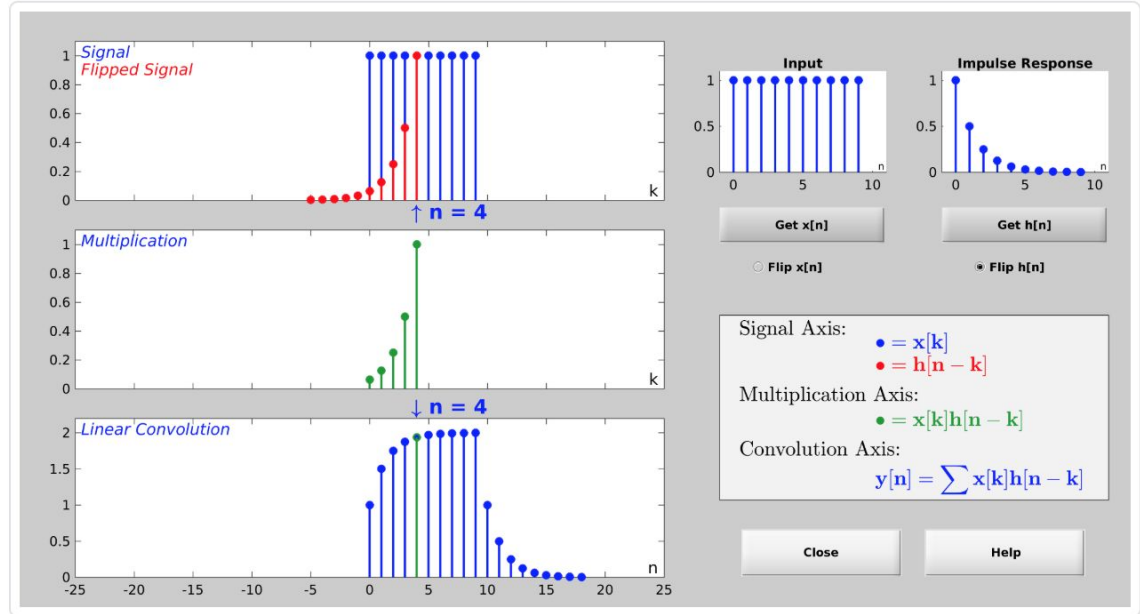


Project Plan

- Modify GUI to have all the properties of the Matlab version
- Implement a smooth control system across the GUI
- Save the state of the GUI to a json file

