ABSTRACT

WEB-BASED SIMULATION OF FLEXIBLE LINK MANIPULATORS AND IIR FILTERS

Mohd Afwaan Ahmed Quadri, M.S.
Department of Electrical Engineering
Northern Illinois University, 2016
Reza Hashemian, Director

This thesis describes an outline for converting an existing stand-alone MATLAB and Simulink-based simulation environment to a web version. With this arrangement the simulation environment will be hosted on a web server and users will access from remote locations. In order to convert the algorithm in web-accessible format one need utilize the existing algorithm, MATLAB Compiler SDK, to translate the code into DLL file, incorporating the DLL with .Net framework using C# language. The main task in this project involves conversion of data between C# and MATLAB arrays, dissolving the GUI and writing MATLAB code in the right order in such a way that right outputs could be captured in .Net Assembly. The next step is to refine the data and plot the data in the web page.

NORTHERN ILLINOIS UNIVERSITY
DE KALB, ILLINOIS

MAY 2016

WEB-BASED SIMULATION OF FLEXIBLE MANIPULATORS AND IIR FILTERS

BY

MOHD AFWAAN AHMED QUADRI

A THESIS SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE

MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

Thesis Director:

Reza Hashemian

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

ACKNOWLEDGMENTS

I take this opportunity to acknowledge all of those who helped me in successfully finishing this thesis. First and foremost, I would like to thank my God, without whom it was impossible to complete this work. I would like to thank my parents, who always supported me. I would like to thank my brother, who always stood beside me. I would like to thank my thesis advisors, Dr. Abul K.M. Azad and Dr. Reza Hashemian, for their constant support, motivation and encouragement. Their useful discussions and valuable suggestions kept me focused on my research. I would like to thank Parthasarathy Krishnamurthy and Bharathwaj Murali, graduate students of the Computer Science Department, for their valuable suggestions and contribution in web design and plotting of data. I would like to thank Dr. Abul K.M. Azad for hiring me as an RA, TA and student employee for two semesters. I would like to thank Dr. Zinger for being part of my thesis committee.

I take this opportunity to convey my special thanks to Mei Chen for her help and great cooperation. She helped me in installing various softwares required by my project. I would like to thank my roommates and my friends who have been a great support and for being there for me at the time of need.

I would like to thank my sister for the love and care she have been showing over the course of my life.

DEDICATION

To Almighty ALLAH S.W.T

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

1. INTRODUCTION

In order to design a robotic arm or a manipulator system in manufacturing industries, one has to understand the mechanism and outcome results before building in order to have better performance economically. The performance of flexible manipulator system depends on various factors like dimensional factors, material properties, payload applied to the system, etc. One can use various algorithms to find efficiency and performance at different parts such as beginning, middle and end of the manipulator system. Once all the performance parameters are evaluated, one can design and redesign to get the best performance out of it. This thesis explains a process of deploying various algorithms over the web to understand the characteristics and performance of a flexible manipulator system depending upon number of inputs. The programming is done on various platforms and is done in different programming languages. One should know the fundamentals of programming to understand the flow. The project was initiated by various other researchers. The software environment developed was known as SCEFMAS. The SCEFMAS was first initiated during the year of 1994. The initial version of the SCEFMAS includes finite difference (FD) simulation along with classical control methods (open-loop and closed-loop methods). The current version of SCEFMAS has a number of areas: finite difference, finite element, genetic algorithm, neural network, open-loop and closed-loop controls, and advanced controls. This version of SCEFMAS is now used for teaching within a Master of Science program offered by the Department of Automatic Control and Systems Engineering of the University of Sheffield, UK.

1.1. Literature Survey

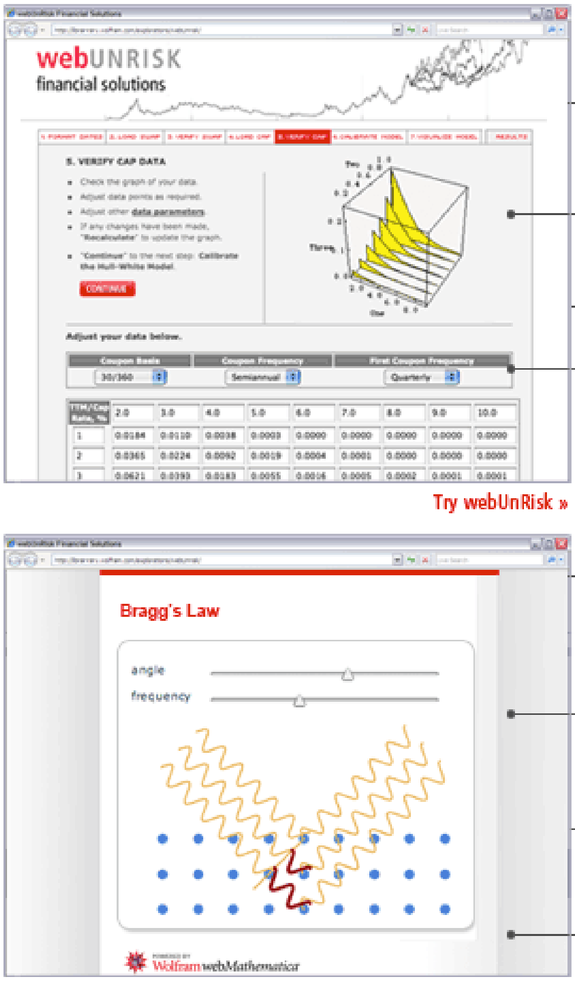
There are a number of software packages that are used for developing web-based simulation environments. Some of them are Mathematica, LabVIEW, GNU Octave, and MATLAB. This section will describe the functionalities of these software packages.

1.1.1 Mathematica

Mathematica is a computational software program used in many scientific, engineering, mathematical and computing fields, based on symbolic mathematics. It was conceived by Stephen Wolfram and is developed by Wolfram Research of Champaign, Illinois. The Wolfram Language is the programming language used in Mathematica [1]. There are several ways to deploy applications written in Mathematica:

- Mathematica Player Pro is a runtime version of Mathematica that will run any Mathematica application but does not allow editing or creation of the code.
- A free-of-charge version, Wolfram CDF Player, is provided for running Mathematica programs that have been saved in the Computable Document Format (CDF). It can also view standard Mathematica files, but not run them. It includes plugins for common web browsers on Windows and Macintosh. WebMathematica allows a web browser to act as a front end to a remote Mathematica server. It is designed to allow a user-written application to be remotely accessed via a browser on any platform. It may not be used to give full access to Mathematica.
- Mathematica code can be converted to C code or to an automatically generated DLL.
- Mathematica code can be run on a Wolfram cloud service as a web app or as an API.

WebMathematica adds interactive calculations and visualization to websites by integrating Mathematica with the latest web server technology (Figures 1 and 2) [2].



webUNRISK
financial solutions

1. FORMAT DATES | 2. LOAD SWAP | 3. VERIFY SWAP | 4. LOAD CAP | 5. VERIFY CAP | 6. CALIBRATE MODEL | 7. VISUALIZE MODEL | RESULTS

5. VERIFY CAP DATA

- Check the graph of your data.
- Adjust data points as required.
- Adjust other data parameters.
- If any changes have been made, "Recalculate" to update the graph.
- "Continue" to the next step: Calibrate the Hull-White Model.

CONTINUE

Adjust your data below.

Coupon Basis	Coupon Frequency	First Coupon Frequency
30/360	Semiannual	Quarterly

TTM/Cap Rate, %	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
1	0.0184	0.0110	0.0038	0.0003	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0365	0.0224	0.0092	0.0019	0.0004	0.0001	0.0000	0.0000	0.0000
3	0.0621	0.0393	0.0183	0.0055	0.0016	0.0005	0.0002	0.0001	0.0001

Try webUnRisk »

Bragg's Law

Client technologies such as JavaScript and Java applets work seamlessly with web*Mathematica*.

Generate and display graphics and visualizations dynamically.

Adjust your calculations interactively.

Use familiar web interface elements such as buttons, drop-down lists, and text fields to control computations.

web*Mathematica* custom tags integrate strongly with the standard Java web technologies Java Servlet and JavaServer Pages.

Use sliders and other dynamic features to adjust parameters and generate new results interactively.

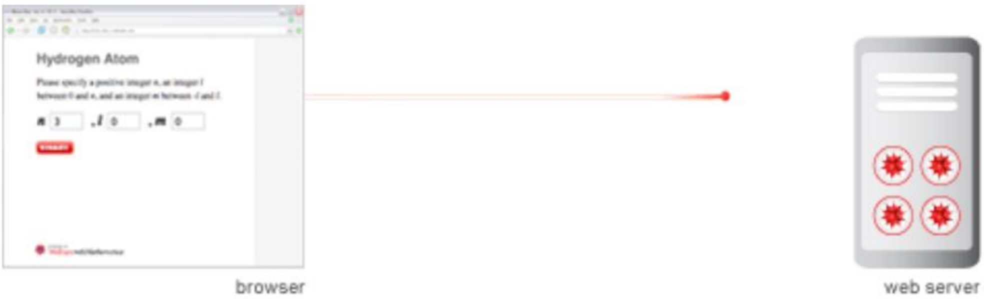
Display output on the web page or create downloadable reports in PDF, notebook, or other formats.

Use templates to give a consistent, professional appearance.

Figure 1: Client technologies [2].

Mathematica:

1. Make Request:



Hydrogen Atom

Please specify a positive integer n, an integer l between 0 and n, and an integer m between -l and l

n 3 , l 0 , m 0

Figure 2: Request from client to server [2].

The browser sends an HTTP request to the web server. The request references a particular WebMathematica page and includes variables and their values. Then the web server performs any preprocessing steps, such as authentication, and forwards the request to WebMathematica. WebMathematica's kernel manager acquires a Mathematica kernel for the request from a pool of preinitialized kernels. Any variables and values are sent to this kernel (Figure 3) [2].

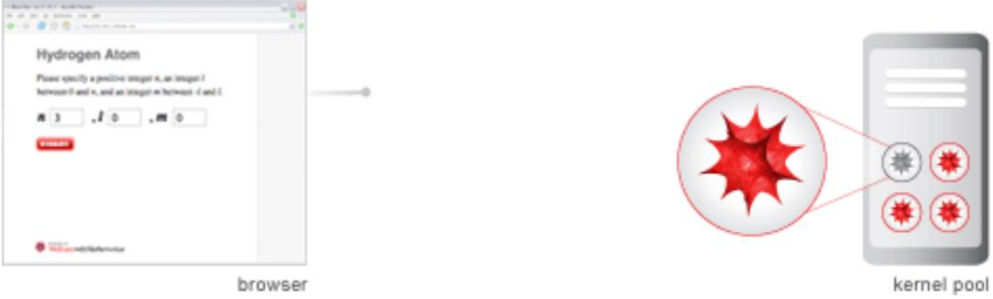
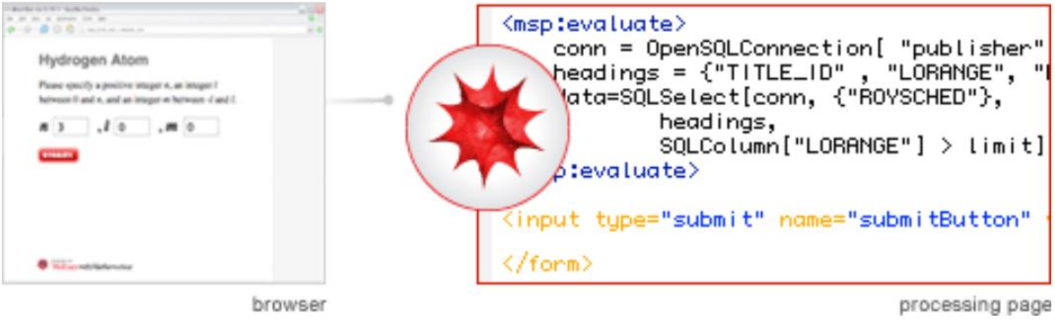


Figure 3: Acquire kernel [2].

2. Process Page:

The Mathematica kernel loads the web page and processes any WebMathematica tags. The kernel builds and then returns the result (Figure 4) [2].



```
<msp:evaluate>
    conn = OpenSQLConnection[ "publisher"
    headings = {"TITLE_ID" , "LORANGE", "
    ata=SQLSelect[conn, {"ROYSCHED"},
            headings,
            SQLColumn["LORANGE"] > limit]
    p:evaluate>

<input type="submit" name="submitButton"

</form>
```

Figure 4: Process page kernel [2].

3. Released Kernel:

WebMathematica accepts the response and adds all of the necessary HTTP headers for return to the browser. WebMathematica then clears any temporary settings in the Mathematica kernel and releases the kernel to its pool of available kernels (Figure 5).

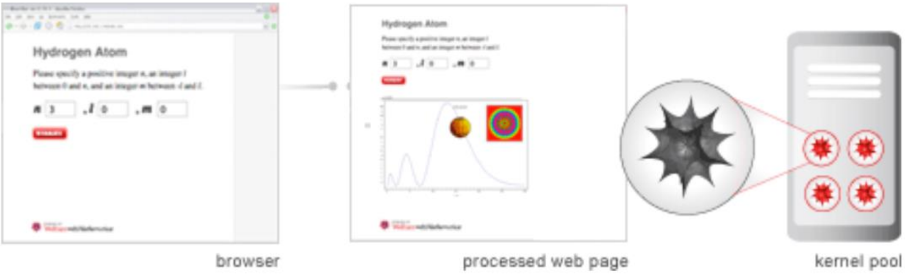


Figure 5: Released kernel [2].

4. Return Result:

The web server performs any post-processing steps and returns an HTML response—which may use applets, plug-ins, or other features of dynamic HTML—to the browser. Alternatively, the response can be in some other format, such as MathML, TeX, or a Mathematica notebook (Figure 6) [2].

Hydrogen Atom
browser
web server

Figure 6: Result displayed on web page [2].

1.1.2 LabVIEW

LavVIEW is a system-design platform and development environment for a visual programming language from National Instruments. The graphical language is named "G" (not to be confused with G-code). LabVIEW is commonly used for data acquisition, instrument control, and industrial automation on a variety of platforms including Microsoft Windows, various versions of UNIX, Linux, and Mac OS X [6].

Support for web services was introduced in LabVIEW 8.6 for Windows and LabVIEW real-time targets. The web services runtime engine in 8.6 and 2009 is loaded and run by LabVIEW's built-in web server. The web server is also available for use on Macintosh and Linux versions of LabVIEW, but without web services support. Web services development requires a Windows version of LabVIEW 8.6, 2009 or 2010 with a Full, Professional, or Student Edition license [4].

In LabVIEW 2010 the Application web Server is installed and runs by default on Windows PCs. This web server hosts all LabVIEW-built Web services and launches at system boot. Unlike earlier versions of the LabVIEW Web Server, the Application Web Server does not require LabVIEW to be running. The Application Web Server can be configured through the NI Web Configuration and Monitoring utility in a web browser. To use this utility you must install Microsoft Silverlight®. To launch the Web Configuration and Monitoring utility for your local machine, point a web browser to http://localhost:3580. From here you must select the Web Server Configuration Tool [4].

To enable the Application Web Server for real-time targets, you must install the NI Application Web Server to the target. In order to remotely configure the Application Web Server on an RT system, you must also install NI System Configuration, NI System Configuration network support, and NI Web-based Configuration and Monitoring [4].

Example: Creating a Web Service [4].

Develop a web service in a LabVIEW project to group together and configure the VIs and other files that will make up the web service. Complete the following steps to create a web service project and two HTTP method VIs [4]:

1. Create a LabVIEW project to organize the web service files. Save the project as Tutorial.lvproj.
2. Right-click **My Computer** and select **New»Web Service**. LabVIEW adds a web service project item and folders under the target.

3. Right-click the web service project item, select **Rename**, and name the web service TutorialService. The project appears as Figure 7:

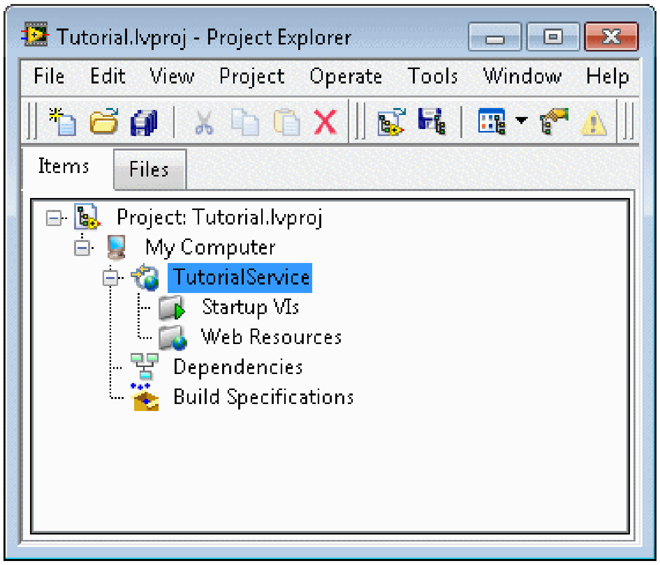


Figure 7: Creating web service [4].

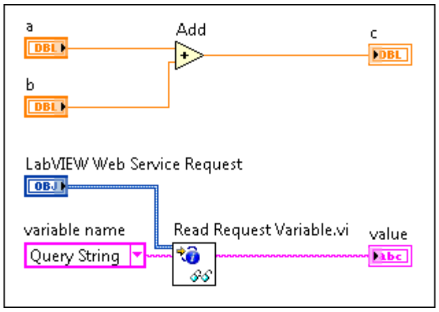
4. Right-click **Web Resources** and select **New VI** to create a new HTTP method VI, which is a VI that receives HTTP requests from clients and returns data to clients.
5. LabVIEW opens a new VI from a template.
6. In the new VI, build the block diagram, front panel, and connector pane as shown in the following Figure 8.
7. When you publish this VI as part of a Web service, a web client can send an HTTP request to invoke it. When invoked, this VI performs the following actions [4]:

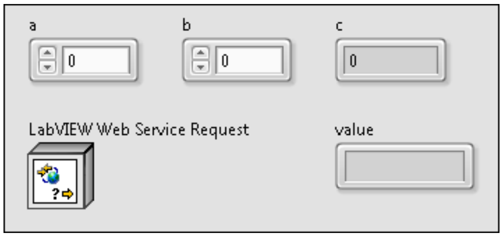
    i. Receives the values of any controls assigned to the connector pane (*except* **LabVIEW Web Service Request**) from the HTTP request the client made. **LabVIEW Web Service Request** acts like a refnum by identifying the current HTTP request.

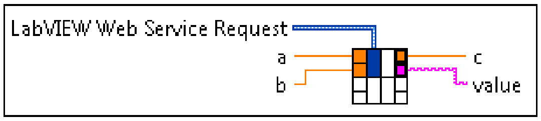
    ii. Computes the sum of **a** and **b**.

    iii. Returns the sum of those values to the client via the **c** indicator assigned to the connector pane.

iv. Executes the Read Request Variable VI and returns the query string the client included in the current HTTP request, which **LabVIEW Web Service Request** identifies. Notice that each control and indicator appears on the connector pane. Web services can receive data from clients or send data to clients only through inputs and outputs assigned to the connector pane of an HTTP method VI. Save the VI as Add.vi.

2. 

3. 

4. 

5. Figure 8: LabVIEW [4].

8. Select **File»Save As** and select **Copy»Open additional copy** in the save dialog. Ensure the ‘**Add Copy**’ **to Tutorial.lvproj** checkbox contains a checkmark, and then click **Continue**.
9. Save the new copy as Subtract.vi.
10. Open the block diagram of Subtract.vi and replace the Add function with the Subtract function. Rewire the controls and indicators.
11. Return to the project, right-click **Web Resources**, and select **New Web Resource**. Rename the new project item AdditionalOperations.
12. Drag Subtract.vi under the **AdditionalOperations** project item. Save the project, which should appear as in the Figure 9.

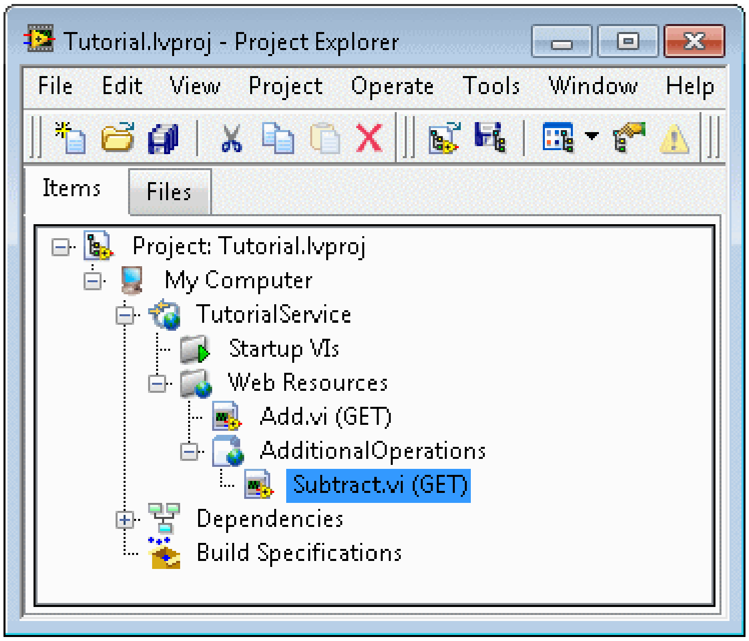


Figure 9: Project explorer [4].

Notice the following features in the project tree [4]:

- LabVIEW appends an HTTP request type to the name of each HTTP method VI, which by default is GET. You can change the type of HTTP request that clients send to a specific HTTP method VI later in this tutorial.
- The web service project item [4] also contains a **Startup VIs** folder. This tutorial does not use startup VIs, but you can find more information about them at the end of the tutorial.

1.1.3 MATLAB Web Server Environment

The MATLAB® web server enables you to create MATLAB applications that use the capabilities of the World Wide Web to send data to MATLAB for computation and to display the results in a web browser. In the simplest configuration, a web browser runs on your client workstation while MATLAB, the MATLAB web server and the web server daemon (http) run on another machine [5]. Figure 10 shows the basic configuration of web server application [5].

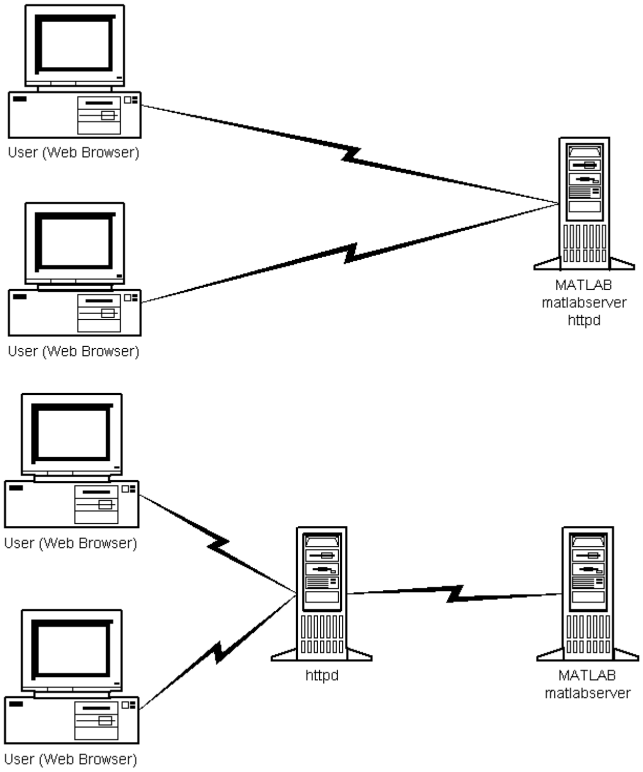


Figure 10: Web Server Configuration [5].

The MATLAB web server depends upon TCP/IP networking for transmission of data between the client system and MATLAB. The required networking Software and hardware must be installed on your system prior to using the MATLAB Web Server [5].

Building MATLAB Web Server Applications (Using HTML) :

MATLAB web server applications are a combination of M-files, Hypertext Markup Language (HTML) and graphics. Knowledge of MATLAB programming and basic HTML are the only requirements [5]. The application development process requires a small number of simple steps [5]:

1) Create the HTML documents for collection of the input data from users
2) Display output. You can code the input documents using a text editor to input HTML directly, or you can use one of the commercially available HTML authoring systems, such as Front Page from Microsoft, PageMill from Adobe, or HoT MetaL from SoftQuad.

3) List the application name and associated configuration data in the configuration files [5] matweb.conf.
4) Write a MATLAB M-file that:
   a) Receives the data entered in the HTML input form.
   b) Analyzes the data and generates any requested graphics.
   c) Places the output data into a MATLAB structure.
   d) Calls html rep to place the output data into an HTML output document template [5].

The maximum amount of HTML data you can receive from MATLAB is 256 KB.

The MATLAB web server has the same supporting hardware and software requirements as MATLAB 5, except for memory required. MATLAB hardware and software requirements are documented in the *Installation Guide* for your computer. Memory requirements while running the MATLAB web server vary with the number of MATLABs configured. Each MATLAB running under the MATLAB. Web server consumes 256 KB of memory. The MATLAB web server requires that TCP/IP networking software must be installed on your computer. Consult the document *Known Software and Documentation Problems* for any last-minute changes to hardware or software requirements [5].

2. PROJECT DESCRIPTION AND PRELIMINARY WORK

In this project, we are going to deploy the finite difference algorithm of flexible manipulators over the web page so that it could be adopted in research globally. We are going to plot the data of hub angle, hub velocity, hub acceleration, endpoint displacement, endpoint acceleration, input torque, and residual movements over the web page depend upon the material properties, simulation inputs and other inputs as provided by the user. In this way any user can test the algorithm over the web remotely to access the in-depth information of the algorithm.

The original algorithm code is written in MATLAB and all the inputs provided by web page are passed to DLL file that is created by MATLAB Compiler SDK [3]. These inputs are taken into MATLAB original file and computations are done to get the inputs of finite difference algorithm. The finite difference is now converted into C code which is called at appropriate times to get the result of all those inputs.

Once all the outputs are ready, these are sent back to web page through output arguments from DLL file to main code written in C# in Visual Studio. This file is referenced to the web page back which captures the output and plots the data appropriately on a graph. Analyzed data is presented to the user over the web page. As input and outputs are shown in one single page, users need not go back and forward to analyze inputs and outputs.

The main aim of this project is to develop a web-based simulation environment for simulation and control of a single link manipulator system. The process involves a number of steps starting from MATLAB code and web design at the end. The steps are: writing m-files for algorithm implementation, converting m-files to dynamic link library (DLL) using MATLAB Compiler SDK, developing console application that references the developed DLL and storing the output result in text files as database/flash file, ASP .NET to use the text files and display the results. A block diagram showing the process is shown in Figure 11.

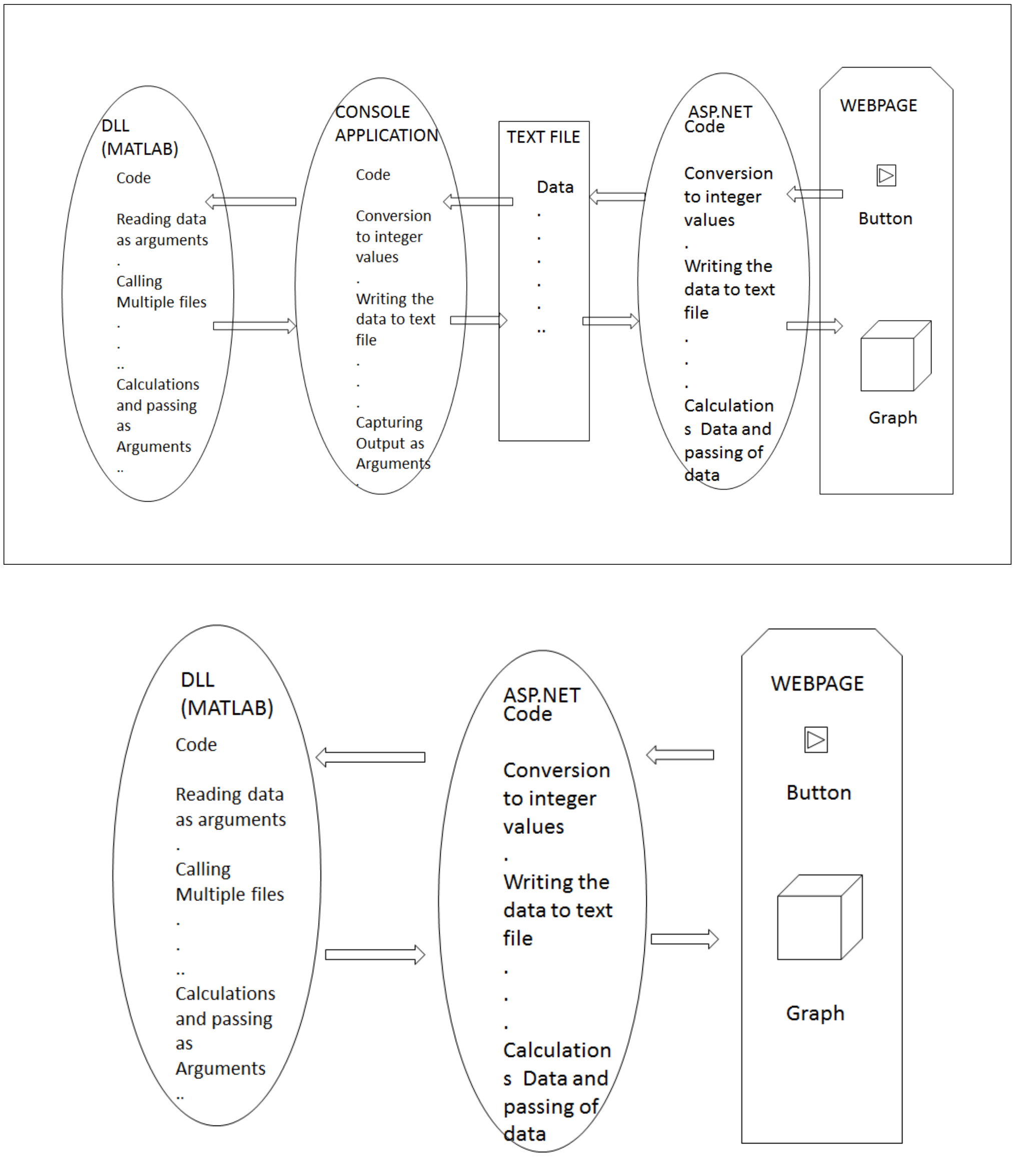

DLL
(MATLAB)
Code
Reading data as arguments
.
Calling Multiple files
.
.
..
Calculations and passing as Arguments
..
CONSOLE APPLICATION
Code
Conversion to integer values
.
Writing the data to text file
.
.
.
Capturing Output as Arguments
.
TEXT FILE
Data
.
.
.
.
.
..
ASP.NET Code
Conversion to integer values
.
Writing the data to text file
.
.
.
Calculations Data and passing of data
WEBPAGE
Button
Graph
DLL
(MATLAB)
Code
Reading data as arguments
.
Calling Multiple files
.
.
..
Calculations and passing as Arguments
..
ASP.NET Code
Conversion to integer values
.
Writing the data to text file
.
.
.
Calculations Data and passing of data
WEBPAGE
Button
Graph

Figure 11: Implementation block diagram with and without Console.

We can also exclude the Console application and text file in order to make a direct connection between ASP.NET and MATLAB. In this case, the block diagram will look like Figure 11.

2.1 MATLAB Coding

The finite difference (FD) algorithm for the flexible manipulator system is coded in MATLAB. A diagram showing the process is shown in Figure 12. All the user-defined specifications like material properties and simulation parameters are provided as inputs via Console application in the form of arguments to the MATLAB main function (labelled as FD). This main function calls a number of sub-functions at various stages. The sub-functions are: Simulink block coded function, generation of output function, and data scaling function. After generation, the results are passed to the Console. A piece of MATLAB code is shown in Figure 13.

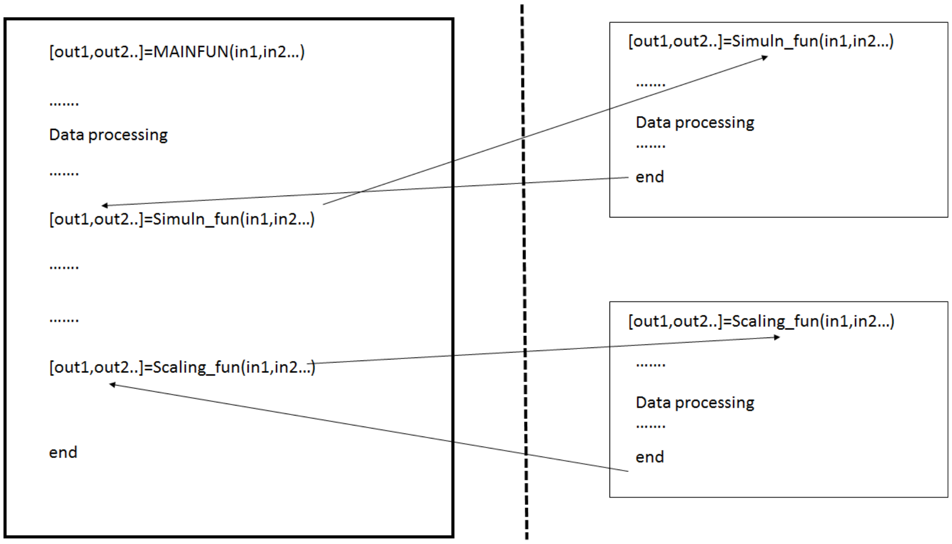

[out1,out2..]=MAINFUN(in1,in2...)
....
Data processing
....
[out1,out2..]=Simuln_fun(in1,in2...)
....
....
[out1,out2..]=Scaling_fun(in1,in2...)
end
[out1,out2..]=Simuln_fun(in1,in2...)
....
Data processing
....
end
[out1,out2..]=Scaling_fun(in1,in2...)
....
Data processing
....
end

Figure 12: MATLAB implementation.

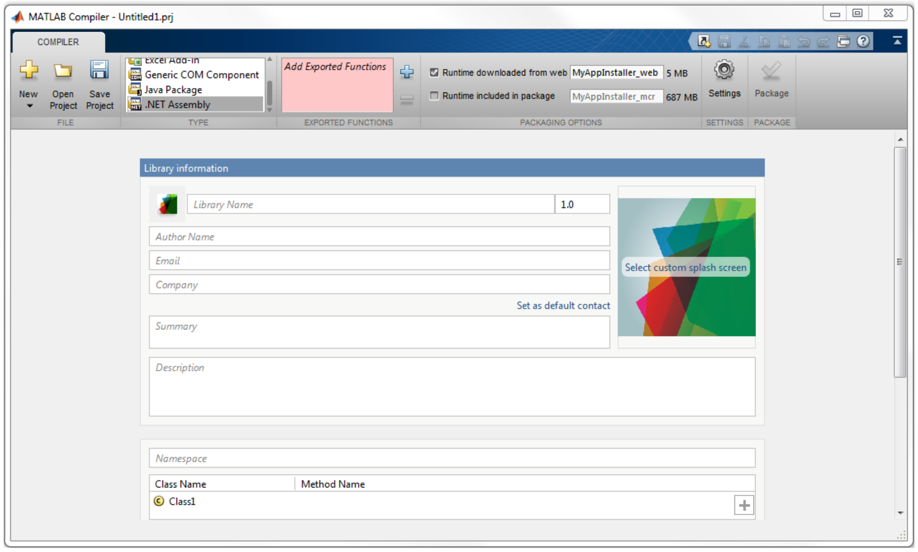
```
• \function
  new_result=fd1(fe1,L1,T1,W1,Ih1,Mp1,time1,n1,lam1,elem1,Ds1,E1,row1,InpOpt1,i123
  ,select_output)
• global  Ds;
• global time;
• global n;
• global lam;
• global W;
• global T;
• global L;
• global E;
• global row;
• global Ih;
• global Mp;
• .
• .
• .
• [ite,yin,t,aa,f,Pn1]=results(yout,time,i123,yin);
```

Figure 13: A piece of MATLAB code.

To make a DLL file from the developed m-file one needs to use the MATLAB Compiler SDK and select the .NET Assembly to generate necessary library for conversion (Figure 14). The steps for the whole conversion process are as below:

1) Add the file by clicking the button
2) Select the main file, it automatically adds dependent files
3) Select all the necessary files, such as Excel files and data files which are required by the main function to execute.
4) Add the name of the DLL
5) Add any other files needed

6) Click on package button.



MATLAB Compiler - Untitled1.prj

☑ Runtime downloaded from web MyAppInstaller_web 5 MB

□ Runtime included in package MyAppInstaller_mcr 687 MB

Library Name 1.0

Figure 14: Screenshot of MATLAB Compiler SDK.

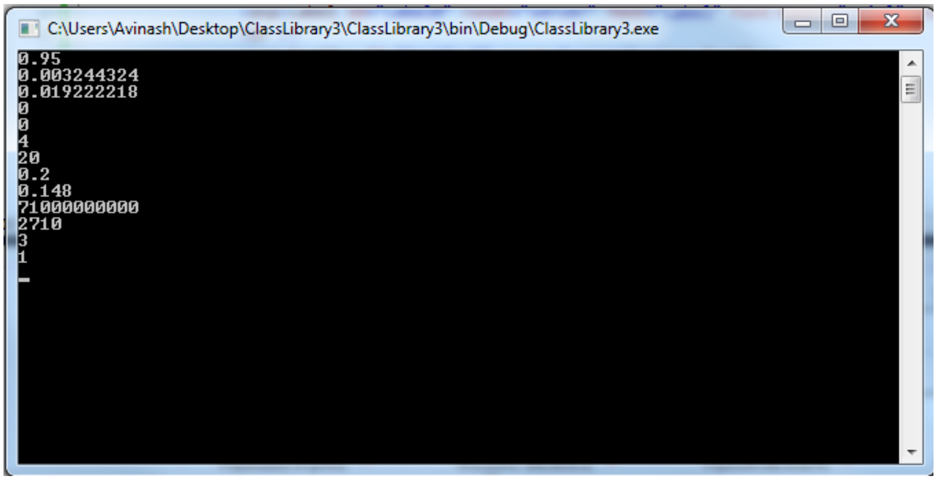
2.2 Microsoft Visual Studio .NET Assembly (Console Coding and ASP.NET)

Compiler SDK is responsible for converting library of MATLAB to .NET Assembly and C# library. This is needed as Microsoft Visual Studio is based on .NET framework. When data is passed directly from Matlab to .NET framework (which can also have C# code at its back end) the data size is large and that increases the processing time extensively and it shows an error message as “out of memory”. To address this issue Console application is used as an intermediate stage between the MATLAB-created DLL and ASP.NET web page. When a client interacts with the web page the values are stored in a text file and passed to the Console application. Console application proceses the input data and forwards to MATLAB code. Once MATLAB takes care of the user request the output data passes back to the Console application which creates a new text file with the output data. The ASP.NET web page uses the new output text file created to plot the graphs.

The first and foremost step in .Net framework is to reference the DLL file from the redistribution folder which is created by MATLAB Compiler SDK from the above step. The next important step is to add another reference called as MW Array which is required by Visual Studio to understand the output data from MATLAB in that format. The Console application acts as an intermediate communicator between MATLAB Compiler SDK created DLL and ASP.NET web page. It communicates through text files to ASP.NET. The inputs provided by the user will be stored in some text file and the Console application is triggered. The ASP.NET web page is kept on hold until the Console is done with its necessary operations. During that time, Console application reads the data from the text file and passes all the data through arguments to the DLL along with its own inputs. MATLAB files executes itself and produces outputs, which are passed back to Console application, which in turns stores those outputs in different text files according to their types.

2.3 Details of Console Programming

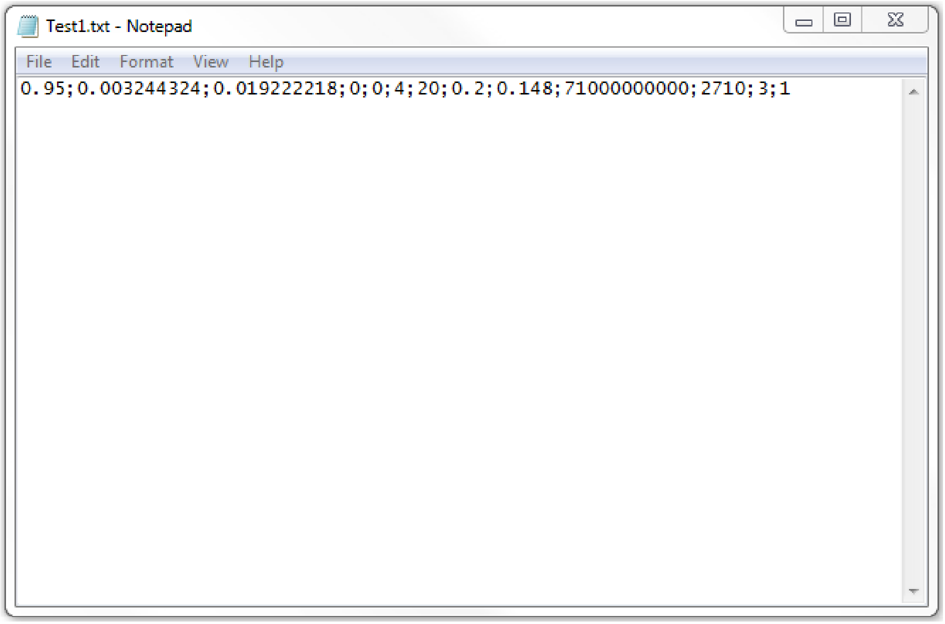
The first step in Console programming is to include all the necessary DLL files through in the reference block of solution. The next important step is to include the namespaces in order to include its class in the program. After initialization of variables, those variables are converted into MATLAB-based array format so that the data passed by the Console application is understandable by MATLAB. The input variables of data are read by the text file and is passed to DLL created by MATLAB and the output is passed back to Console application. This data is converted to double format and the intermediate output is displayed on the screen. The output is converted to string format and stored in text files. Totally, five text files were involved in the process, one for input and remaining for outputs. The data stored in the text files are separated by semicolon. The semicolon acts as a divider between data stored in the text files. Figure 15-17 show different text files for inputs and outputs and intermediate Console application output.



C:\Users\Avinash\Desktop\ClassLibrary3\ClassLibrary3\bin\Debug\ClassLibrary3.exe

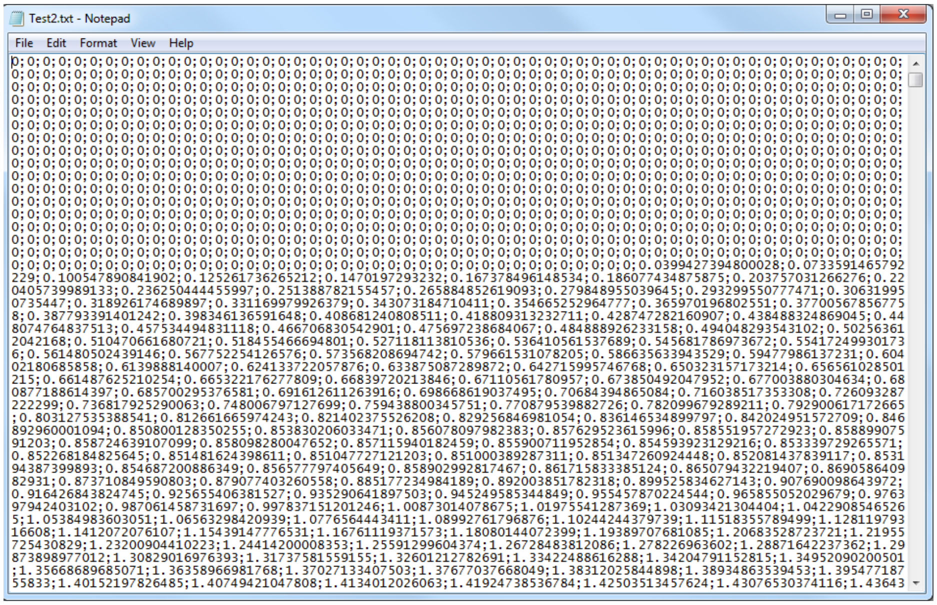
```
0.95
0.003244324
0.019222218
0
0
4
20
0.2
0.148
71000000000
2710
3
1
_
```

Figure 15: Intermediate output with Console application.



```
0.95;0.003244324;0.019222218;0;0;4;20;0.2;0.148;71000000000;2710;3;1
```

Figure 16: Text files for input data.



```
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0.0399427394800028;0.0733591465792
229;0.100547890841902;0.125261736265212;0.1470197293232;0.167378496148534;0.186077434875875;0.203757031266276;0.22
0405739989133;0.236250444455997;0.251388782155457;0.265884852619093;0.279848955039645;0.293299550777471;0.30631995
0735447;0.318926174689897;0.331169979926379;0.343073184710411;0.354665252964777;0.365970196802551;0.37700567856775
8;0.387793391401242;0.398346136591648;0.408681240808511;0.418809313232711;0.428747282160907;0.438488324869045;0.44
8074764837513;0.457534494831118;0.466706830542901;0.475697238684067;0.484888926233158;0.494048293543102;0.50256361
2042168;0.510470661680721;0.518455466694801;0.527118113810536;0.536410561537689;0.545681786973672;0.55417249930173
6;0.561480502439146;0.567752254126576;0.573568208694742;0.579661531078205;0.586635633943529;0.59477986137231;0.604
02180685858;0.6139888140007;0.624133722057876;0.633875087289872;0.642715995746768;0.650323157173214;0.656561028501
215;0.661487625210254;0.665322176277809;0.66839720213846;0.67110561780957;0.673850492047952;0.677003880304634;0.68
0877188614397;0.685700295376581;0.691612611263916;0.698668619037495;0.70684394865084;0.716038517353308;0.726093287
222299;0.736817925290063;0.748006797127699;0.759438800345751;0.770879539882726;0.782099679289211;0.792900617172665
;0.803127535388541;0.812661665974243;0.821402375526208;0.829256846981054;0.836146534899797;0.842024951572709;0.846
892960001094;0.850800128350255;0.853830206033471;0.856078097982383;0.857629523615996;0.858551957272923;0.858899075
91203;0.858724639107099;0.858098280047652;0.857115940182459;0.855900711952854;0.854593923129216;0.853339729265571;
0.852268184825645;0.851481624398611;0.851047727121203;0.851000389287311;0.851347260924448;0.852081437839117;0.8531
94387399893;0.854687200886349;0.856577797405649;0.858902992817467;0.861715833385124;0.865079432219407;0.8690586409
82931;0.873710849590803;0.879077403260558;0.885177234984189;0.892003851782318;0.899525834627143;0.907690098643972;
0.916426843824745;0.925655406381527;0.935290641897503;0.945249585344849;0.955457870224544;0.965855052029679;0.9763
97942403102;0.987061458731697;0.997837151201246;1.00873014078675;1.01975541287369;1.03093421304404;1.0422908546526
5;1.05384983603051;1.06563298420939;1.0776564443411;1.08992761796876;1.10244244379739;1.11518355789499;1.128119793
16608;1.1412072076107;1.15439147776531;1.16761119371573;1.18080144072399;1.19389707681085;1.20683528723721;1.21955
725430829;1.23200904410223;1.24414200008353;1.25591299604374;1.26728483812086;1.278226963602;1.28871642237362;1.29
873898977012;1.30829016976393;1.31737581559155;1.32601212782691;1.33422488616288;1.34204791152815;1.34952090200501
;1.35668689685071;1.36358966981768;1.37027133407503;1.37677037668049;1.38312025844898;1.38934863539453;1.395477187
55833;1.40152197826485;1.40749421047808;1.4134012026063;1.41924738536784;1.42503513457624;1.43076530374116;1.43643
```

Figure 17: Text files for output data.

Here are the steps involved in Console application [7]:

1) Install the .NET assembly and the MATLAB runtime.
2) Open Microsoft® Visual Studio®.
3) Create a new project.
4) Create a reference to your assembly.
5) Create a reference to the MWArray API. (The API is located in *MATLABROOT*\toolbox\dotnetbuilder\bin\*arch**version*\MWArray.dll.)
6) Make .NET namespaces available to your generated component and MWArray libraries. Add the following using statements to your C#/.NET code:
7) Initialize your classes before you use them.
8) Instantiate your component class.
9) Invoke your component. After you complete the tasks of initializing and instantiating the classes you are working with, invoke the function component. Invoke the component method using a signature containing both the number of output arguments expected and the number of input arguments the MATLAB function requires.