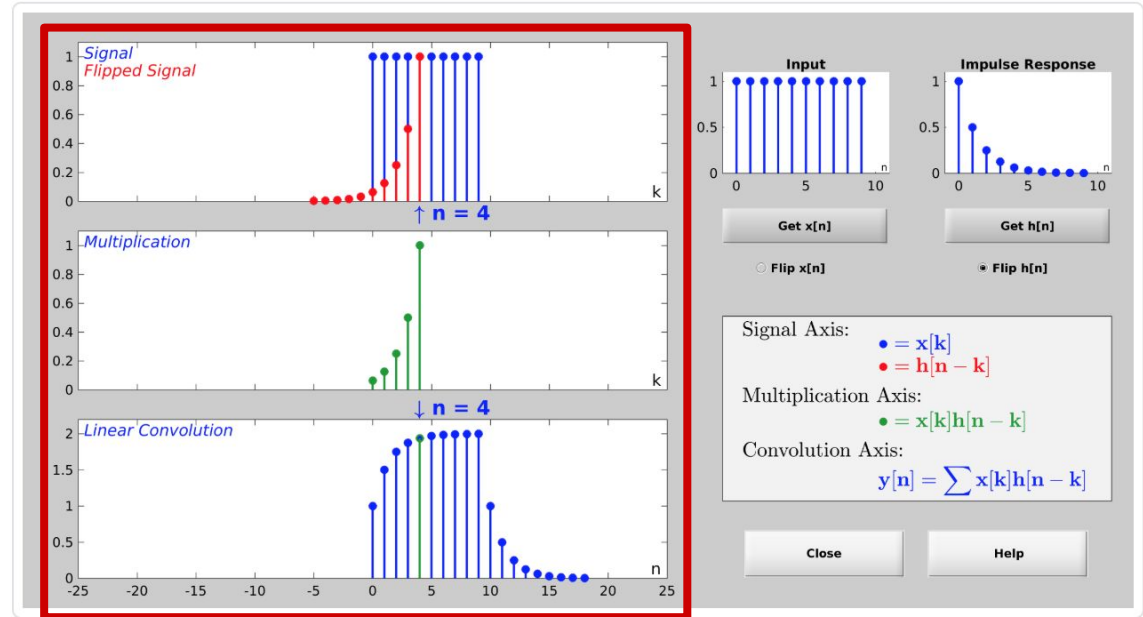
GUI Changes

Matlab GUI



Matlab GUI

GUI Changes

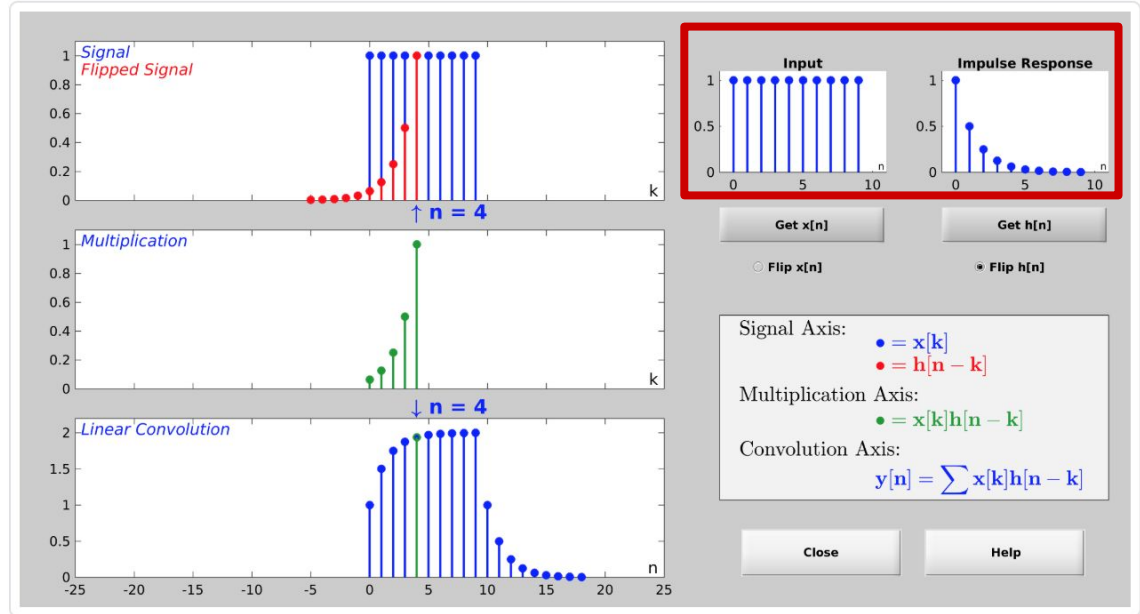


Three plots

(input signals, multiplication plot, and convolution result)