10) When calling your component, you can take advantage of implicit conversion from .NET types to MATLAB types by passing the native C# value directly to function.
11) The function method returns an array of MW Arrays.
12) Because class instantiation and method invocation make their exceptions at run time, you should enclose your code in a try-catch block to handle errors.
13) After you finish writing your code, you build and run it with Microsoft Visual Studio (Figure 18-19).

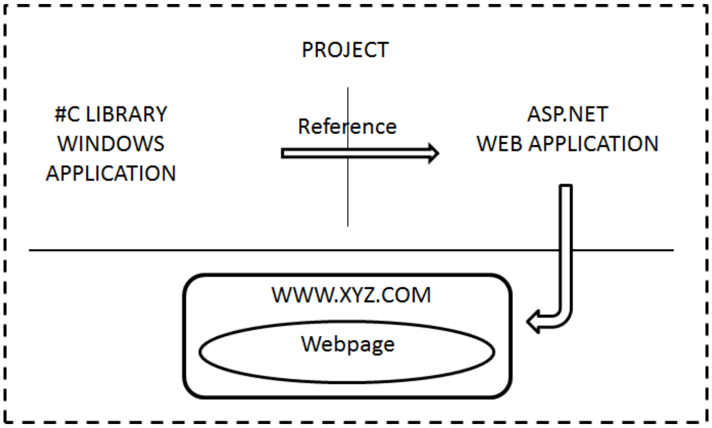


Figure 18: Microsoft Visual Studio implementation.

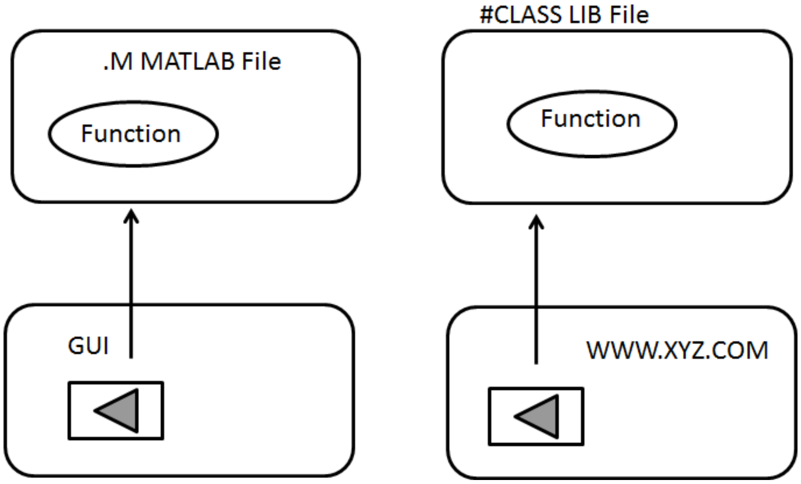


Figure 19: Relation between functions.

```
\using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using fd1;
using MathWorks.MATLAB.NET.Arrays;
using MathWorks.MATLAB.NET.Utility;

namespace fd1
{
  class Program
   {
       static void Main(string[] args)
       {
           fd1.Class1 obj = null;
           MWNumericArray input1 = null;
           MWNumericArray input2 = null;
           MWNumericArray input3 = null;
           MWNumericArray input4 = null;
```

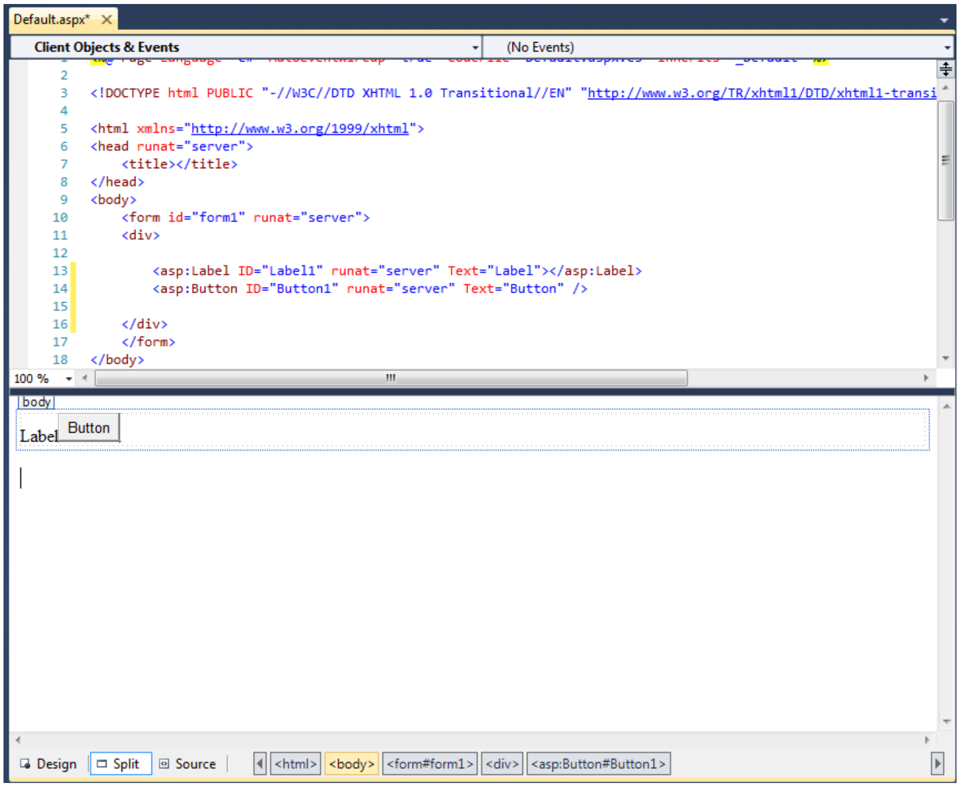
2.4 Designing the Web Page

ASP.NET reads the data from the necessary text files and stores those outputs in different variables. Those data is processed according to the needs the graph on the web page and the data is plotted.

Following are the steps required (Figure 20):

1) Open a new project in Visual Studio of ASP.NET.
2) Design various buttons, sliders and down drop menus as required by number of inputs.

3) Every attribute will have a backend calling function which needs to be related to various inputs being called to reference the inputs of DLL file.
4) Once you capture the output data in some variables, plot the data using some graphs on the web page provided, giving all the necessary information for graphs like pixels, interval time and index.



```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transi

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div>

        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
        <asp:Button ID="Button1" runat="server" Text="Button" />

    </div>
    </form>
</body>
```

Figure 20: Screenshot of web page design.

3. CHALLENGES AND RESOLUTIONS

As the main aim of this project is to deploy the finite difference algorithm over the web page, we started looking means of transferring the information. The original work was developed in the GUI version of MATLAB, which also used Simulink blocks in it. MATLAB released Compiler SDK along with Library Compiler which supports .Net Assembly with 2015 edition. This tool was used to transfer the language of the original code to DLL which could be integrated in a .Net application.

We need to generate the code from the Simulink block as Simulink is not supported by MATLAB Compiler SDK. Later referencing of data has to be done to send data to and fro between the code generated and MATLAB file (Figure 21).

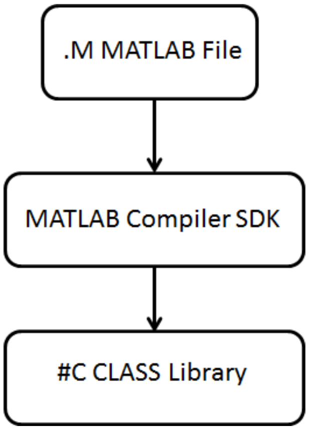


Figure 21: Conversion of library.

DLL should be in a format that which accepts inputs and generates output only in the form of arguments. The GUI version of MATLAB was completely dissolved and code was written instead to generate necessary inputs. The finite difference algorithm was re-written in a completely new format which retains the original algorithm process. Once, the algorithm was written, a

Simulink block was called which does further operations to generate all the main outputs like hub angle, endpoint displacement, etc.

Unfortunately, Compiler SDK which is used to deploy the code does not support GUI interfaces and Simulink block simulations. The main task involved in MATLAB coding was to re-write the complete algorithm in such a format that can take inputs and generate output which should be independent of Simulink block. All the blocks has to be redesigned and hard coded on MATLAB. The other issue that we faced during this process was to match all the sampling frequencies at different points in the Simulink block the same as that of code. Hence, complete Simulink block is re-written in M-file which is called at appropriate times by the main file along with the necessary inputs in order to artificially regenerate environment of Simulink block as before in original work.

Once, the outputs are ready at MATLAB it is ready to be deployed over the web page. The web page is designed to give the input to MATLAB file from ASP.NET. One of the issues which was faced here, as the data coming from MATLAB as output is huge, ASP.NET was throwing errors as it was going short of memory to plot it. The only solution to this problem here is to send the inputs from web page to text file which would be read by .NET Console application, which later on sends this data as an input to DLL which calls MATLAB runtime to get executed. The data is generated and passed to Console application of .Net which transfers this data to text files and creates a database. The array data which is written in text files are separated by semicolons so that it can be easily separated at the destination end and can could be stored in separate or same variables without any hassle. Hence, a lot of memory was saved at runtime to do the other important processes and output was later captured by ASP.NET to plot the necessary graphs on the web page.

3.1 Filter Implementations [8]

In signal processing domain, filters can be used to remove the unwanted component or feature from the signal. Filters can be used to either completely remove the signal components or uproot a few of them according to the application. Filters are majorly employed to remove the interference signals or to remove noise from the signal. Filters are employed in both frequency domain and time domain. They are classified on the basis of a few properties like linearity, causality, time variance or finite or infinite impulse response of the system.

Transfer function of filters is mostly defined in complex frequencies. It can be expressed in transfer domains like Z-transforms or Laplace transforms. This is a typical transfer function of a filter [8].

$$H(s) = \frac{Y(s)}{X(s)}$$

where S = **σ + j ω.**

3.1.1 FIR Filters (Finite Impulse Response [9])

FIR filters are basically those filters whose impulse response is finite. The impulse response of an Nth-order discrete-time FIR filter has exactly *N* + 1 samples before it then settles to zero. These filters only depend upon present and past inputs, which implies that they don't have any feedback associated with it. The basic transfer function of FIR filters is [9]:

$$y[n] = b_0x[n] + b_1x[n-1] + \cdots \ldots \ldots .. + b_Nx[n-N]$$

$$(x+a)^n = \sum_{i=0}^{n} b_i . x[n-i]$$

where b0, b1, b2, b3 ... represent filter coefficients; x[0], x[1], x[2] represent input samples; and y[i] represents the corresponding *i*-th output.

3.1.2 IIR Filters (Infinite Impulse Response) [10]

Unlike the finite response filters, IIR filters have infinite response time. Basically, all IIR filters are linear and time-invariant systems. These types of filters are found in digital and electronic domains. Unlike FIR filters, IIR filters never converge to zero in finite time. There are various trade-offs between FIR and IIR filters, which have to be taken into consideration depending upon application constraints. Some of advantages of IIR filters over FIR filters are: IIR filters are used for lower taps, they achieve desired characteristics in lower orders, etc. Advantages of FIR filters over IIR filters are: FIR filters are always stable and they linear characteristics, etc.

Typical transfer function of IIR filters is:

$$y[n] = \frac{1}{a_0}(b_0x[n] + b_1x[n-1] + \cdots + b_px[n-P] - (a_0y[n] + a_1y[n-1] + \cdots +$$
$$a_Qy[n-Q]\,)$$

where b0, b1, b2, b3 .. . , a0, a1, a2, . . . represent the filter coefficients, x[i] represents i-th input sample and y[j] represents the corresponding j-th output [10] .

In this project, IIR filter is implemented using classical technique. In this process, poles and zeros of analog low pass filter is designed and is transformed to digital domain through filter transformation and filter discretization. MATLAB is used to design these filters and data is passed to appropriate domain. Four types of IIR filters are implemented:

a. Butterworth
b. Chebyshev Type I
c. Chebyshev Type II
d. Elliptical

Following are the typical graphs obtained using this approach.

3.1.2.1 Butterworth Filter [11]

The Butterworth filter provides the best Taylor series approximation to the ideal low pass filter response at analog frequencies Ω = 0 and Ω = ∞; for any order N, the magnitude squared response has 2N – 1 zero derivatives at these locations (maximally flat at Ω = 0 and Ω = ∞). Response is monotonic overall, decreasing smoothly from Ω = 0 to Ω = ∞. ↓H (jΩ)↓=1/G2 at Ω = 1.

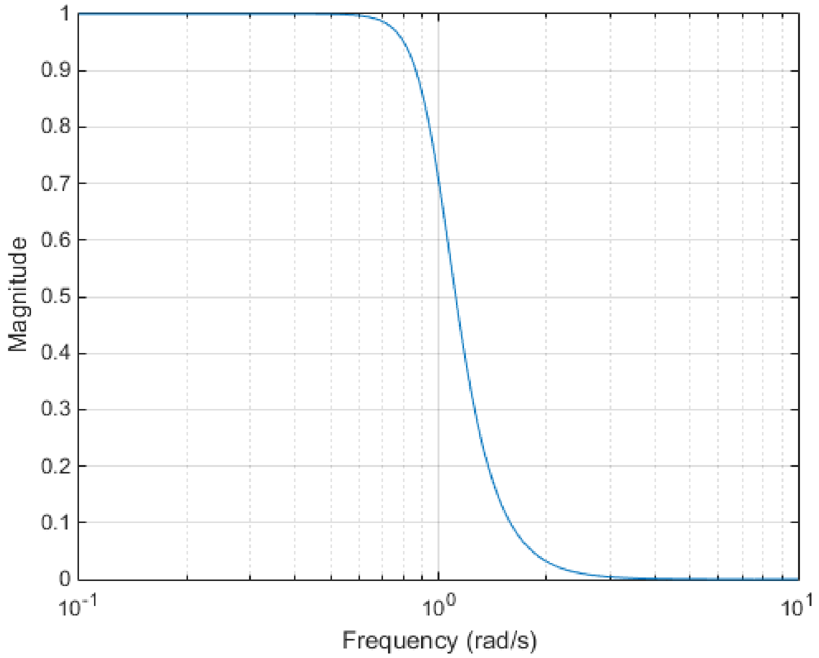


Figure 22: Butterworth low pass filter [11].

3.1.2.2 Chebyshev Type I Filter [11]

The Chebyshev Type I filter minimizes the absolute difference between the ideal and actual frequency response over the entire passband by incorporating an equal ripple of Rp dB in the passband. Stopband response is maximally flat. The transition from passband to stopband is more rapid than for the Butterworth filter. $\downarrow H(j\Omega)\downarrow=10^{-Rp/20}$ at $\Omega = 1$.

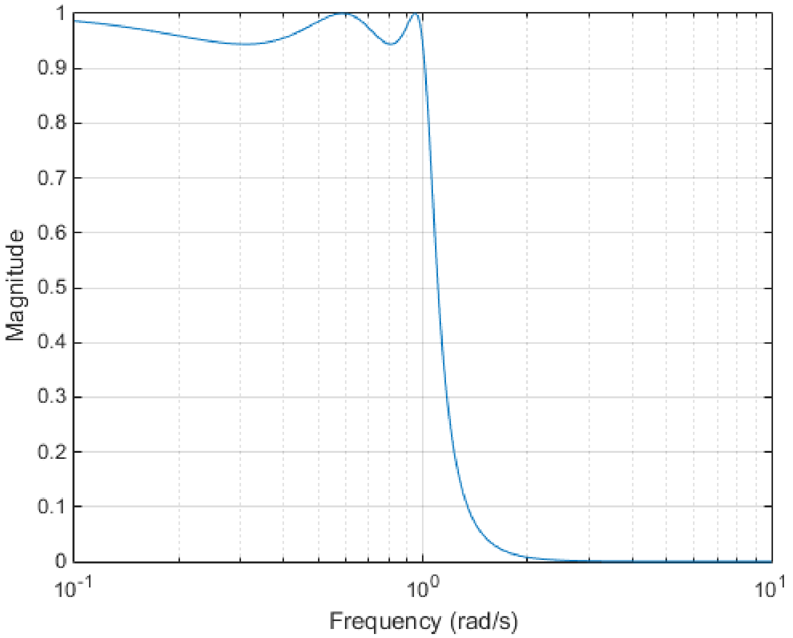


Figure 23: Chebyshev Type I low pass filter [11]

3.1.2.3 Chebyshev Type II Filter [11]

The Chebyshev Type II filter minimizes the absolute difference between the ideal and actual frequency response over the entire stopband by incorporating an equal ripple of Rs dB in the stopband. Passband response is maximally flat.

The stopband does not approach zero as quickly as the Type I filter (and does not approach zero at all for even-valued filter order *n*). The absence of ripple in the passband, however, is often an important advantage. $\downarrow H\,(j\Omega)\downarrow = 10^{-Rs/20}$ at $\Omega = 1$.

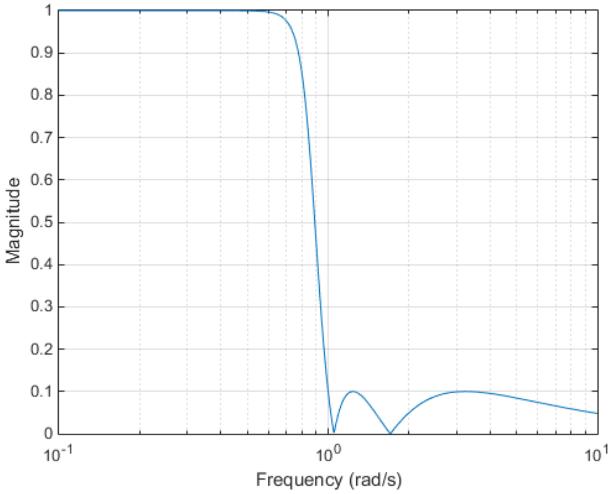


Figure 24: Chebyshev Type II Low pass Filter [11].

3.1.2.4 Elliptic Filter [11]

Elliptic filters are equiripple in both the passband and stopband. They generally meet filter requirements with the lowest order of any supported filter type. Given a filter order *n*, passband rippleRp in decibels, and stopband ripple Rs in decibels, elliptic filters minimize transition width. ↓H ($j\Omega$) ↓=$10^{-\mathrm{Rp}/20}$ at $\Omega = 1$ [11].

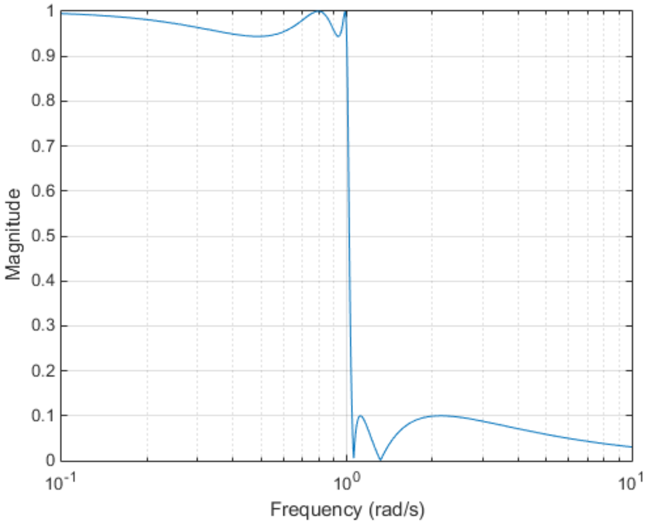
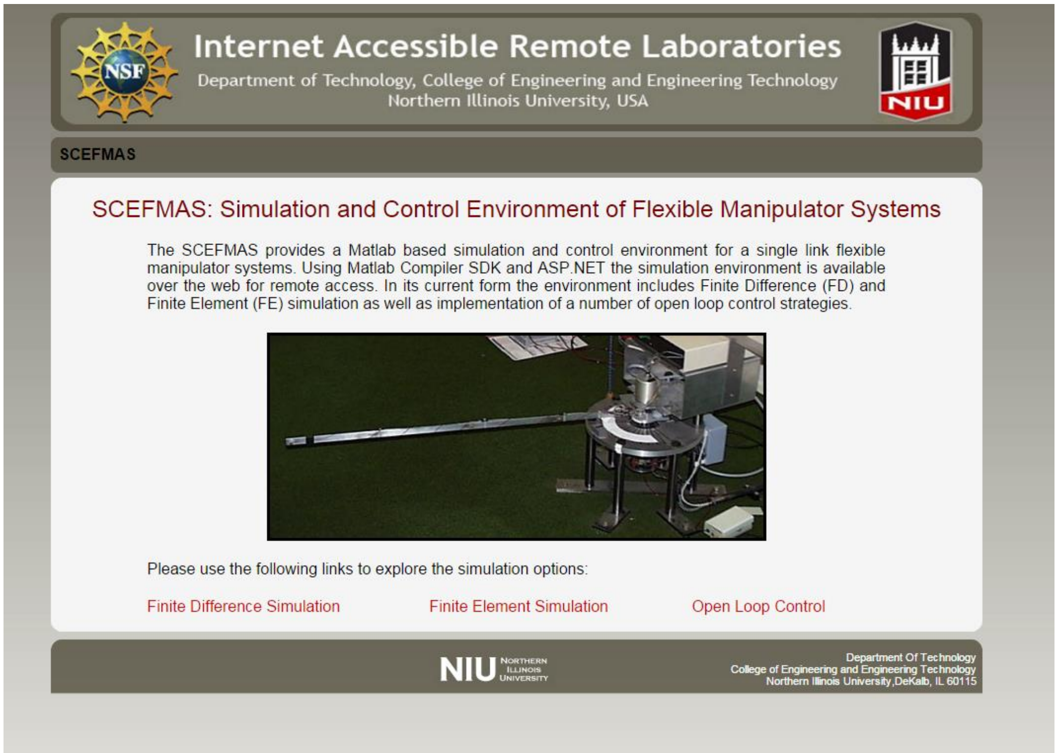


Figure 25: Elliptical low pass filter [11].

4 RESULTS

4.1 Flexible Manipulator



Internet Accessible Remote Laboratories

Department of Technology, College of Engineering and Engineering Technology
Northern Illinois University, USA

SCEFMAS

SCEFMAS: Simulation and Control Environment of Flexible Manipulator Systems

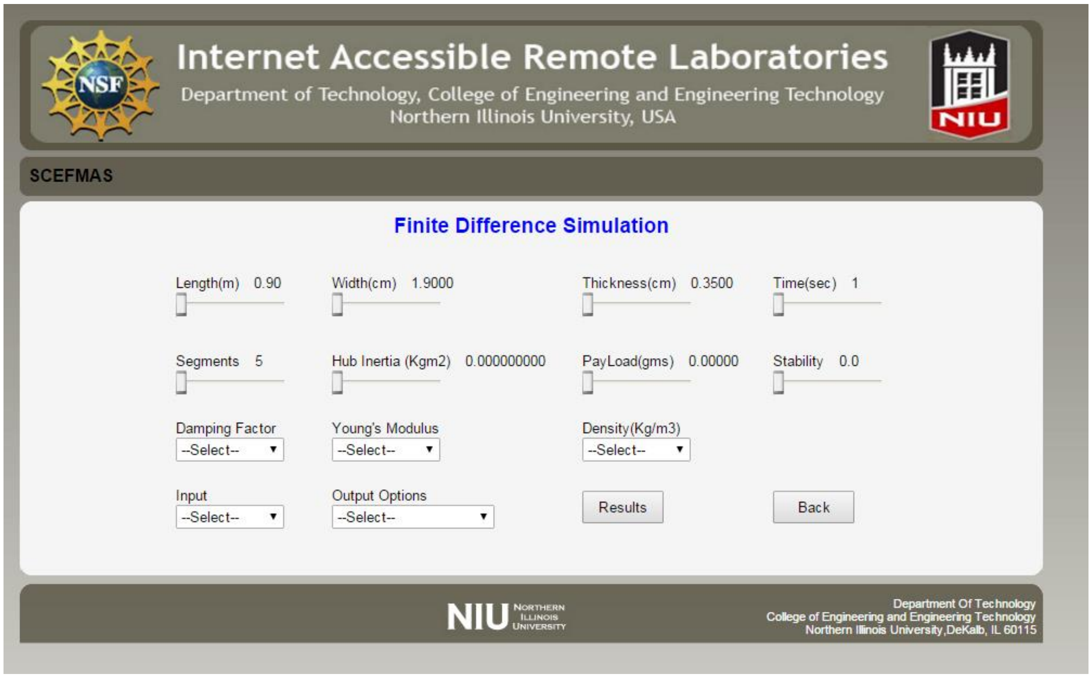
The SCEFMAS provides a Matlab based simulation and control environment for a single link flexible manipulator systems. Using Matlab Compiler SDK and ASP.NET the simulation environment is available over the web for remote access. In its current form the environment includes Finite Difference (FD) and Finite Element (FE) simulation as well as implementation of a number of open loop control strategies.