GUI Changes

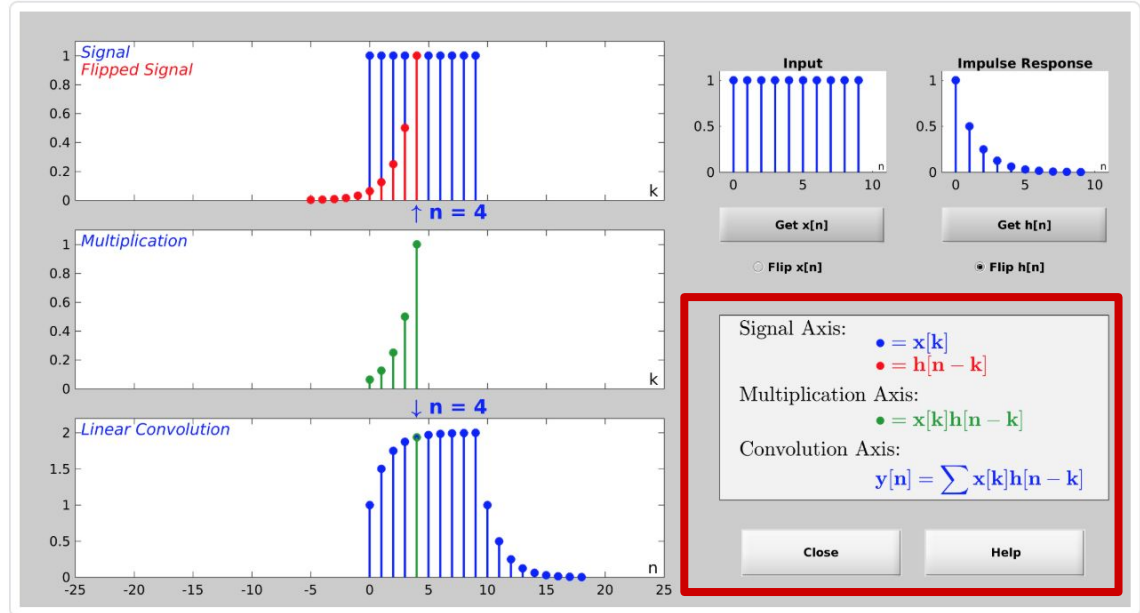
Matlab GUI



Static input plots

Matlab GUI

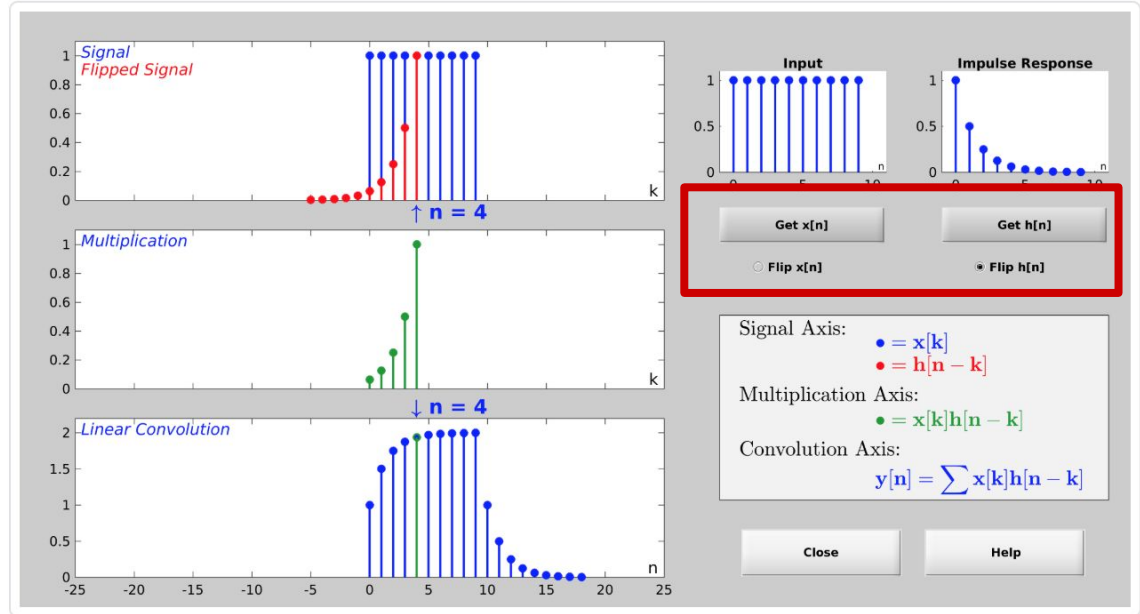
GUI Changes



Displayed equations

Matlab GUI

GUI Changes

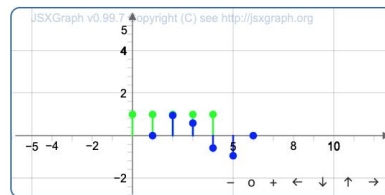


Choice of Signals

GUI Changes

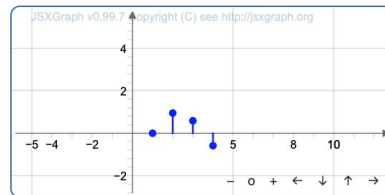
DisConvDemo

Input Graph

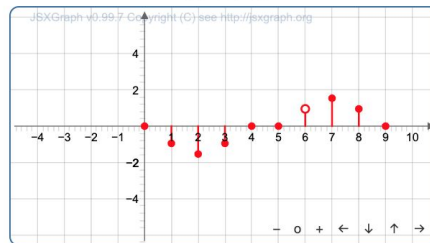


n value: 1

Multiplication Graph



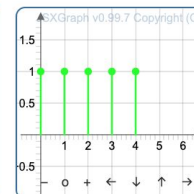
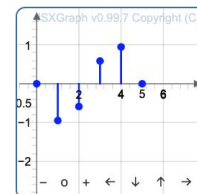
Result Graph



Save Progress

Restore Progress

Input Signals



h[k] Functions

x[k] Functions

Signal Axis

$x[k]$

$h[n - k]$

Multiplication Axis

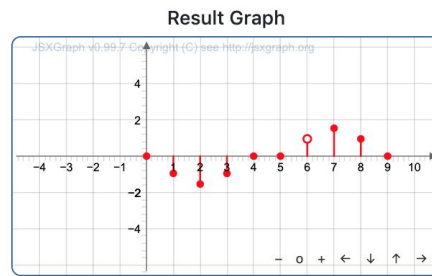
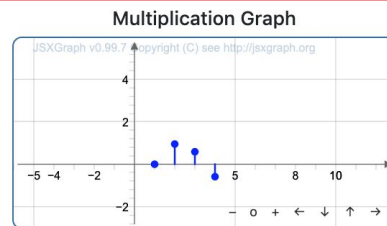
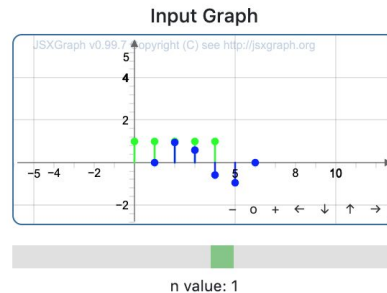
$x[n] * h[n - k]$