Please use the following links to explore the simulation options:

Finite Difference Simulation
Finite Element Simulation
Open Loop Control
NIU Northern Illinois University

Department Of Technology
College of Engineering and Engineering Technology
Northern Illinois University,DeKalb, IL 60115

Figure 26: SCEFMAS main page.



Internet Accessible Remote Laboratories

Department of Technology, College of Engineering and Engineering Technology
Northern Illinois University, USA

SCEFMAS

Finite Difference Simulation

Length(m) 0.90

Width(cm) 1.9000

Thickness(cm) 0.3500

Time(sec) 1

Segments 5

Hub Inertia (Kgm2) 0.000000000

PayLoad(gms) 0.00000

Stability 0.0

Damping Factor --Select--

Young's Modulus --Select--

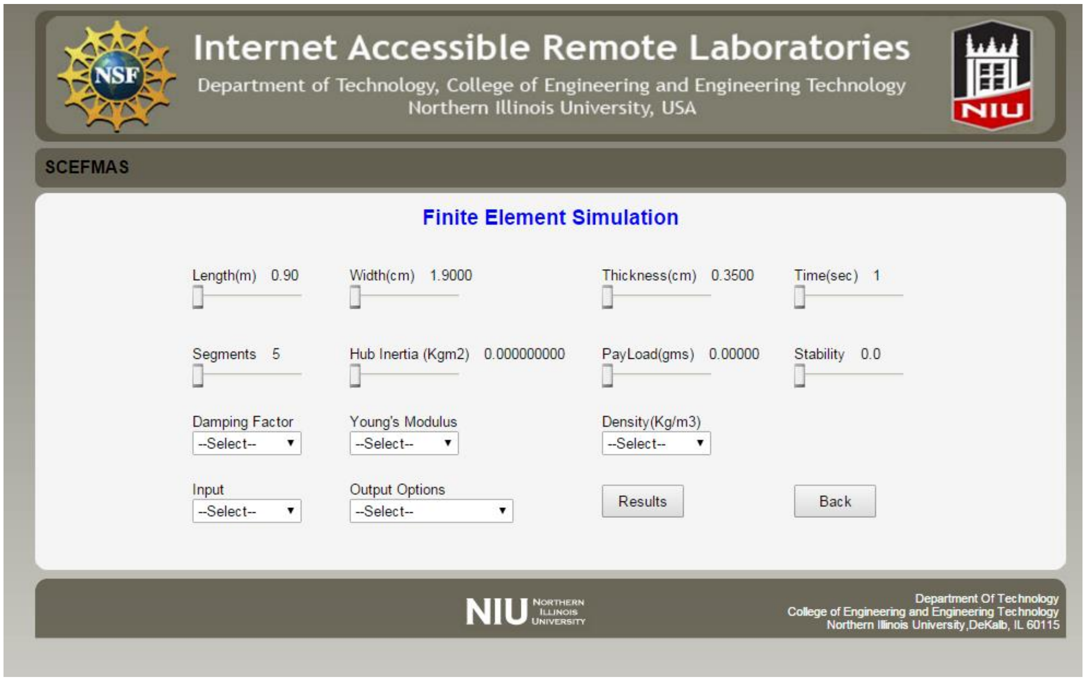
Density(Kg/m3) --Select--

Input --Select--

Output Options --Select--

Department Of Technology
College of Engineering and Engineering Technology
Northern Illinois University,DeKalb, IL 60115

Figure 27: finite difference input configuration page.



Internet Accessible Remote Laboratories

Department of Technology, College of Engineering and Engineering Technology
Northern Illinois University, USA

SCEFMAS

Finite Element Simulation

Length(m) 0.90

Width(cm) 1.9000

Thickness(cm) 0.3500

Time(sec) 1

Segments 5

Hub Inertia (Kgm2) 0.000000000

PayLoad(gms) 0.00000

Stability 0.0

Damping Factor --Select--

Young's Modulus --Select--

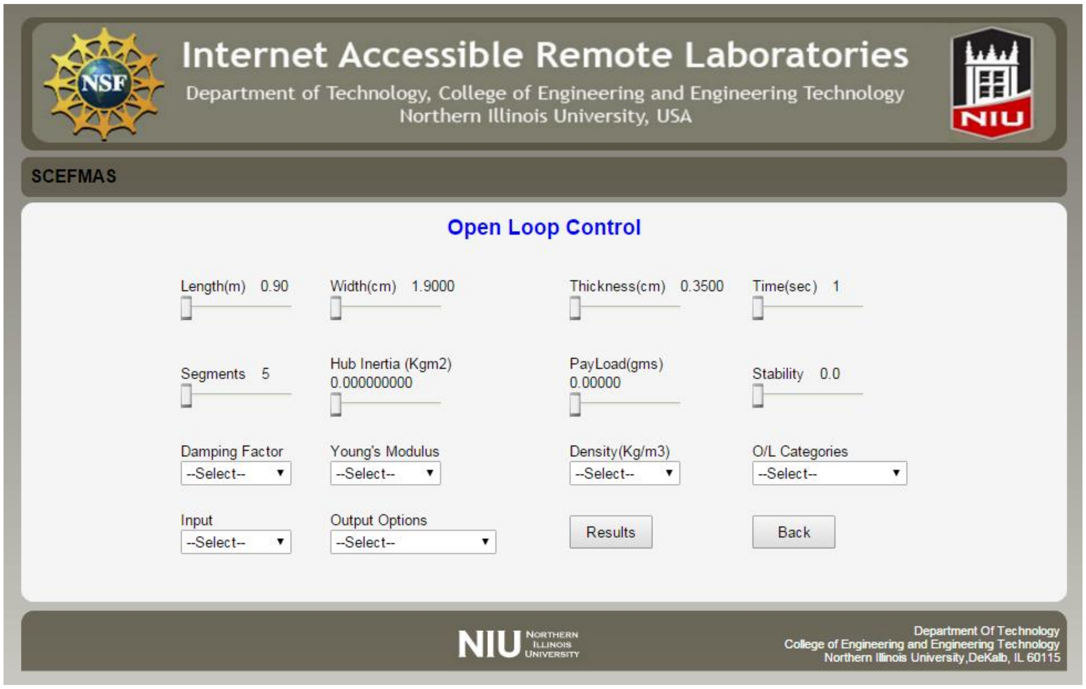
Density(Kg/m3) --Select--

Input --Select--

Output Options --Select--

Department Of Technology
College of Engineering and Engineering Technology
Northern Illinois University,DeKalb, IL 60115

Figure 28: finite element input configuration page.



Internet Accessible Remote Laboratories

Department of Technology, College of Engineering and Engineering Technology
Northern Illinois University, USA

SCEFMAS

Open Loop Control

Length(m) 0.90

Width(cm) 1.9000

Thickness(cm) 0.3500

Time(sec) 1

Segments 5

Hub Inertia (Kgm2) 0.000000000

PayLoad(gms) 0.00000

Stability 0.0

Department Of Technology
College of Engineering and Engineering Technology
Northern Illinois University,DeKalb, IL 60115

Figure 29: Finite Open Loop input configuration page.

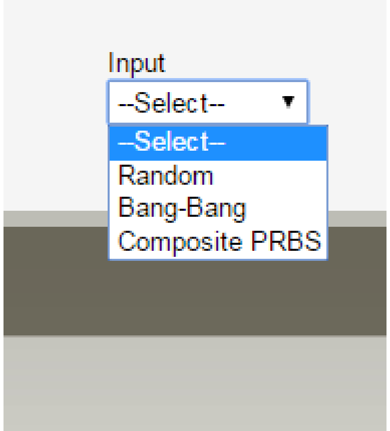


Figure 30: Input options.

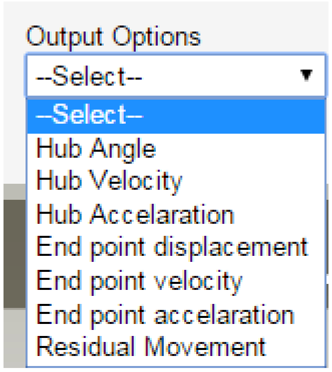
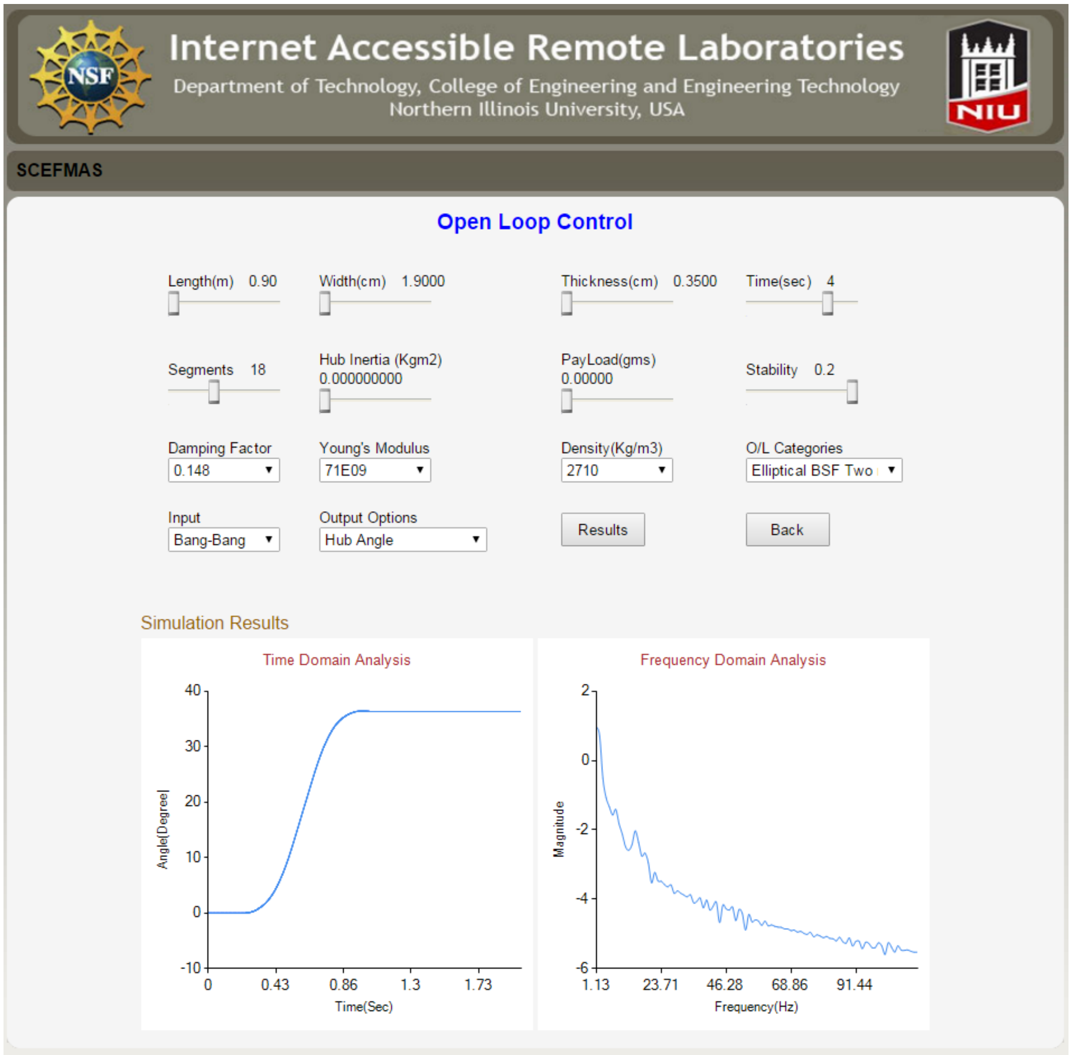


Figure 31: Output options.


Internet Accessible Remote Laboratories
Department of Technology, College of Engineering and Engineering Technology
Northern Illinois University, USA
NSF
NIU
SCEFMAS
Results
Back

Open Loop Control

Length(m) 0.90

Width(cm) 1.9000

Thickness(cm) 0.3500

Time(sec) 4

Segments 18

Hub Inertia (Kgm2) 0.000000000

PayLoad(gms) 0.00000

Stability 0.2

Damping Factor 0.148

Young's Modulus 71E09

Density(Kg/m3) 2710

O/L Categories Elliptical BSF Two

Input Bang-Bang

Output Options Hub Angle

Simulation Results

Time Domain Analysis
40
30
20
10
0
-10
Angle[Degree]
0
0.43
0.86
1.3
1.73
Time(Sec)
Frequency Domain Analysis
2
0
-2
-4
-6
Magnitude
1.13
23.71
46.28
68.86
91.44
Frequency(Hz)

Figure 32: Full screenshot.

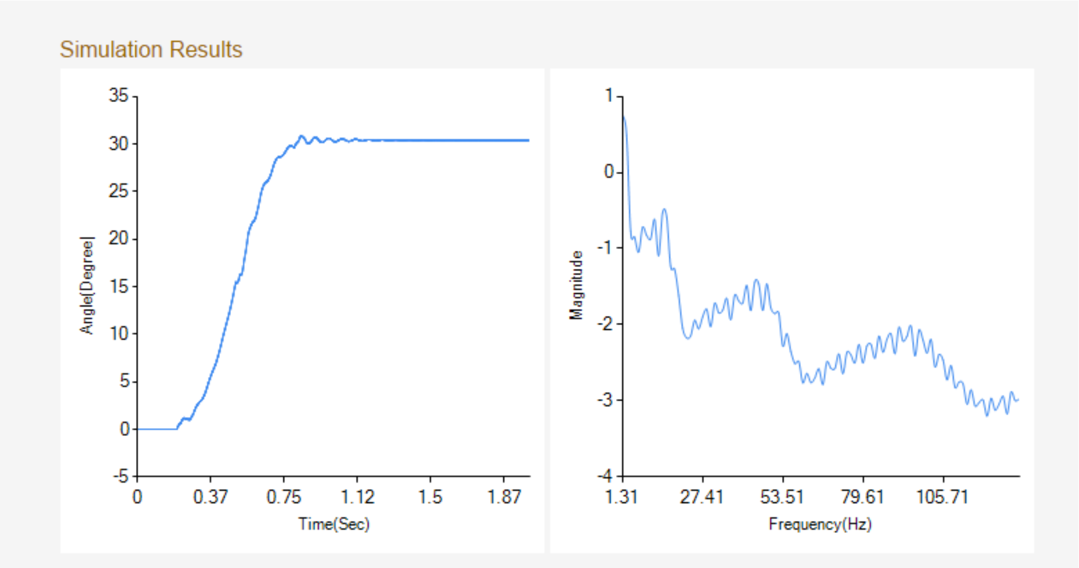


Figure 33: Hub angle without payload.

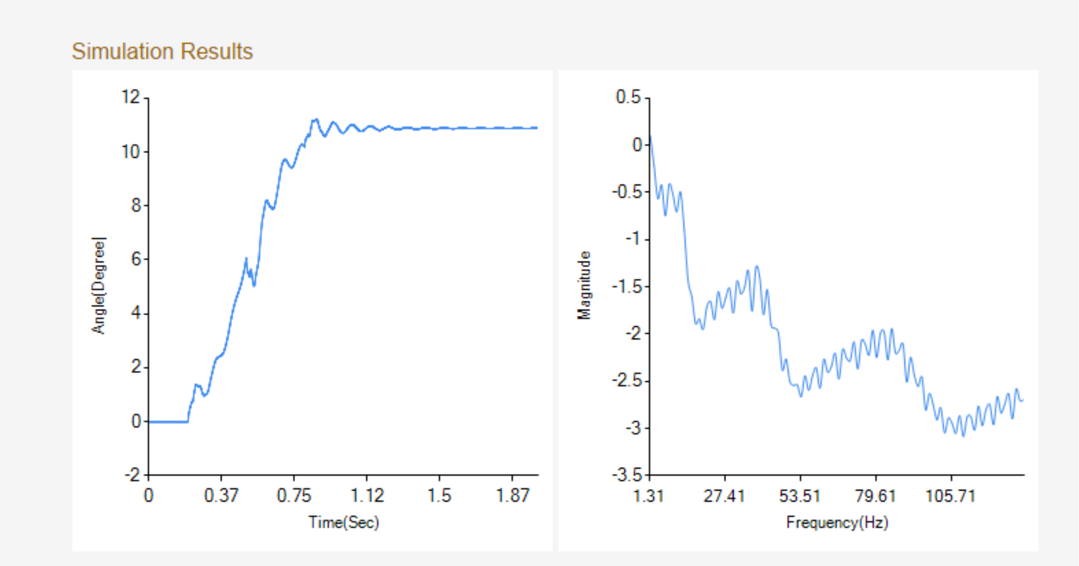


Figure 34: Hub angle with payload.

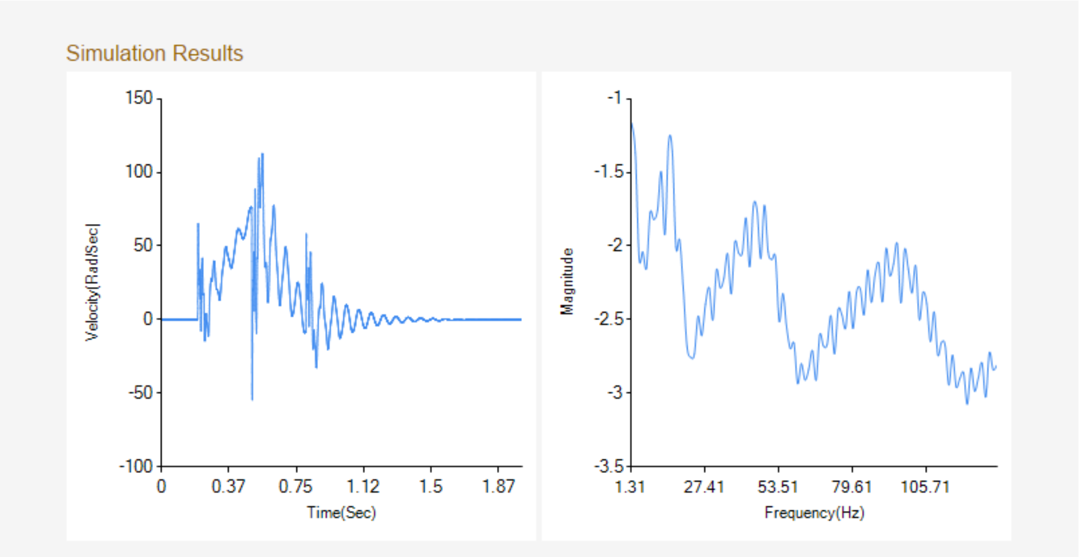


Figure 35: Hub velocity without payload.

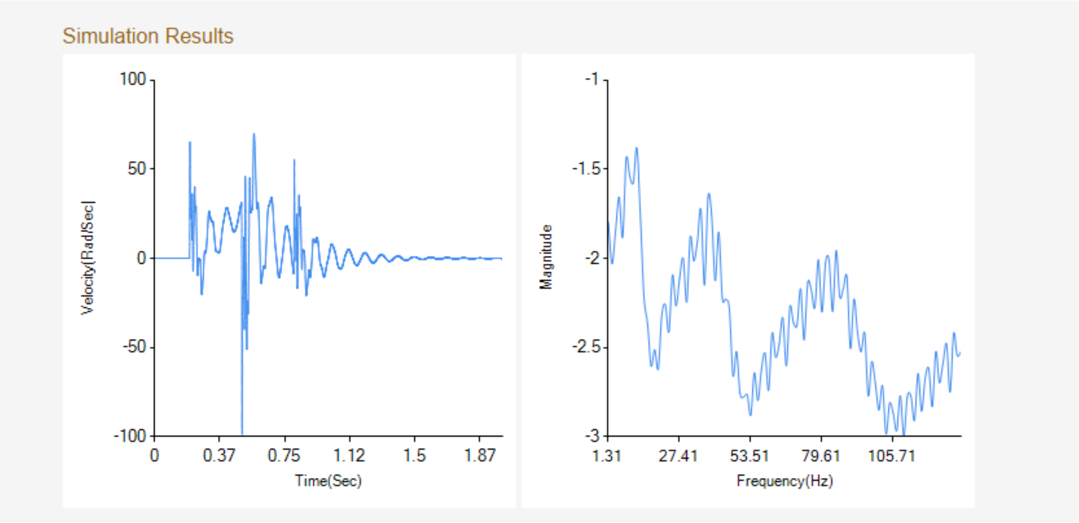


Figure 36: Hub velocity with payload.

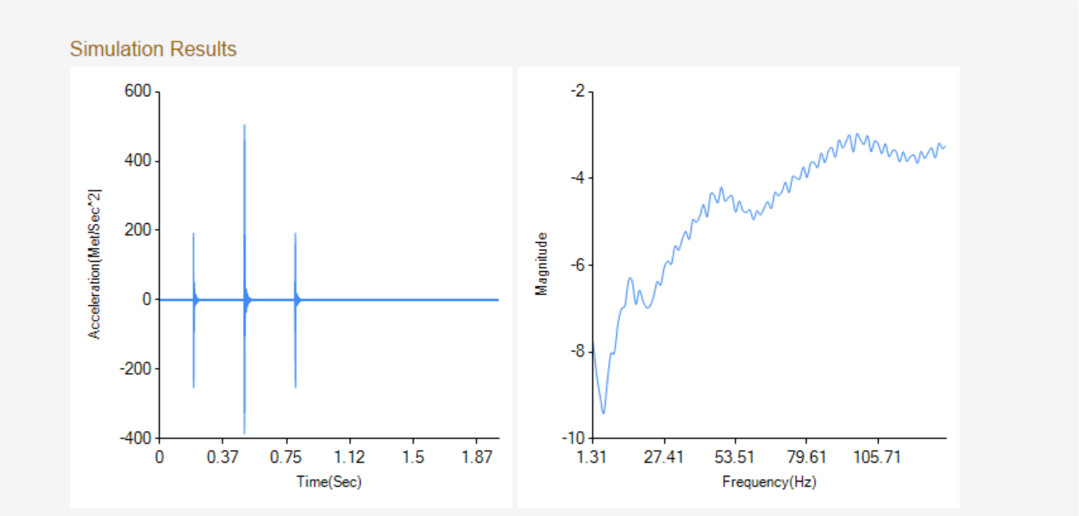


Figure 37: Hub acceleration without payload.

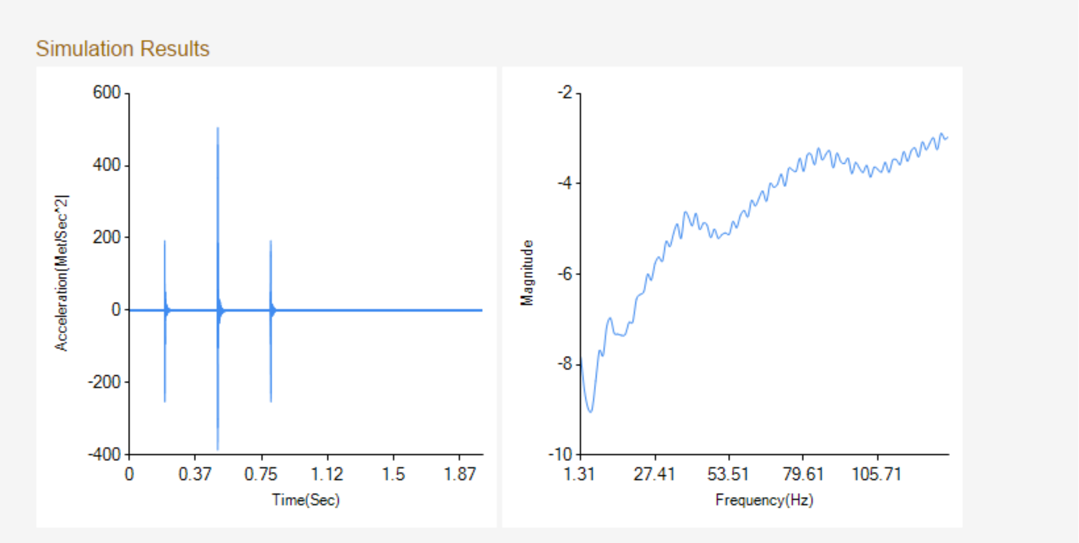


Figure 38: Hub acceleration with payload.

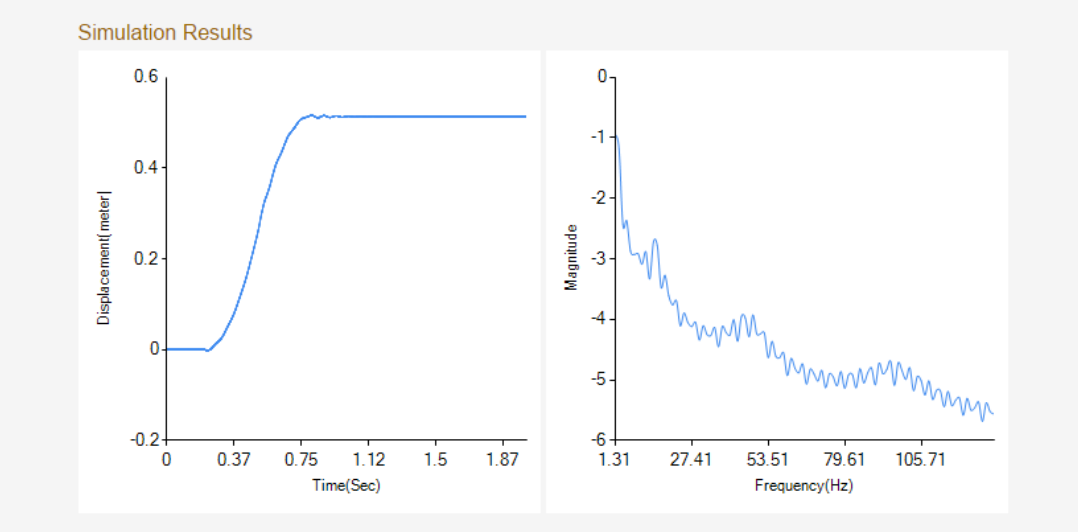


Figure 39: Endpoint displacement without payload.

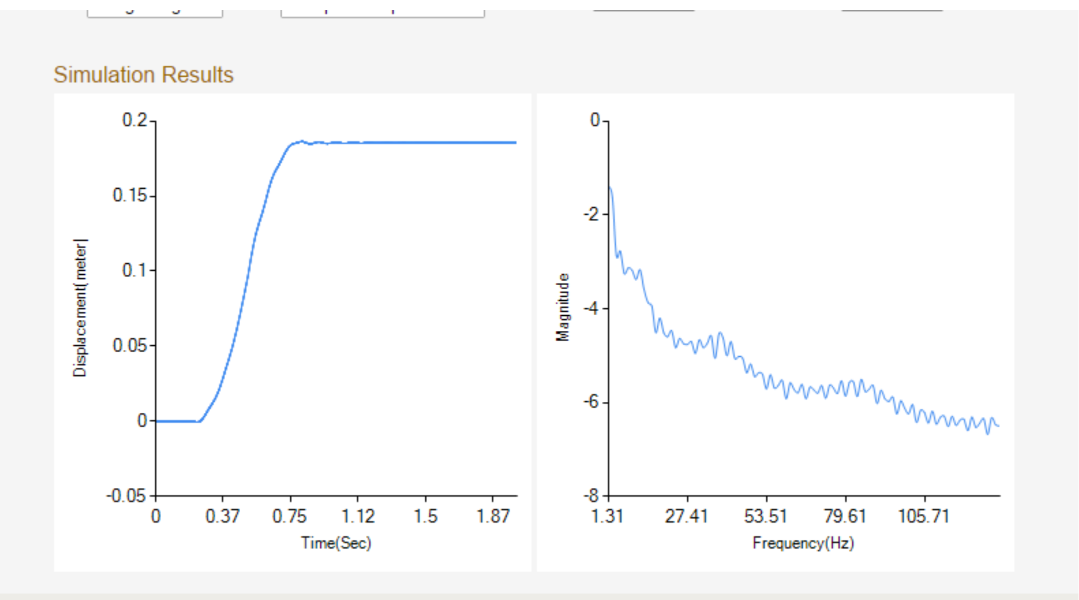


Figure 40: Endpoint displacement with payload.

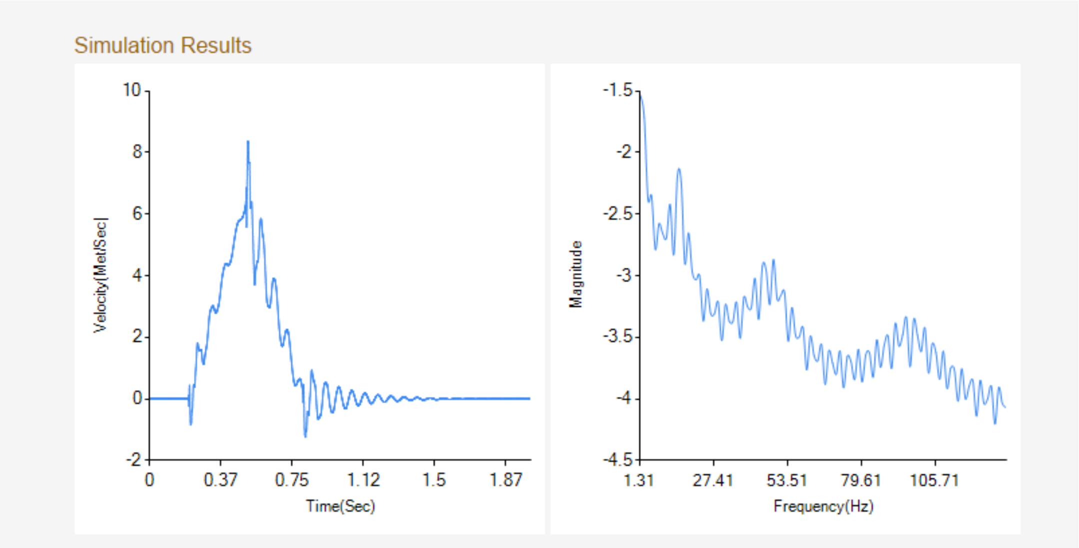


Figure 41: Endpoint velocity without payload.

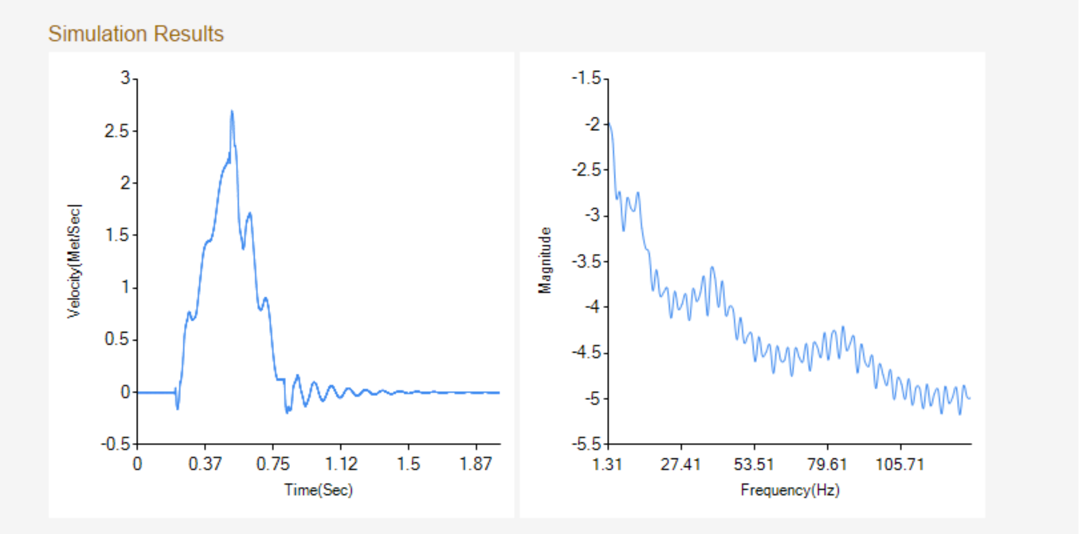


Figure 42: Endpoint velocity with payload.

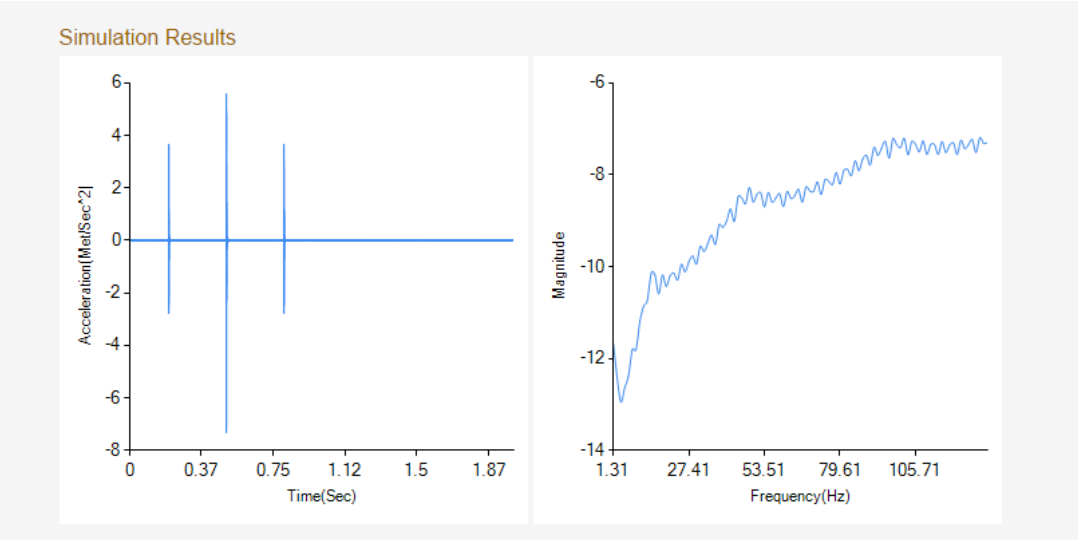


Figure 43: Endpoint acceleration without payload.

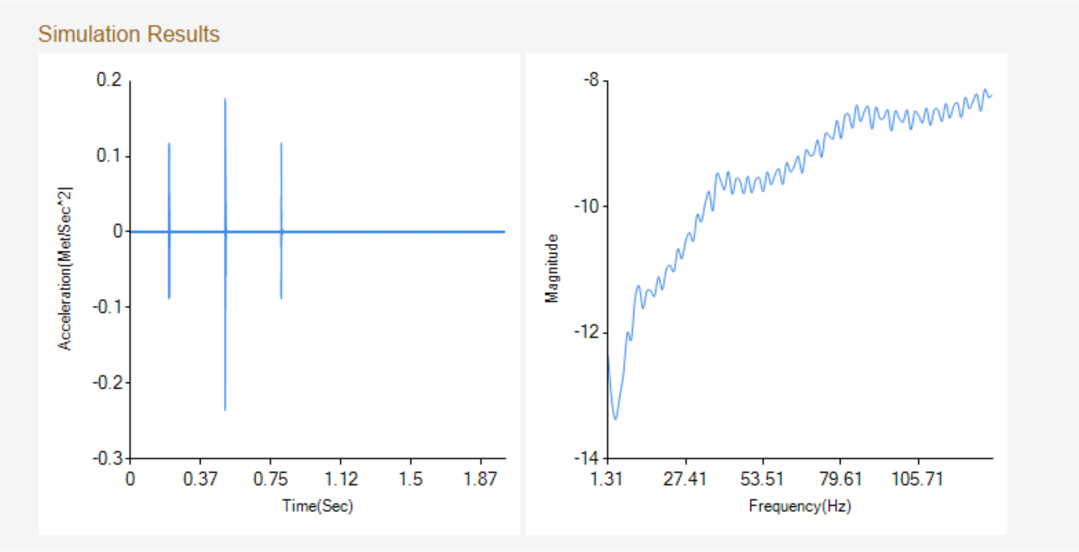


Figure 44: Endpoint acceleration with payload.

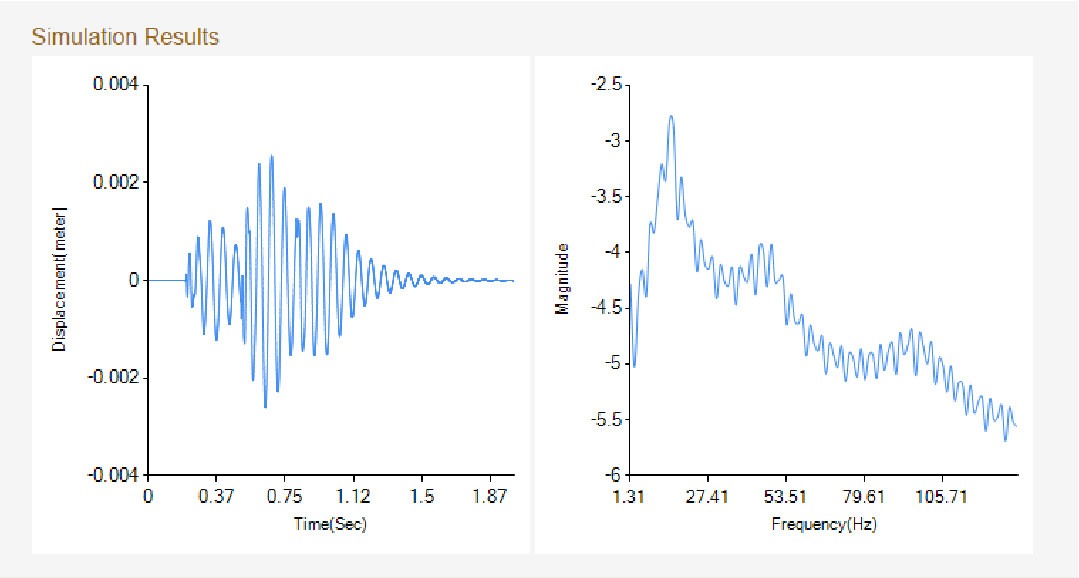
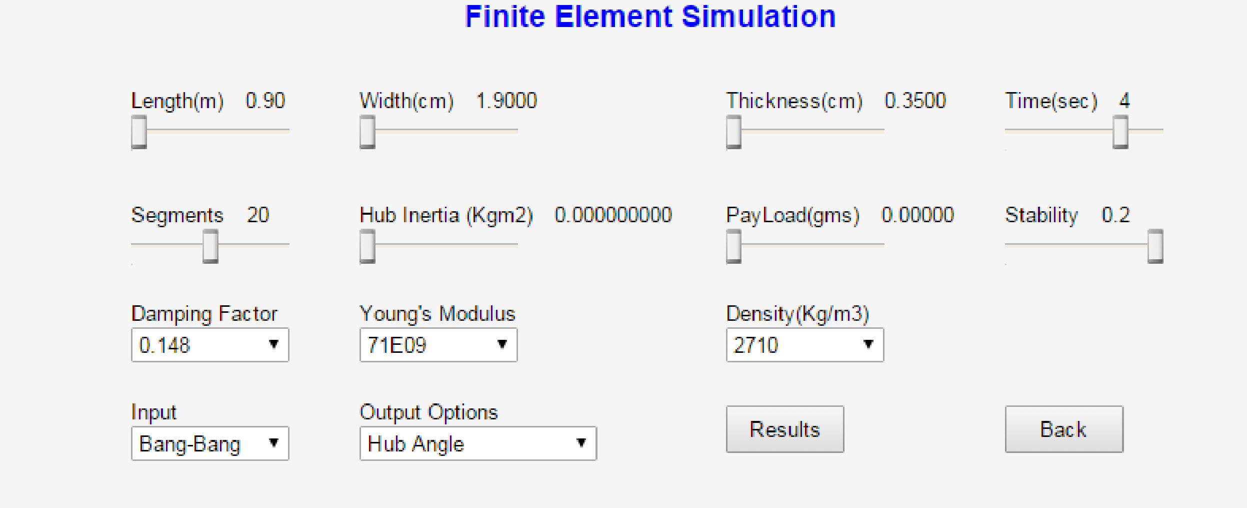


Figure 45: Residual movement.



Length(m) 0.90

Width(cm) 1.9000

Thickness(cm) 0.3500

Time(sec) 4

Segments 20

Hub Inertia (Kgm2) 0.000000000

PayLoad(gms) 0.00000

Stability 0.2

Damping Factor 0.148

Young's Modulus 71E09

Density(Kg/m3) 2710

Input Bang-Bang

Output Options Hub Angle

Figure 46: Finite element input configuration.

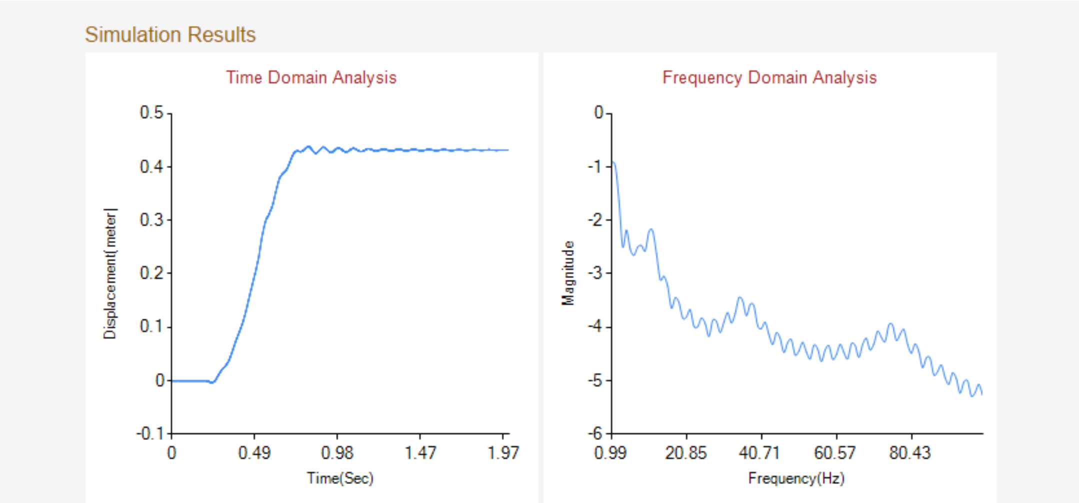


Figure 47: Finite element end point displacement.

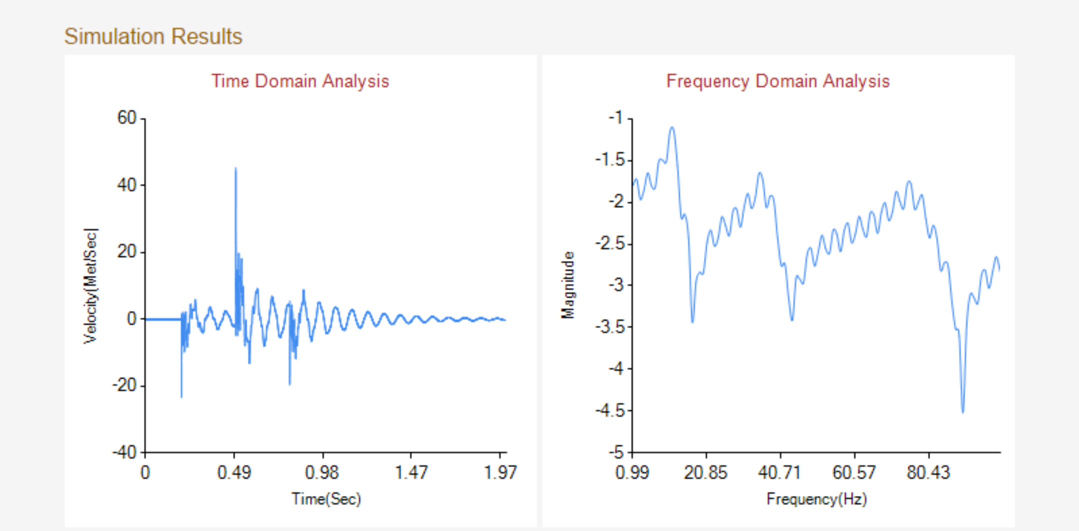


Figure 48: Finite element end point velocity.

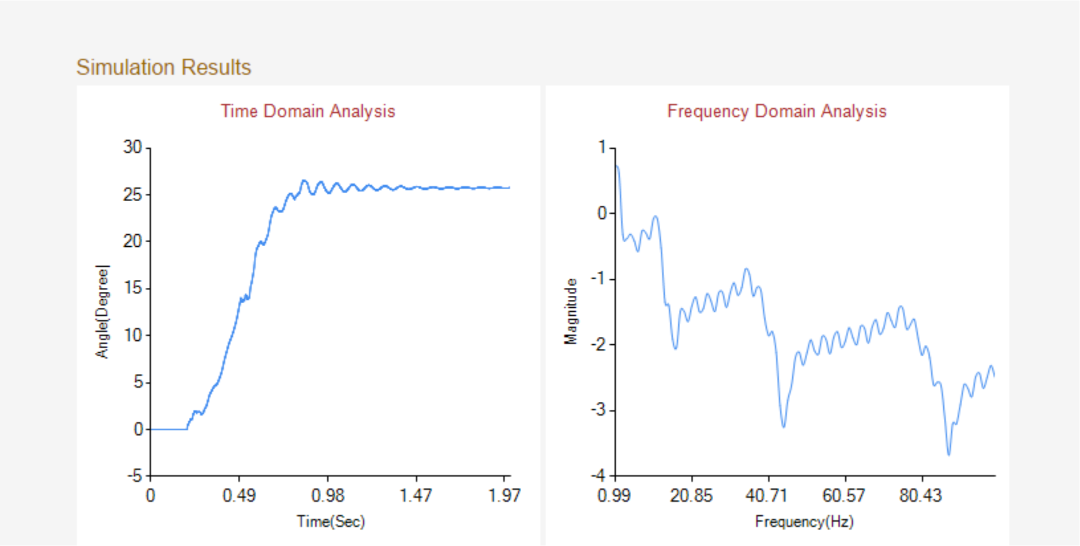


Figure 49: Finite element hub Angle.