Convolution Axis

$\sum x[k]h[n - k]$

GUI Changes

DisConvDemo

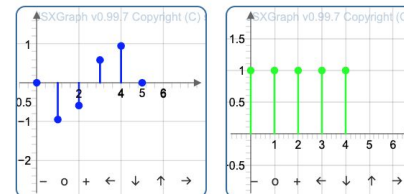


Three plots

Save Progress

Restore Progress

Input Signals



h[k] Functions

x[k] Functions

Signal Axis

$x[k]$

$h[n - k]$

Multiplication Axis

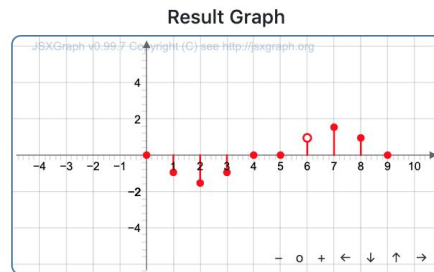
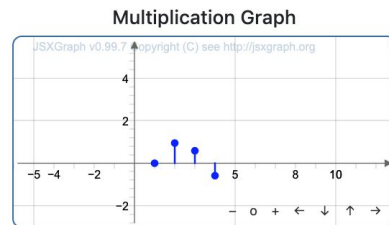
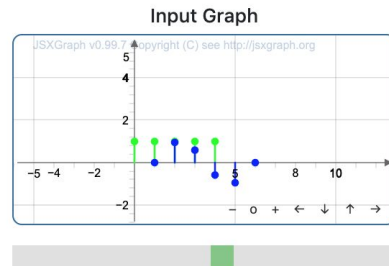
$x[n] * h[n - k]$

Convolution Axis

$\sum x[k]h[n - k]$

GUI Changes

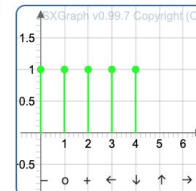
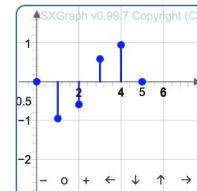
DisConvDemo



Save Progress

Restore Progress

Input Signals



h[k] Functions

x[k] Functions

Signal Axis

$x[k]$

$h[n - k]$

Multiplication Axis

$x[n] * h[n - k]$

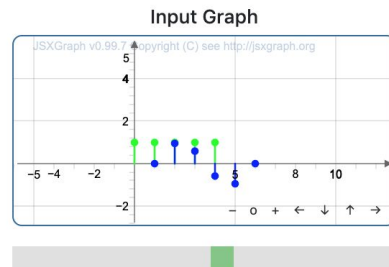
Convolution Axis

$\sum x[k]h[n - k]$

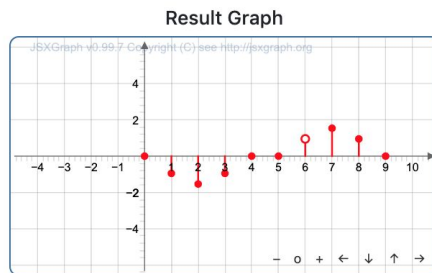
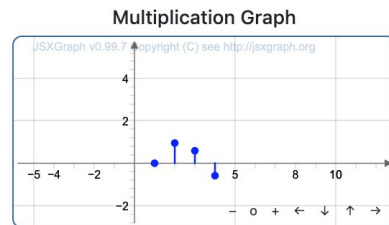
Static input plots

GUI Changes

DisConvDemo



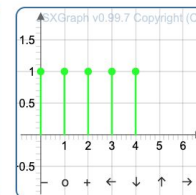
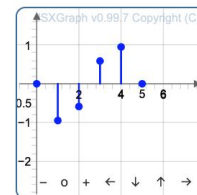
n value: 1