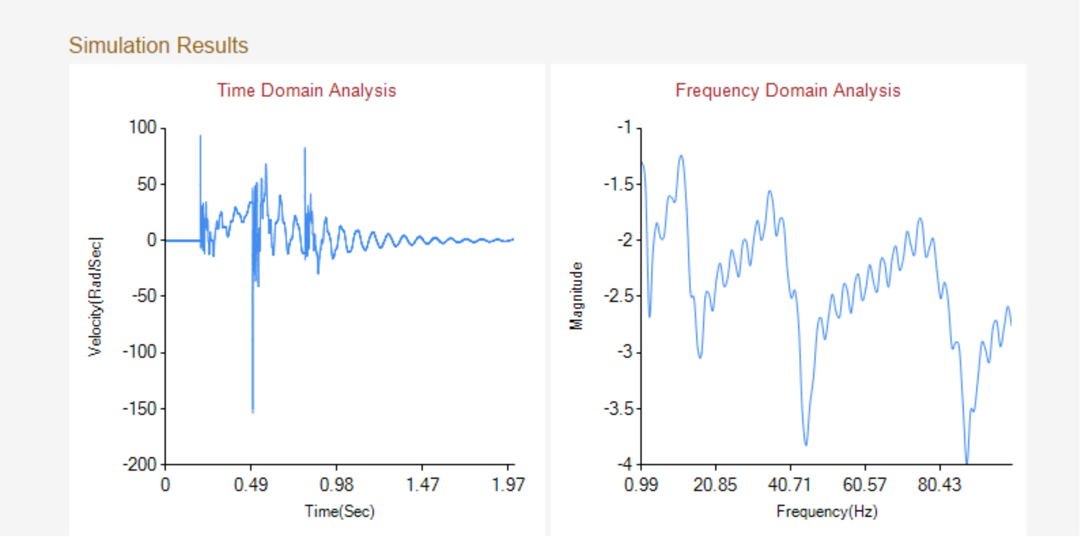
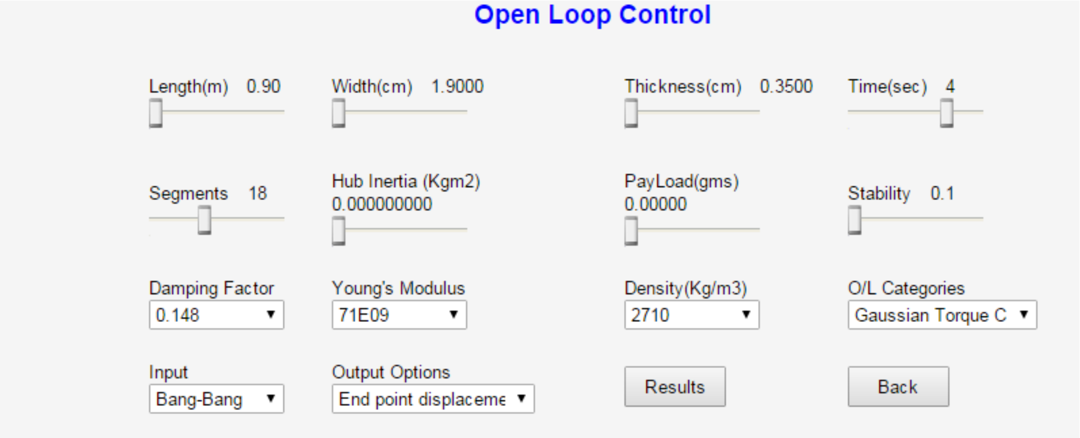


Figure 50: Finite element hub velocity.



Open Loop Control

Length(m) 0.90

Width(cm) 1.9000

Thickness(cm) 0.3500

Time(sec) 4

Segments 18

Hub Inertia (Kgm2) 0.000000000

PayLoad(gms) 0.00000

Stability 0.1

Damping Factor 0.148

Young's Modulus 71E09

Density(Kg/m3) 2710

O/L Categories Gaussian Torque C

Input Bang-Bang

Output Options End point displaceme

Figure 51: Open-loop Gaussian torque input configuration.

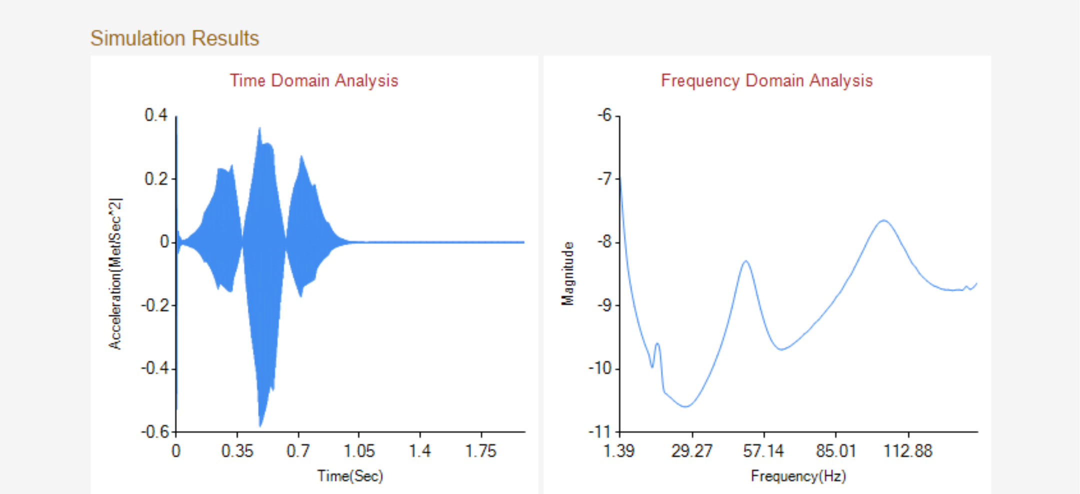


Figure 52: Open-loop Gaussian torque hub acceleration.

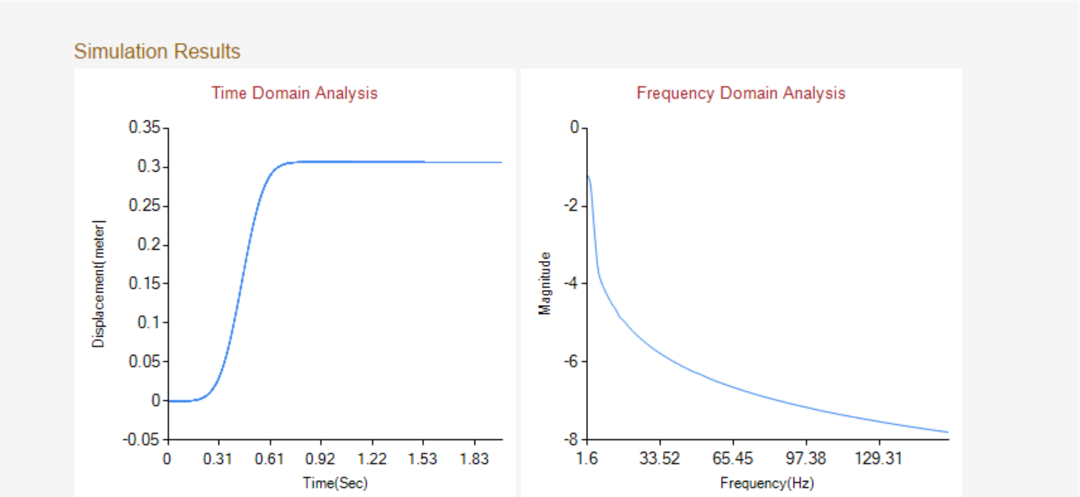
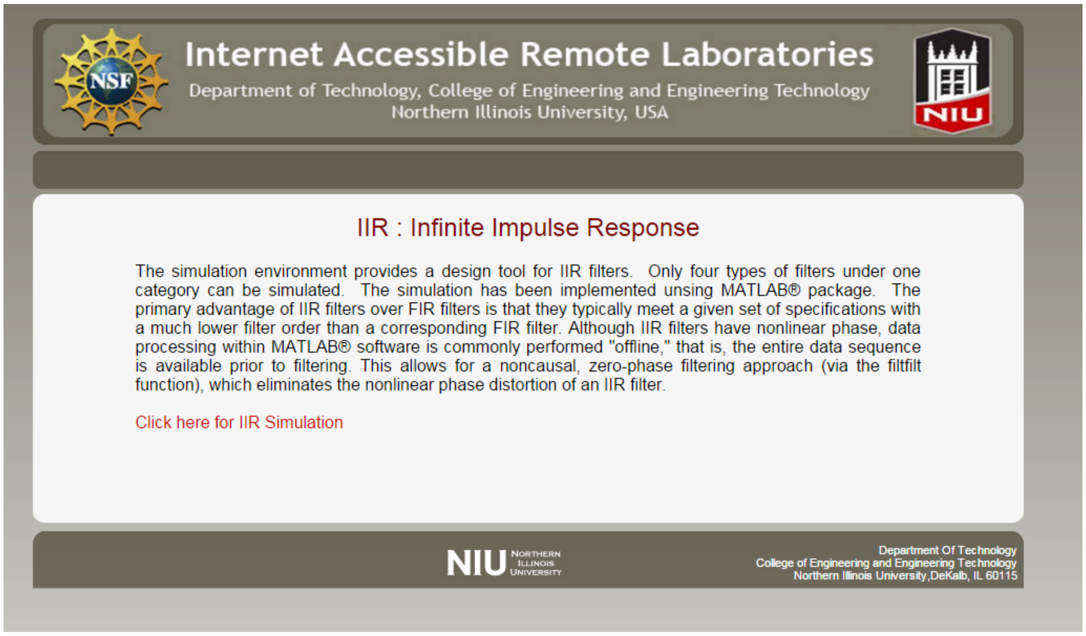


Figure 53: Open-loop Gaussian torque endpoint displacement.

4.2 IIR FILTER



Internet Accessible Remote Laboratories

Department of Technology, College of Engineering and Engineering Technology
Northern Illinois University, USA

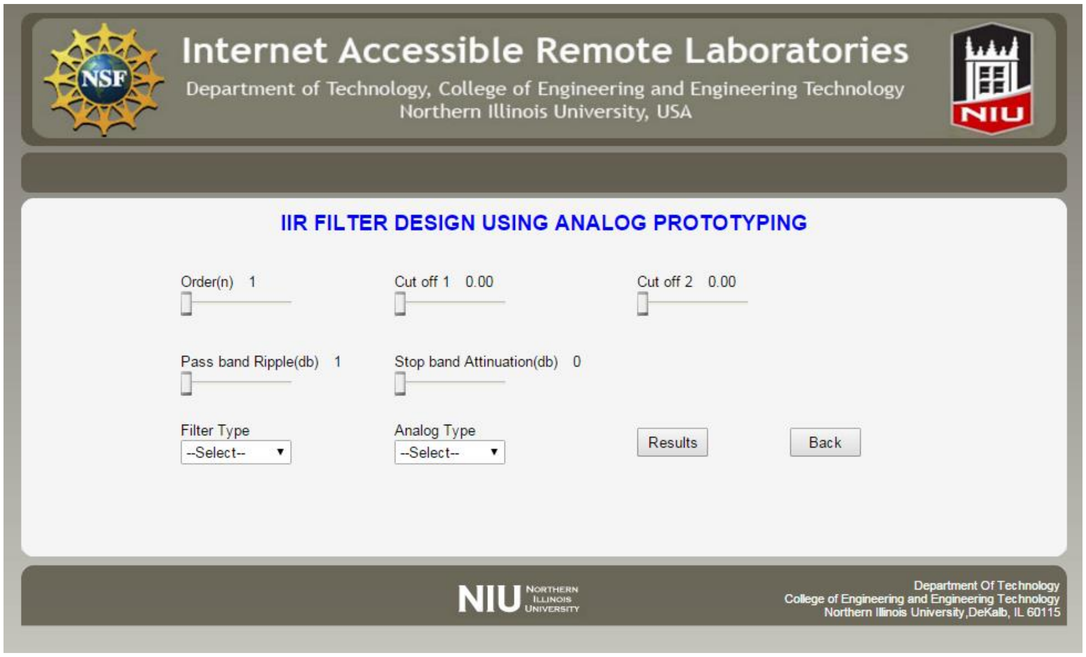
IIR : Infinite Impulse Response

The simulation environment provides a design tool for IIR filters. Only four types of filters under one category can be simulated. The simulation has been implemented unsing MATLAB® package. The primary advantage of IIR filters over FIR filters is that they typically meet a given set of specifications with a much lower filter order than a corresponding FIR filter. Although IIR filters have nonlinear phase, data processing within MATLAB® software is commonly performed "offline," that is, the entire data sequence is available prior to filtering. This allows for a noncausal, zero-phase filtering approach (via the filtfilt function), which eliminates the nonlinear phase distortion of an IIR filter.

Click here for IIR Simulation

Department Of Technology
College of Engineering and Engineering Technology
Northern Illinois University,DeKalb, IL 60115

Figure 54: IIR filter home page.



Internet Accessible Remote Laboratories

Department of Technology, College of Engineering and Engineering Technology
Northern Illinois University, USA

IIR FILTER DESIGN USING ANALOG PROTOTYPING

Order(n) 1

Cut off 1 0.00

Cut off 2 0.00

Pass band Ripple(db) 1

Stop band Attinuation(db) 0

Filter Type --Select--

Analog Type --Select--

Department Of Technology
College of Engineering and Engineering Technology
Northern Illinois University,DeKalb, IL 60115

Figure 55: IIR filter input configuration page.

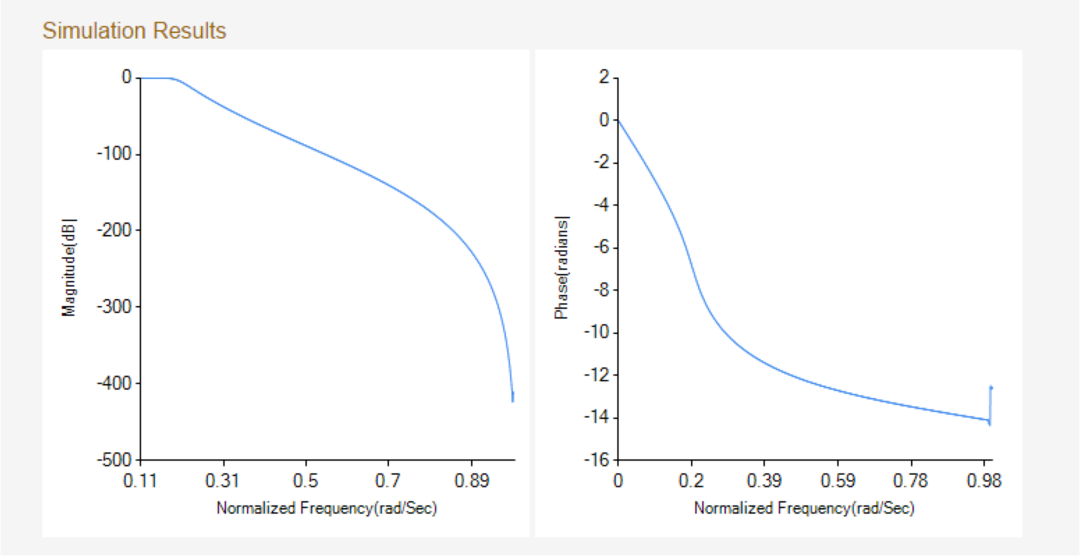


Figure 56: IIR filter Butterworth low pass filter.

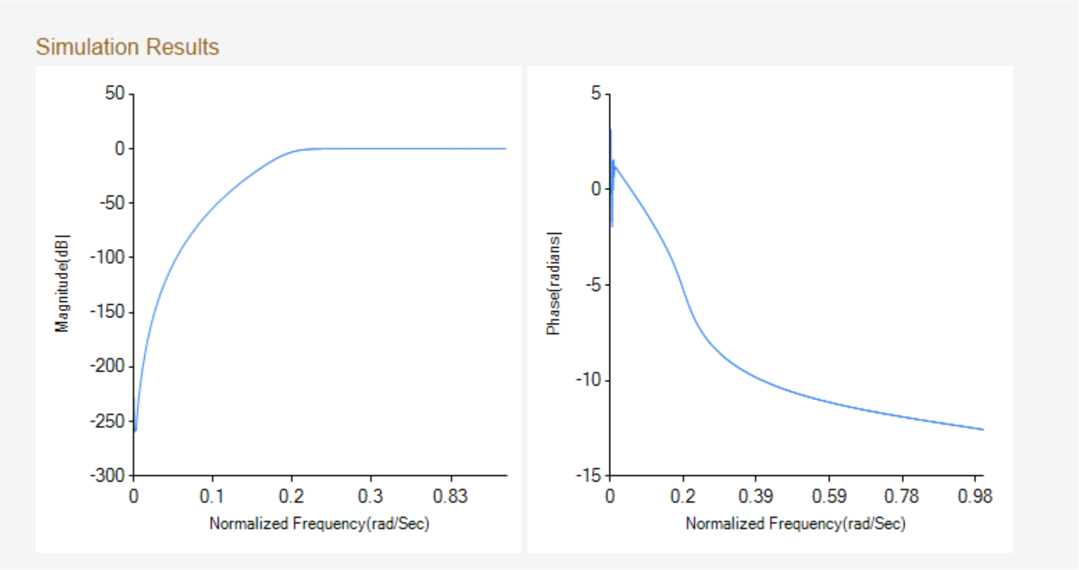


Figure 57: IIR filter Butterworth high pass filter.

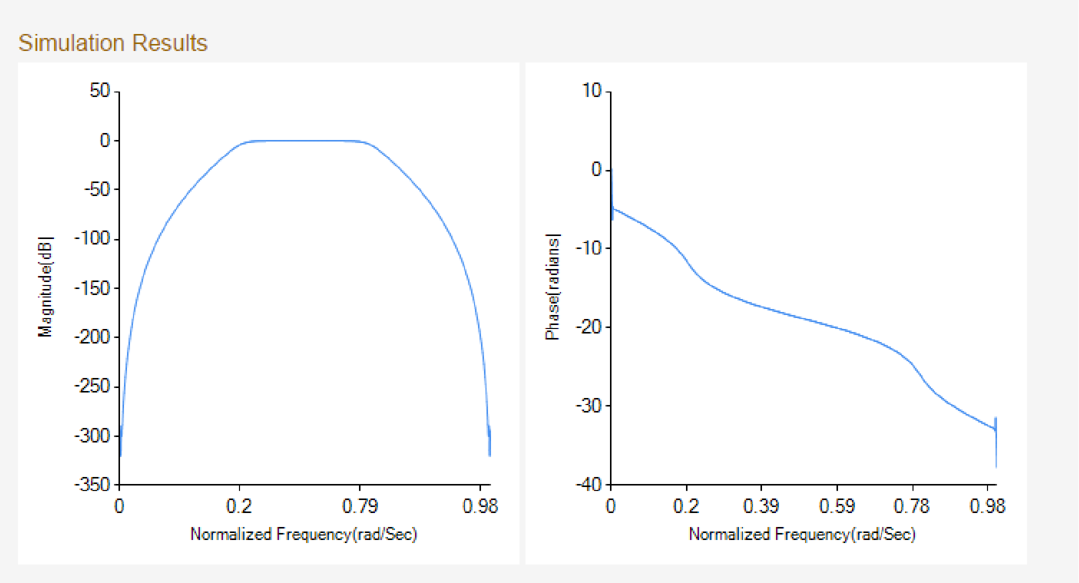


Figure 58: IIR filter Butterworth band pass filter.

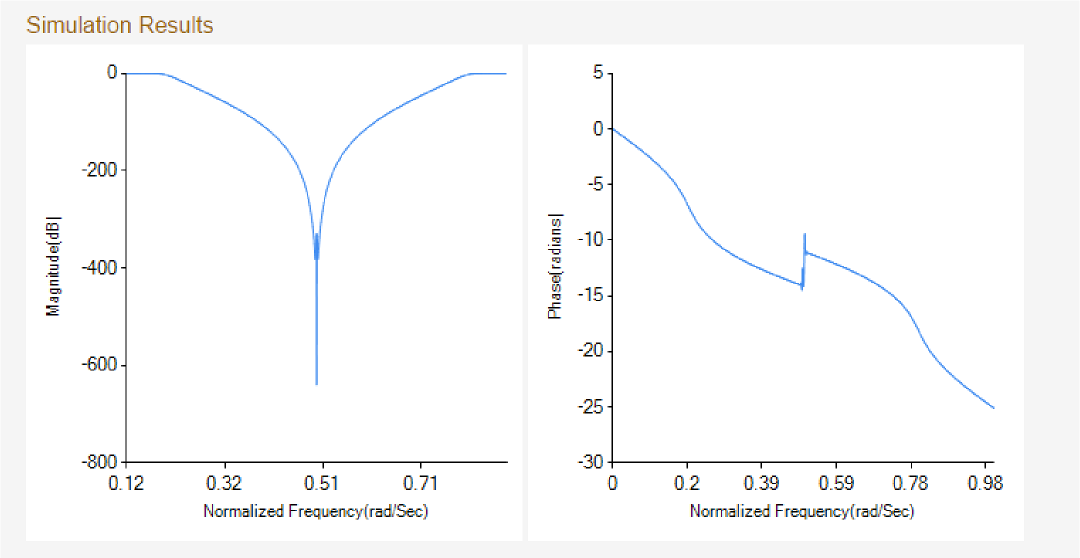


Figure 59: IIR filter Butterworth band stop filter.

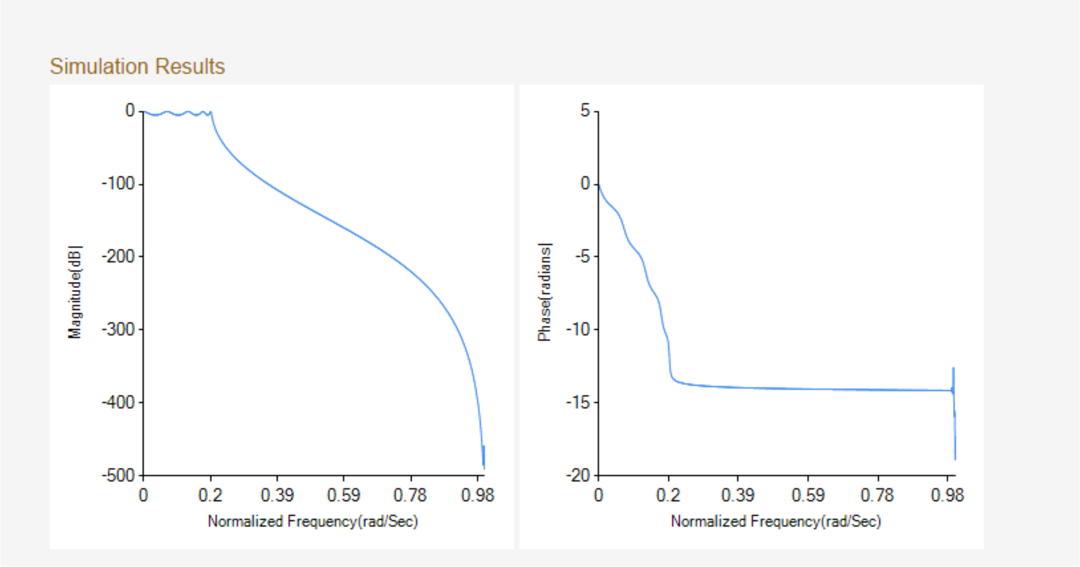


Figure 60: IIR filter Chebyshev’s Type I low pass filter.

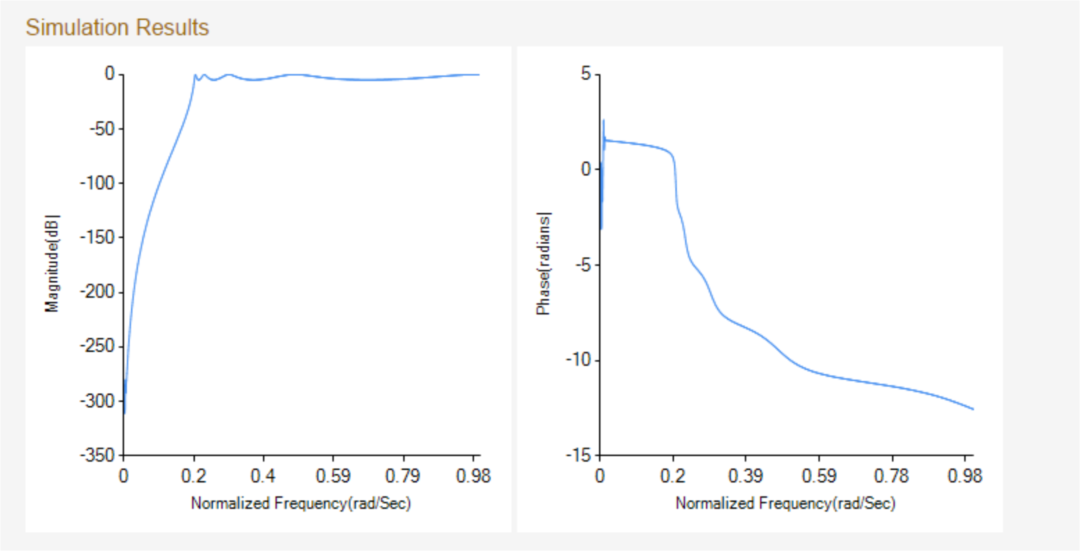


Figure 61: IIR filter Chebyshev’s Type I high pass filter.