Save Progress

Restore Progress

Input Signals



h[k] Functions

x[k] Functions

Signal Axis

$x[k]$

$h[n - k]$

Multiplication Axis

$x[n] * h[n - k]$

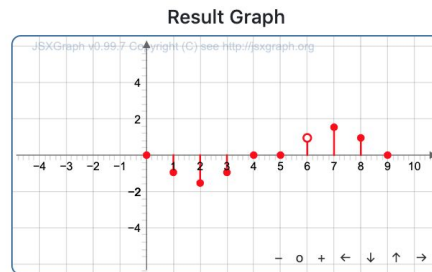
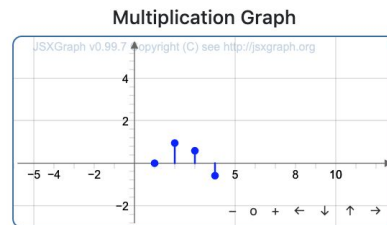
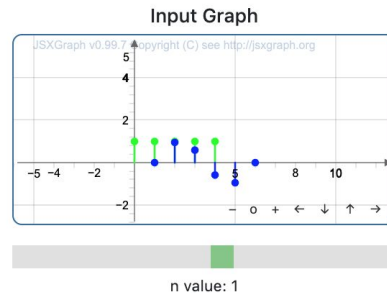
Convolution Axis

$\sum x[k]h[n - k]$

Displayed equations

GUI Changes

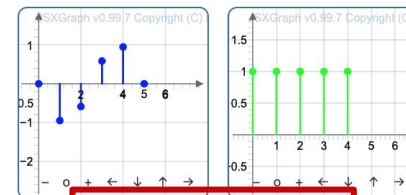
DisConvDemo



Save Progress

Restore Progress

Input Signals



h[k] Functions

x[k] Functions

Signal Axis

$x[k]$

$h[n - k]$

Multiplication Axis

$x[n] * h[n - k]$

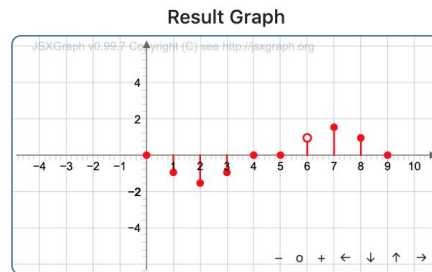
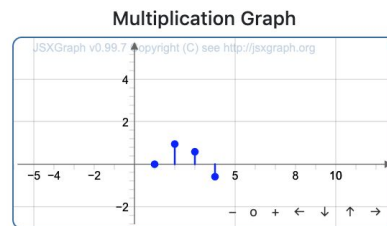
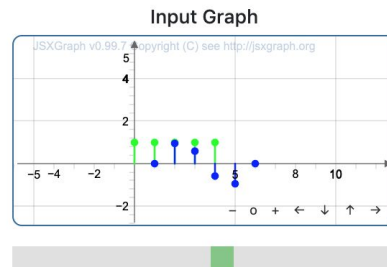
Convolution Axis

$\sum x[k]h[n - k]$

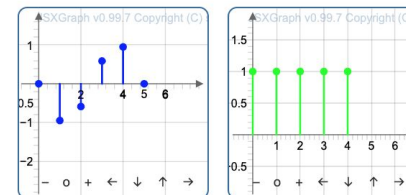
Choice of Signal

GUI Changes

DisConvDemo



Input Signals



Signal Axis

$x[k]$

$h[n - k]$

Multiplication Axis

$x[n] * h[n - k]$

Convolution Axis

$\sum x[k]h[n - k]$

Save State Option

GUI States

Goal: Create a way to **save/restore** student progress for ease of grading.

Solution

- Locally hosted PHP server to render lab website.
- On HTML button click send a JSON containing GUI state to the server.
- Server side PHP scripts:
 - JSON writing
 - JSON reading

Client Side State Process

- HTML Buttons:
 - save_state
 - restore_state
- JavaScript (In conjunction with JQuery)
 - Document Object Model - getElementById
 - Ajax (Asynchronous JavaScript and XML)
 - Send JSON data to server.
 - Load data from JSON and restore state without page refresh.

Server Side Save Process

- JSON processing with PHP scripts:
 - save.php:
 - HTTP POST request and JSON string sent from the client.
 - JSON information written to state.json.
 - restore.php
 - HTTP GET request is sent from client.
 - JSON string is returned to the client for state restoration.
- PHP Local Host Setup:
 - Terminal: "php -S localhost:4000/*"
 - Web Browser: "localhost:4000/DisConvDemo.html"

Future Improvements

- SQL Database Integration and User IDs for state differentiation.
- Letting the user control a wider variety of inputs to the GUI.
- Create an abstract state saving feature to save the state of any future project.
- Complete CSS styling so the lab website is user friendly.

Questions?