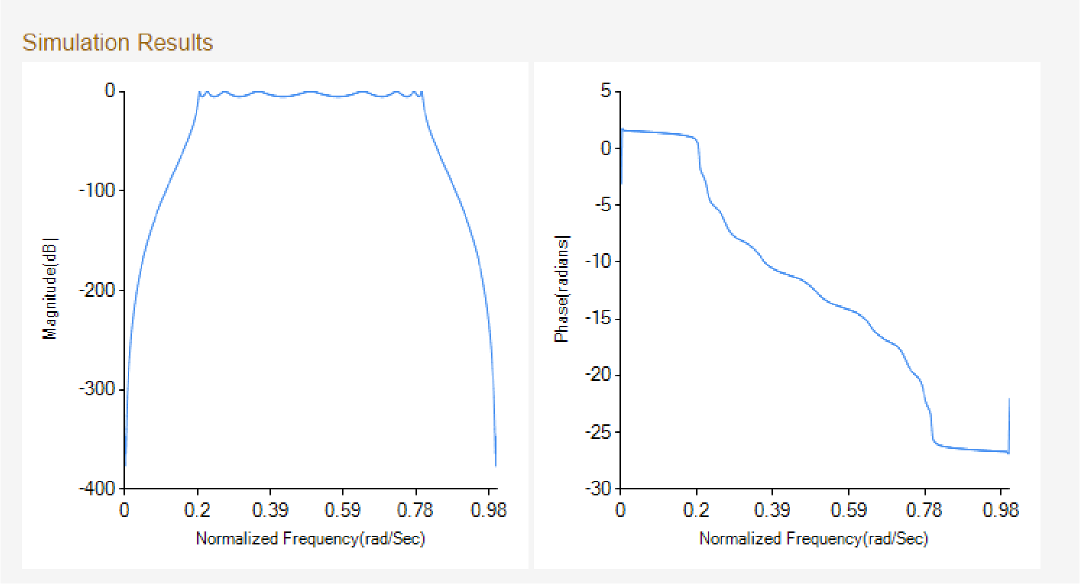


Figure 62: IIR filter Chebyshev’s Type I band pass filter.

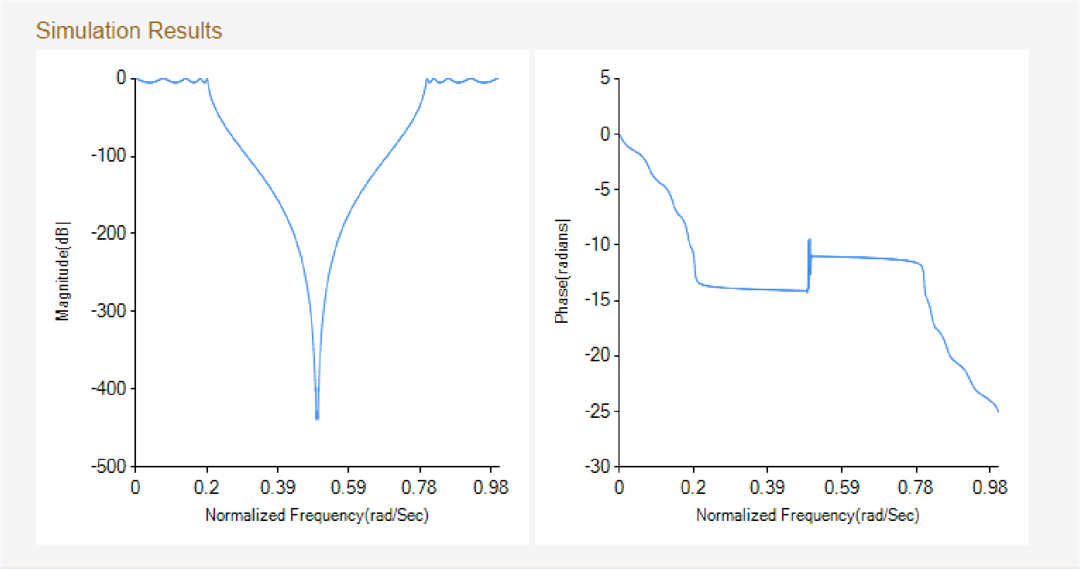


Figure 63: IIR filter Chebyshev’s Type I band stop filter.

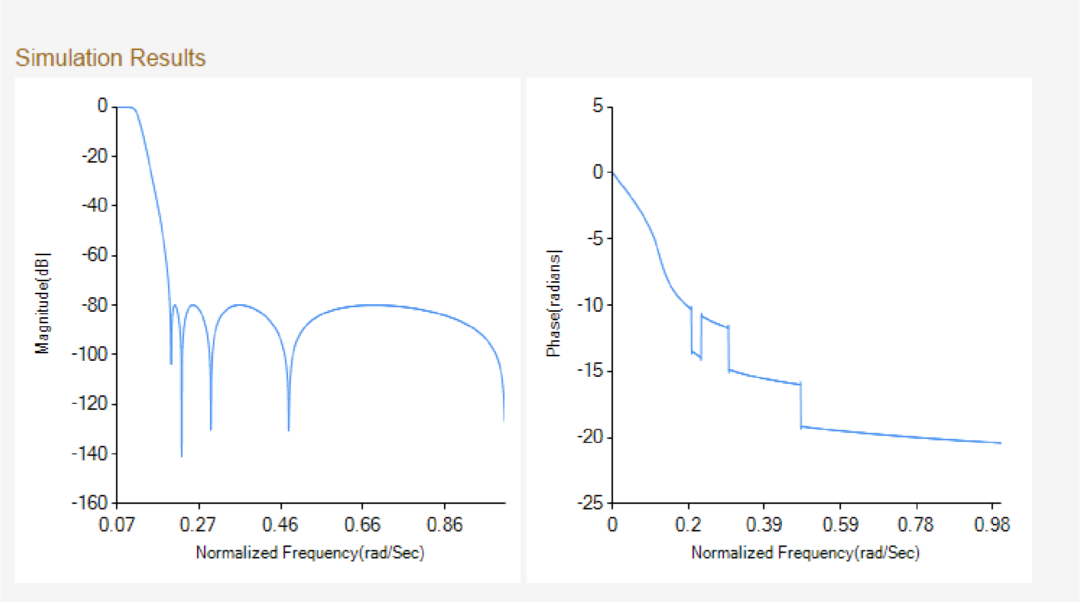


Figure 64: IIR filter Chebyshev’s Type II low pass filter.

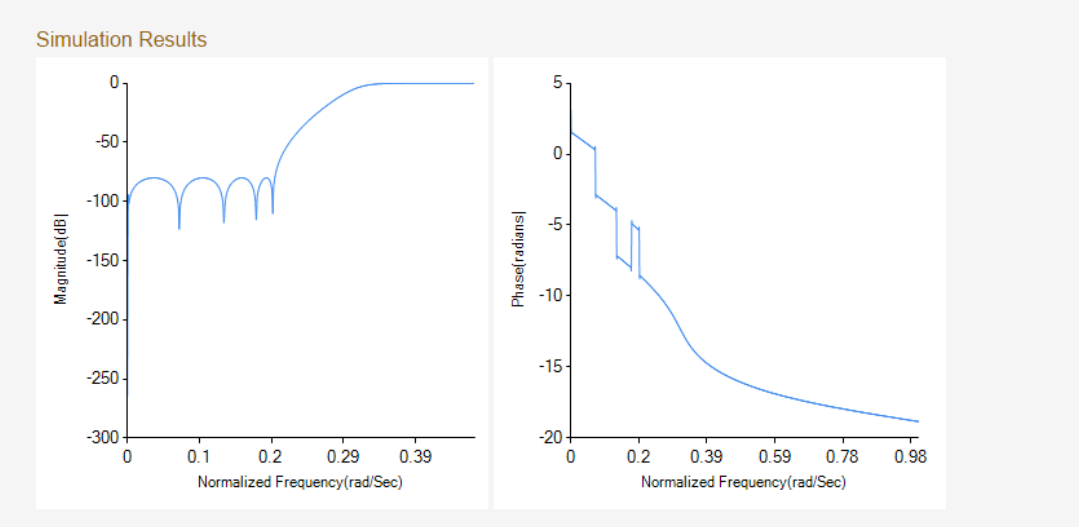


Figure 65: IIR filter Chebyshev’s Type II high pass filter.

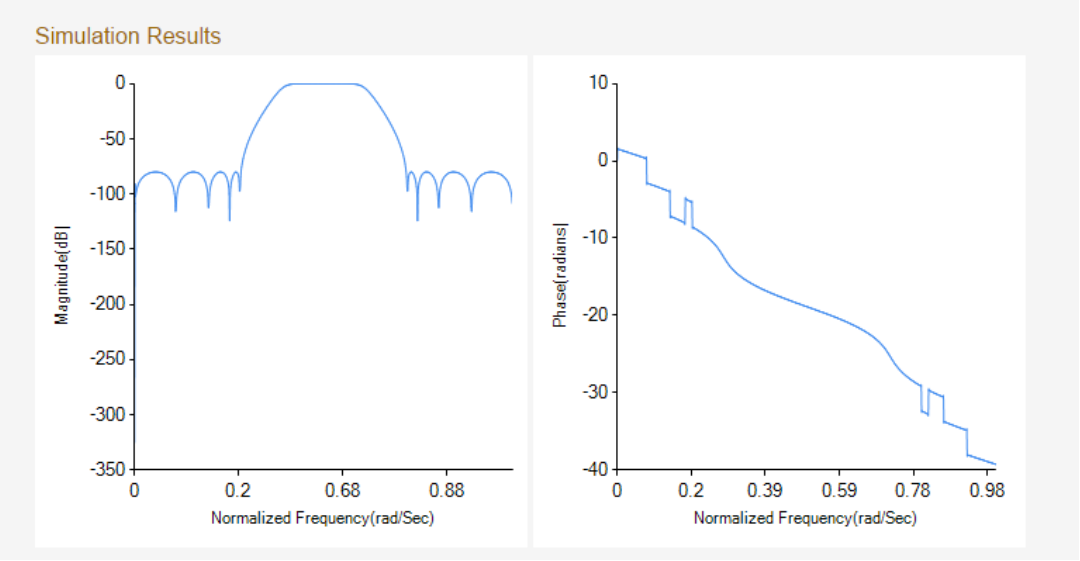


Figure 66: IIR filter Chebyshev’s Type II band pass filter.

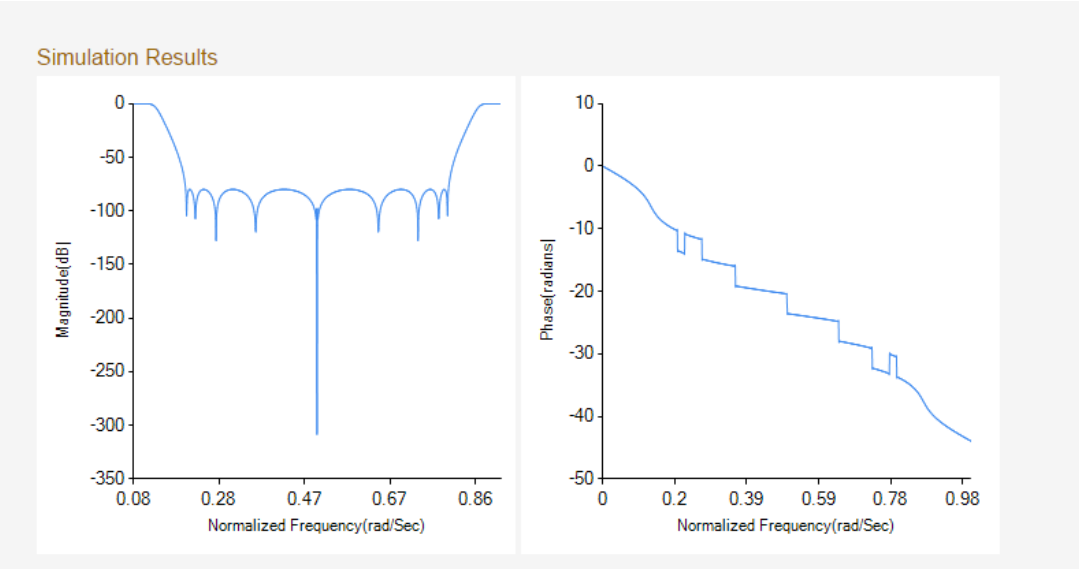


Figure 67: IIR filter Chebyshev’s Type II band stop filter.

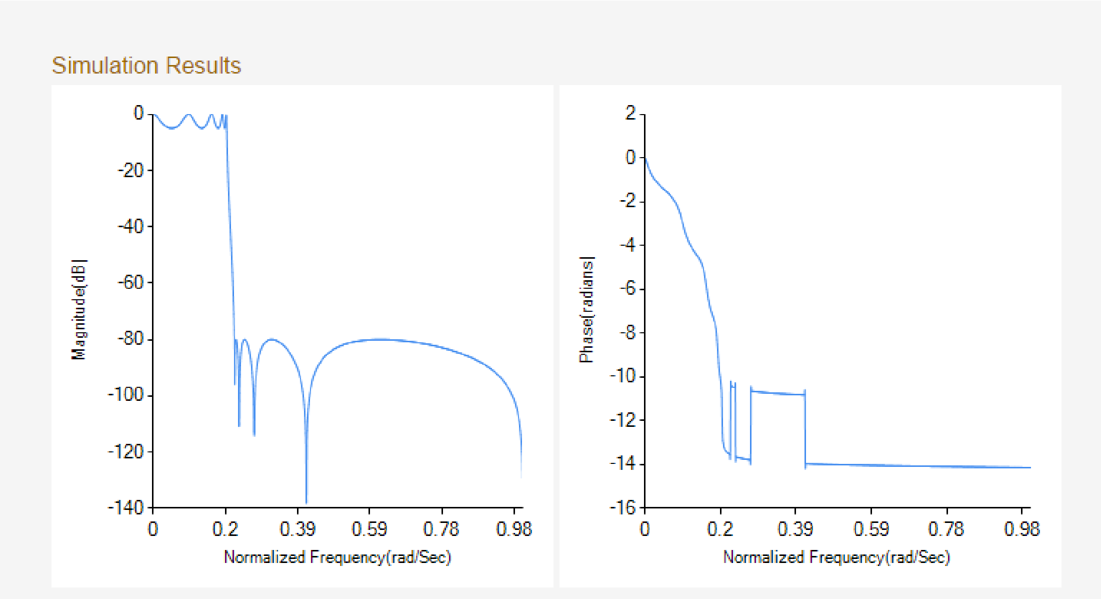


Figure 68: IIR filter elliptical low pass filter.

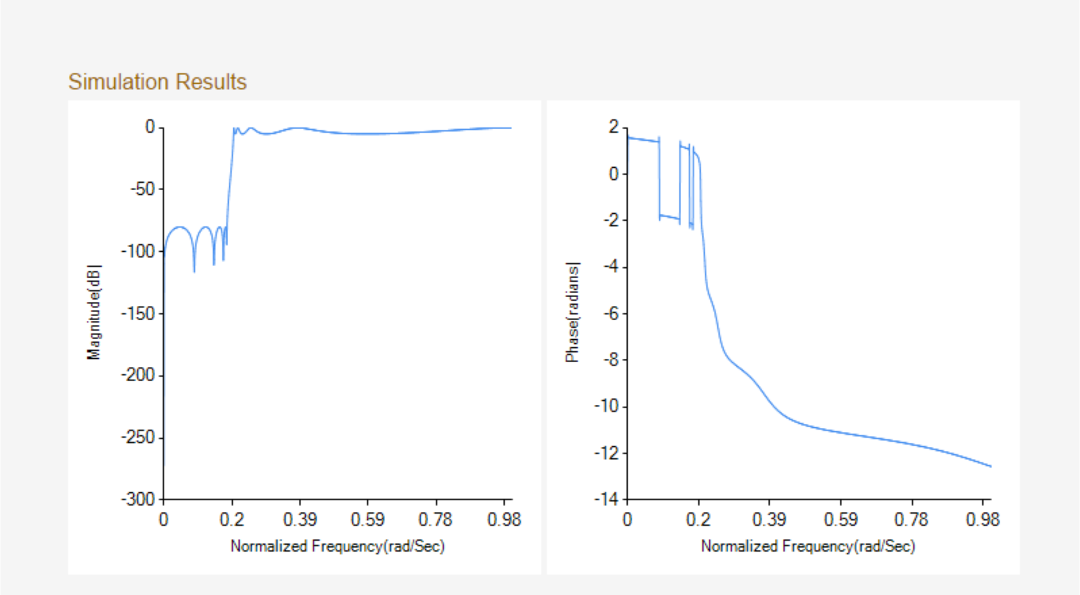


Figure 69: IIR filter elliptical high pass filter.

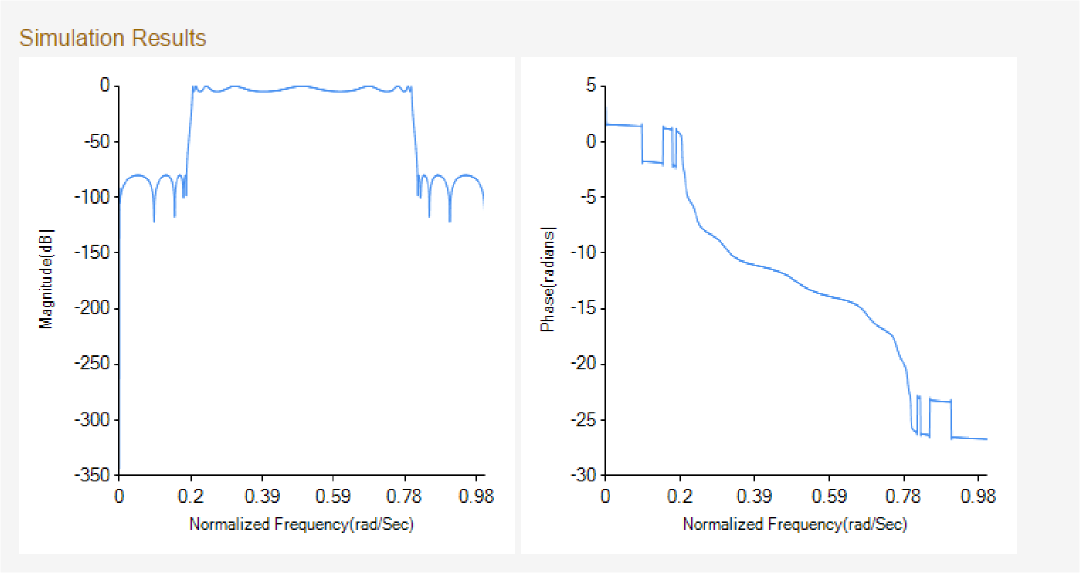


Figure 70: IIR filter elliptical band pass filter.

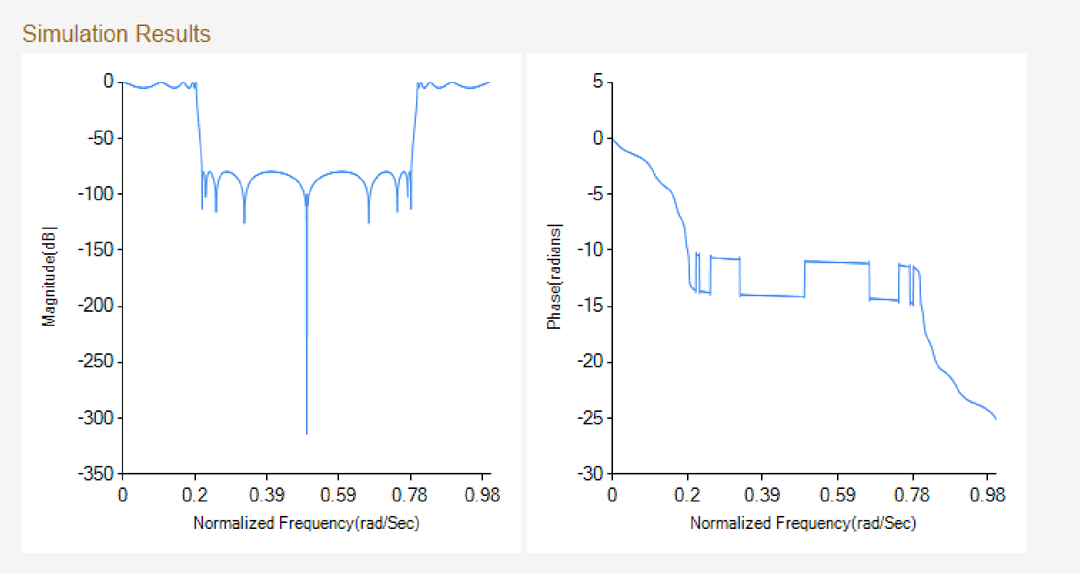


Figure 71: IIR filter elliptical band stop filter.

5 Conclusion:

The algorithms and filter implementation is a prototype of remote simulation for educational and research purposes. By using this approach on filter implementation and Flexible Manipulator implementation errors can be rectified without any knowledge of MATLAB and efficiency can be improved drastically.

The project converts the data between C# and MATLAB arrays, dissolving the GUI and writing of MATLAB code in the right order in such a way that right outputs could be captured in .Net Assembly. The data captured at the .Net assembly is subjected to various refinement processes so that, it is suitable for plotting the data over the webpage.

Any algorithm or program written in MATLAB can be easily deployed over the web page. The Finite Difference, Finite Element and Open Loop algorithms of Flexible Manipulator and Classical IIR Filters are deployed over the webpage using Compiler SDK of MATLAB and Microsoft Visual Studio. The input data from the client is passed to the various algorithms to compute the results using text files.

REFERENCES:

1. Mathamatica from Wikipedia, the free encyclopedia (Viewed on 09/02/15) Retrieved from: https://en.wikipedia.org/wiki/Mathematica
2. Web from Webmathematica (Viewed on 10/28/15)Retrieved from: http://www.wolfram.com/products/webmathematica/whatis.html
3. MATLAB Compiler SDK from MATLAB programs (Viewed on 08/01/15)Retrieved from: http://www.mathworks.com/products/matlab-compiler-sdk/features.html#key-features
4. LabVIEW Application Server from LabVIEW (Viewed on 11/28/15)Retrieved from: http://zone.ni.com/reference/en-XX/help/371361M-01/lvhowto/build_web_service/
5. MATLAB Web Server manual from MATLAB(Rohan Academy) (Viewed on 11/28/15)Retrieved from: http://www-rohan.sdsu.edu/doc/matlab/pdf_doc/webserver/webserver.pdf
6. LabVIEW from Wikipedia, the free encyclopedia (Viewed on 09/02/15) Retrieved from: http://en.wikipedia.org/wiki/LabVIEW
7. Compiler SDK MATLAB .Net Assembly ((Viewed on 08/01/15)Retrieved from: http://www.mathworks.com/products/matlab-compiler-sdk/
8. Filter(Signal Processing) on Wikipedia (Viewed on 11/28/15)Retrieved from: https://en.wikipedia.org/wiki/Filter_(signal_processing)
9. FIR Filters (Signal Processing) from Wikipedia (Viewed on 11/28/15)Retrieved from: https://en.wikipedia.org/wiki/Filter_(signal_processing)
10. IIR Filters (Signal Processing) from Wikipedia (Viewed on 11/28/15)Retrieved from: https://en.wikipedia.org/wiki/Filter_(signal_processing)
11. IIR Filter design from Mathworks (MATLAB) (Viewed on 11/28/15)Retrieved from: http://www.mathworks.com/help/signal/ug/iir-filter-design.html

Appendix: Code

MATLAB:

I. Main function:

```
function new_result1=fd2(L1,T1,W1,Ih1,Mp1,time1,n1,lam1,Ds1,E1,row1,InpOpt1,sel_point,sel_output)
% Predefined values
%n=20;
n=n1;
%Ds=0;
Ds=Ds1;

   %L = 0.96;          % in meter length of the manipulator
   L=L1;
   %W = 0.019230;       % in meter width of the beam
   W=W1;
   %T = 0.0032004;      % in meter thickness of the beam
   T=T1;
   %E = 7.11e10;        % young's modulus of the beam material
   E=E1;
   I =  (W*(T*T*T))/12; % second moment of inertia
   EI = E*I;           % in N-m^2 flexural rigidity of the beam
   area = W*T;         % in m^2 area of cross-section
   %row = 2710;         % in kg/m^3 density of the manipulator
   row=row1;
   m = area*row;       % in kg mass per unit length
   Ib = (1/3)*m*(L^2);  % in kg-m^2 inertia of the beam only
   %time = 4.0;
   time=time1;
   %lam  = 0.2;
lam=lam1;
   %Ih=0;
Ih=Ih1;
   %Mp=0;
Mp=Mp1;

%InpOpt=3;
InpOpt=InpOpt1;
disp(sel_point);

%

% CALCULATION FOR SAMPLING TIME
h = L/n;
dt = sqrt((lam*m*L^4)/((n^4)*EI));
%dt = 0.000217904;
ite = round(time/dt);     % total number of iteration
%
```

```
%
% Calculation common variables from given constants
a = 2-((6*EI)*dt^2)/(m*h^4);
b = 4*(EI*dt^2)/(m*h^4);
c = (EI*dt^2)/(m*h^4);
d = Ds*dt/(m*h^2);
t=0:dt:(ite-1)*dt;

%
% Calculation of system parameters
K1 = ((c*EI*dt^2) + (2*c*h*Ih) + (a-2*d)*EI*dt^2)/((EI*dt^2) + (c*h*Ih));
K2 = ((b+d)*EI*dt^2) / ((EI*dt^2) + (c*h*Ih));
K3 = -(c*EI*dt^2) / ((EI*dt^2) + (c*h*Ih));
K4 = -((c*h*Ih) + (1-2*d)*EI*dt^2) / ((EI*dt^2) + (c*h*Ih));
K5 = -(d*EI*dt^2) / ((EI*dt^2) + (c*h*Ih));
K6 = (c*(dt^2)*h^2)/ ((EI*dt^2) + (c*h*Ih));

K7 = -c;
K8 = (b+d);
K9 = (c+a-2*d);
K10= -(c*2-b-d);
K11= -d;
K12= -(1-2*d);
K13= -d;

K14 = (-2*c) * EI*(dt^2) / ((dt^2)*EI + 2*c*(h^3)*Mp);
K15 = (4*c * EI*(dt^2) / ((dt^2)*EI + 2*c*(h^3)*Mp));
K16 = (a+2*b-4*c+(4*c*(h^3)*Mp/((dt^2)*EI)))  * (EI*(dt^2) / ((dt^2)*EI + 2*c*(h^3)*Mp));
K17 = -(2*c*(h^3)*Mp/((dt^2)*EI) + 1)  * (EI*(dt^2) / ((dt^2)*EI + 2*c*(h^3)*Mp));

% filling up matricies

for i=1:n
    for j=1:n
      if i==j
        A(i,j)=(a-2*d);
        B(i,j)=(2*d-1);
      elseif abs(i-j)==1
        A(i,j)=b+d;
        B(i,j)=-d;
      elseif abs(i-j)==2
        A(i,j)=-c;
        B(i,j)=0;
      end
    end
end

A(1,1)=K1; A(1,2)=K2; A(1,3)=K3;
A(n-1,n-3)=K7; A(n-1,n-2)=K8; A(n-1,n-1)=K9; A(n-1,n)=K10;
A(n,n-2)=K14; A(n,n-1)=K15; A(n,n)=K16;
B(1,1)=K4; B(1,2)=K5;
B(n-1,n-2)=K11; B(n-1,n-1)=K12; B(n-1,n)=K13;
```

```
B(n,n-1)=0; B(n,n)=K17;
C=K6*eye(n,1);




% START SIMULATION WITH STATE-SPACE OPERATION

P = [A B ; eye(n) zeros(n)];
Q = [C;zeros(n,1)];
R = eye(2*n);
S = zeros(2*n,1);

if InpOpt == 1
    timevect = linspace(0,time,ite);

    %RandInput = [timevect; (rand(ite,1)-mean(rand(ite,1)))*0.2]
    RandInput = [timevect; (rand(1,ite)-mean(rand(1,ite)))*0.2];
   % RandInput = RandInput(:);

elseif InpOpt == 2
    timevect = linspace(0,time,ite);

    load cmp_prbs.dat;
    cmp_data = cmp_prbs(:,1)';
    RepLen = length(timevect)/length(cmp_data);
    NewCmpData = repmat(cmp_data,1,ceil(RepLen));
    NewCmpData = NewCmpData(1:length(timevect));
    RandInput = [timevect;NewCmpData./2];
elseif InpOpt == 3
    timevect = linspace(0,time,ite);

%     t1 = 0.2;
%     t2 = 0.3;
%     t3 = 0.3;
%     block1 = t1/dt;
%     block2 = (t2)/dt;
%     block3 = (t3)/dt;
%     blk1 = zeros(1,block1);
%     blk2 = 0.3*ones(1,block2-1);
%     blk3 = -0.3*ones(1,block3-1);
%     blk4 = zeros(1,ite-block3);
%     inp_val = [blk1,blk2,blk3,blk4];


    inp_val = [zeros(1,length(0:dt:0.2)),0.3*ones(1,length(0.2:dt:0.5)),-
0.3*ones(1,length(0.5:dt:0.8)),zeros(1,length(0.8:dt:time))];
    RandInput = [timevect;inp_val(1:length(timevect))];
```

```
end
RandInput=RandInput';




% initial conditions
x0 = zeros(1,2*n);


t0=cputime;

% simulation
[y,x]=dlsim(P,Q,R,S,RandInput(:,2));
runtime = cputime-t0

gg = length(x);
% calculate the end point displacements

for i=1:gg
  net(i) = y(i,n);
end

yn = net;
end



if sel_output==1
    new_result1=ite;
end

if sel_output==2
    new_result1=t;
end

if sel_output==3
    new_result1=yin;
end

if sel_output==4
    new_result1=aa;
end

if sel_output==5
    new_result1=f;
end
```

```
if sel_output==6
    Pn1=log10(Pn1);
    new_result1=Pn1;
end
```

II. *Function 2:*

```
if sel_point==1
 p=1;
for i=1:gg
  net(i) = y(i,p);
end
t=0:dt:(ite-1)*dt;
yn = net;
```

```
for k=1:length(yn)
if yn(k)>0
    I2(k)=1;
end
if yn(k)<0
    I2(k)=-1;
end
if yn(k)==0
    I2(k)=0;
end
end
threshold1=0;
for k2=1:length(I2)
if I2(k2)>=threshold1
I8(k2)=yn(k2);
else
I8(k2)=0;
end
end
ref=-1;
for k3=1:length(I2)
if I2(k3)==ref
I4(k3)=1;
else
    I4(k3)=0;
end
end
```

```
threshold2=1;
for k4=1:length(yn)
if I4(k4)>=threshold2
I5(k4)=yn(k4);
else
I5(k4)=0;
end
end

for k5=1:length(I5)
I6(k5)=abs(I5(k5));
end

for k6=1:length(I6)
I7(k6)=-acos(1-(((n-1)^2*I6(k6)^2)/(2*L^2)))/pi*180;

end

for k7=1:length(I8)
    I9(k7)=acos(1-(((n-1)^2*I8(k7)^2)/(2*L^2)))/pi*180;
end

xn=I9+I7;
new_result=xn;
%plot(t,new_result);

%time
[ite,yin,t]=display_nu(new_result,time,dt,ite,t,2,1,1);



%freq
[aa,f,Pn1]=display_nu2(new_result,time,dt,ite,t,2,2,1);
new_result=yin;
end

if sel_point==2
     p=1;
for i=1:gg
  net(i) = y(i,p);
end
t=0:dt:(ite-1)*dt;
yn = net;
  NUM=[1,-1];
DEN=0.005;
x1=filter(NUM,DEN,yn);
  [g1,h1]=butter(4,0.08,'low');
  %[b,a]=butter(4,0.05,'high');
  %yn1=filter(g1,h1,x1);
 yn=filter(g1,h1,x1);
```

```
  new_result=yn;
  %plot(t,new_result);
  %time
  new_result = new_result*180*7/22;
  [ite,yin,t]=display_nu(new_result,time,dt,ite,t,3,1,1);
  %freq
  [aa,f,Pn1]=display_nu2(new_result,time,dt,ite,t,3,2,1);
new_result=yin;
yin=yin.*100*4;
new_result=yin;
end

if sel_point==3
     p=1;
for i=1:gg
  net(i) = y(i,p);
end
t=0:dt:(ite-1)*dt;
yn = net;
  NUM=[1,-1];
DEN=0.005;
x1=filter(NUM,DEN,yn);
[b,a]=butter(4,0.05,'high');
[g1,h1]=butter(4,0.800,'low');
x2=filter(b,a,x1);
x3=filter(g1,h1,x2);

 NUM=[1,-1];
DEN=0.005;
x4=filter(NUM,DEN,x3);
new_result=x4;
%time
[ite,yin,t]=display_nu(new_result,time,dt,ite,t,4,1,1);
%freq


[aa,f,Pn1]=display_nu2(new_result,time,dt,ite,t,4,2,1);
yin=yin.*100*4;
new_result=yin;
end


 if sel_point==4
 for i=1:gg
  net(i) = y(i,n);
 end

yn = net;

%plot(t,yn);
%xlabel('Time (sec)');
%ylabel('Displacement (m)');
```

```
%grid
new_result=yn;
%time
[ite,yin,t]=display_nu(new_result,time,dt,ite,t,5,1,1);
%freq

%display_nu(new_result,time,dt,ite,t,5,1,1);
[aa,f,Pn1]=display_nu2(new_result,time,dt,ite,t,5,2,1);

new_result=yin;
end
if sel_point==5
     for i=1:gg
  net(i) = y(i,n);
 end

yn = net;
NUM=[1,-1];
DEN=0.005;
yn=filter(NUM,DEN,yn);
new_result=yn;
%time
[ite,yin,t]=display_nu(new_result,time,dt,ite,t,6,1,1);
%freq

[aa,f,Pn1]=display_nu2(new_result,time,dt,ite,t,6,2,1);

yin=yin.*100;
new_result=yin;
end
if sel_point==6
   for i=1:gg
  net(i) = y(i,n);
 end

yn = net;
NUM=[1,-1];
DEN=0.005;
x1=filter(NUM,DEN,yn);
x2=filter(NUM,DEN,x1);
[b,a]=butter(4,0.55,'high');
[g1,h1]=butter(4,0.8,'low');

x3=filter(b,a,x2);
yn=filter(g1,h1,x3);
new_result=yn;
%plot(t,yn);
%time

%freq
[ite,yin,t]=display_nu(new_result,time,dt,ite,t,7,1,1);
```

```
[aa,f,Pn1]=display_nu2(new_result,time,dt,ite,t,7,2,1);
yin=yin.*100;
new_result=yin;
end
if sel_point==7
[b,a]=butter(4,0.005,'high');
[g1,h1]=butter(4,0.1200,'low');
x1=filter(b,a,yn);
yn=filter(g1,h1,x1);
new_result=yn;
%time
[ite,yin,t]=display_nu(new_result,time,dt,ite,t,8,1,1);
%freq
[aa,f,Pn1]=display_nu(new_result,time,dt,ite,t,8,2,1);
Pn1=Pn1';
End
```

III. Function 3 :

```
function [ite,yin,t]=display_nu(yinm,time,dt,ite,t,chc,chca,chcb)

%ite =56;  % added on 5th of June

if chc == 8
 clc
  % frc is the cutoff frequency of highpass filter
  % frc=input('Enter cutoff frequency of highpass filter: ');

 frc = 11;
 freq = (1/dt);
 Fs = round(freq);
 ws = frc/(Fs/2);

 % Filtering the signal
 [B A] = butter(6,ws,'high');
 yinf = filter (B,A,yinm);
end

%chca= 2;

%while chca < 3U

%chca=menu('Required Domain','Time','Frequency','Main menu')

if chc <= 8
  yin = yinm;
end

%chcb = 0;UUU   % n
 if chca==1
```

```
  if chc > 1
  %chcb = menu('Scalling','Meter','Millimeter')
   if chcb ==2
     yin = yinm*1000;
     else
     yin = yinm;
    end
    if chc == 8
     if chcb ==2
      yin = yinf*1000;
     else
       yin = yinf;
     end
    end
   end

   % figure(1)UUUUUUUUUUUUUUUUUUUU
   %yin(1:ite).LineWidth = 2;
%comment
    %plot(t(1:ite),yin(1:ite),'k','LineWidth',1)   %n
   % axis([0 1.5 -0.15 0.15]);


   if chcb==1
     if chc==1
      %title('Input Torque')
     % ylabel('Torque (Nm)')
     elseif chc==2
      %title('Hub Angle')
      %ylabel('Angle (degree)')
     elseif chc==3
      %title('Hub Velocity')
      %ylabel('Velocity (rad/sec)')
     elseif chc==4
      %title('Hub Acceleration')
     %ylabel('Acceleration (meter/sec-sec)')
     elseif chc==5
      %title('End-Point Position')
      %ylabel('Displacement (meter)')
     elseif chc==6
      %title('End-Point Velocity')
     % ylabel('Velocity (meter/sec)')
     elseif chc==7
      %title('End-Point Acceleration')
     % ylabel('Acceleration (meter/sec-sec)')
     elseif chc==8
      %title('End-Point Residual')
     % ylabel('Displacement (meter)')
    end
   end
  if chcb==2
     if chc==1
     % title('Torque input to the manipulator')
    % ylabel('Torque (Nm)')
    elseif chc==2
   %  title('Position of the hub')
    % ylabel('Displacement (mm)')
    elseif chc==3
    % title('Velocity of the hub')
   %  ylabel('Velocity (mm/sec)')
    elseif chc==4
     %title('Acceleration of the hub point')
    % ylabel('Acceleration (mm/sec-sec)')
    elseif chc==5
    % title('Position of the end-point')
    % ylabel('Displacement (mm)')
    elseif chc==6
    % title('Velocity of the end-point')
    % ylabel('Velocity (mm/sec)')
    elseif chc==7
     %title('Acceleration of the end-point')
    % ylabel('Acceleration (mm/sec-sec)')
    elseif chc==8
     %title('Residual motion of the end-point')
    % ylabel('Displacement (mm)')
   end
  end

elseif chca==2

  freq=(1/dt);
  Fs = round(freq);
  len = length(yin);
  n = round(log(len)/log(2));
  if 2^n>len
    n = n-1;
  end

  %uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu

  S = 4096;

  % aa = round(S/40.96);
  aa = round(S/47);

% Spectrum analysis

  X = spectrum(yinm,S);
  n = max(size(X));
  m = min(size(X));
  f = (1:n-1)/n*Fs/2;
  Pn1=2*sqrt(X(2:n,1)/length(X(2:n,1))); Pn1=Pn1*2.5;

  Xtemp = fft(yinm,S);
  Pnu=2*sqrt(X(2:n,1)/length(X(2:n,1))); Pnu=Pnu*2.5;
```

```
end
```

D2 Console Application code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using fd2;
using MathWorks.MATLAB.NET.Arrays;
using MathWorks.MATLAB.NET.Utility;
using System.IO;
using System.Diagnostics;

namespace fd2
{
    class Program
    {
        static void Main(string[] args)
        {
            fd2.Class1 obj = null;
            MWNumericArray input1 = null;
            MWNumericArray input2 = null;
            MWNumericArray input3 = null;
            MWNumericArray input4 = null;
            MWNumericArray input5 = null;
            MWNumericArray input6 = null;
            MWNumericArray input7 = null;
            MWNumericArray input8 = null;
            MWNumericArray input9 = null;
            MWNumericArray input10 = null;
            MWNumericArray input11 = null;
            MWNumericArray input12 = null;
            MWNumericArray input13 = null;
            MWNumericArray input14 = null;


            MWNumericArray output1 = null;
            MWArray[] result1 = null;
            MWNumericArray output2 = null;
            MWArray[] result2 = null;
            MWNumericArray output3 = null;
            MWArray[] result3 = null;
            MWNumericArray output4 = null;
            MWArray[] result4 = null;
            MWNumericArray output5 = null;
            MWArray[] result5 = null;
            MWNumericArray output6 = null;
            MWArray[] result6 = null;
            string[] arr;
            try
            {
                using (StreamReader sr = new StreamReader("C:\\work\\Test1.txt"))
                {
```

```
            String line = sr.ReadToEnd();
            //Console.WriteLine(line);
            arr = line.Split(';');
            foreach (string str in arr)
            {
                Console.WriteLine(str);
                //input1 = Convert.ToDouble(arr[0]);

            }


        }
        input1 = Convert.ToDouble(arr[0]);
        input2 = Convert.ToDouble(arr[1]);
        input3 = Convert.ToDouble(arr[2]);
        input4 = Convert.ToDouble(arr[3]);
        input5 = Convert.ToDouble(arr[4]);
        input6 = Convert.ToDouble(arr[5]);
        input7 = Convert.ToDouble(arr[6]);
        input8 = Convert.ToDouble(arr[7]);
        input9 = Convert.ToDouble(arr[8]);
        input10 = Convert.ToDouble(arr[9]);
        input11 = Convert.ToDouble(arr[10]);
        input12 = Convert.ToDouble(arr[11]);
        //Console.WriteLine(input8);

        obj = new Class1();
        //input1 = 0.96;// Convert.ToDouble(str[0]);
        //input2 = 0.0032004;//str[1]
        //input3 = .01923004;
        //input4 = 0;
        //input5 = 0;
        //input6 = 4;
        //input7 = 20;
        //input8 = 0.20;
        //input9 = 0;
        //input10 = 71E09;
        //input11 = 2710;
        //input12 = 3;
        input13 = Convert.ToDouble(arr[12]);
        //input13 = 7;
        input14 = 3;



        result1 = obj.fd2(1, input1, input2, input3, input4, input5, input6, input7, input8, input9, input10, input11,
input12, input13, input14);
        output1 = (MWNumericArray)result1[0];
        input14 = 2;
        result2 = obj.fd2(1, input1, input2, input3, input4, input5, input6, input7, input8, input9, input10, input11,
input12, input13, input14);
        output2 = (MWNumericArray)result2[0];
        input14 = 3;
        result3 = obj.fd2(1, input1, input2, input3, input4, input5, input6, input7, input8, input9, input10, input11,
input12, input13, input14);
        output3 = (MWNumericArray)result3[0];
        input14 = 4;
        result4 = obj.fd2(1, input1, input2, input3, input4, input5, input6, input7, input8, input9, input10, input11,
input12, input13, input14);
        output4 = (MWNumericArray)result4[0];
        input14 = 5;
        result5 = obj.fd2(1, input1, input2, input3, input4, input5, input6, input7, input8, input9, input10, input11,
input12, input13, input14);
        output5 = (MWNumericArray)result5[0];
        input14 = 6;
        result6 = obj.fd2(1, input1, input2, input3, input4, input5, input6, input7, input8, input9, input10, input11,
input12, input13, input14);
        output6 = (MWNumericArray)result6[0];

        double[,] arr2 = (double[,])((MWNumericArray)output3).ToArray(MWArrayComponent.Real);
        double[,] arr3 = (double[,])((MWNumericArray)output2).ToArray(MWArrayComponent.Real);
        double[,] arr4 = (double[,])((MWNumericArray)output5).ToArray(MWArrayComponent.Real);
        double[,] arr5 = (double[,])((MWNumericArray)output6).ToArray(MWArrayComponent.Real);
        for (int i = 0; i < arr2.Length ; i++)
        {
        //Console.Write(arr2[0,i]);
        StreamWriter sw = new StreamWriter("C:\\work\\Test2.txt", true, Encoding.ASCII);
        sw.Write((arr2[0, i]));
        sw.Write(";");
        sw.Close();
        }



        for (int i = 0; i < arr3.Length; i++)
        {

          StreamWriter sw = new StreamWriter("C:\\work\\Test3.txt", true, Encoding.ASCII);
          sw.Write(Convert.ToDouble(arr3[0, i]));
          sw.Write(";");
          sw.Close();
        }




        for (int i = 0; i < arr4.Length; i++)
        {

          StreamWriter sw = new StreamWriter("C:\\work\\Test4.txt", true, Encoding.ASCII);
          sw.Write(Convert.ToDouble(arr4[0, i]));
          sw.Write(";");
          sw.Close();
        }
        for (int i = 0; i < arr5.Length; i++)
        {
          Console.WriteLine(arr5[i,0]);
          StreamWriter sw = new StreamWriter("C:\\work\\Test5.txt", true, Encoding.ASCII);
          sw.Write(Convert.ToDouble(arr5[i,0]));
          sw.Write(";");
          sw.Close();
        }
        //StreamWriter sw = new StreamWriter("C:\\work\\Test2.txt", true, Encoding.ASCII);
       // sw.Write(Convert.ToDouble(arr2[0,i]));
       // sw.Close();
      }
      catch (Exception ex)
      {
        Console.WriteLine(ex.Message);
      }
    }
  }
}
```

D3 ASP.NET Back End (Web Design Code):

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="~/finiteDiffEleSimulation.aspx.cs" Inherits="_Default"
MasterPageFile="~/Login.master" %>

<%@ Register assembly="AjaxControlToolkit" namespace="AjaxControlToolkit" tagprefix="cc1" %>

<%@ Register assembly="System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" namespace="System.Web.UI.DataVisualization.Charting" tagprefix="asp" %>


<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
  <title></title>
  <style type="text/css">
    .style1
    {
      width: 680px;
    }
    .style20
    {
    }
    .lblPadding
    {
      padding-right:10px
    }
    .style21
    {
      height: 60px;
    }
    .style28
    {
      width: 338px;
    }
    .style29
    {
      width: 338px;
    }
    .style30
    {
      height: 10px;
    }
    </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
  <body>
  <center>
  <asp:ScriptManager ID="ScriptManager1" runat="server">
        </asp:ScriptManager>
  <table class="style1" cellpadding="10px">
    <tr>
      <td colspan="4" class="style30" align="center">
        <asp:Label ID="Label1" runat="server"
          Text="SCEFMAS - Finite Difference Simulation"
          ForeColor="Blue" Font-Bold="True" Font-Size="Larger"></asp:Label>
      </td>
    </tr>
    <tr>
      <td class="style21">
        <asp:Label ID="lblLength" runat="server" Text="Length" CssClass="lblPadding"
          Font-Names="Arial" Font-Size="Small"></asp:Label>
        <asp:Label ID="Label4" runat="server" Text="Label" Font-Names="Arial" Font-Size="Small"></asp:Label>
        <br />
        <asp:TextBox ID="TextBox4" runat="server" Width="100px"></asp:TextBox>
        <cc1:SliderExtender ID="SliderExtender1" runat="server"
          BehaviorID="TextBox4_SliderExtender" Maximum="1" Minimum="0.1"
          Steps = "100000" BoundControlID="Label4"
          Decimals="2" Length="100"
          TargetControlID="TextBox4">
        </cc1:SliderExtender>
      </td>
      <td class="style21">
        <asp:Label ID="lblWidth" runat="server" Text="Width(cm)" CssClass="lblPadding"
        Font-Names="Arial" Font-Size="Small"></asp:Label>
        <asp:Label ID="Label8" runat="server" Text="Label" Font-Names="Arial" Font-Size="Small"></asp:Label>
        <br />
        <asp:TextBox ID="TextBox8" runat="server"></asp:TextBox>
        <cc1:SliderExtender ID="TextBox8_SliderExtender" runat="server"
          BehaviorID="TextBox8_SliderExtender" Maximum="3" Minimum="1"
          Steps="500000" BoundControlID="Label8"
          Decimals="4" Length="100"
          TargetControlID="TextBox8">
        </cc1:SliderExtender>
      </td>
      <td class="style21">
        <asp:Label ID="lblThick" runat="server" Text="Thick(cm)" CssClass="lblPadding"
        Font-Names="Arial" Font-Size="Small"></asp:Label>
        <asp:Label ID="Label6" runat="server" Text="Label"
        Font-Names="Arial" Font-Size="Small"></asp:Label>
        <br />
        <asp:TextBox ID="TextBox6" runat="server"></asp:TextBox>
        <cc1:SliderExtender ID="TextBox6_SliderExtender" runat="server"
          BehaviorID="TextBox6_SliderExtender" Maximum="0.6" Minimum="0.2"
          Steps = "10000" BoundControlID="Label6"
          Decimals="4" Length="100"
          TargetControlID="TextBox6">
        </cc1:SliderExtender>
      </td>
      <td class="style21">
      <asp:Label ID="lblTime" runat="server" Text="Time(sec)" CssClass="lblPadding"
      Font-Names="Arial" Font-Size="Small"></asp:Label>
       <asp:Label ID="Label5" runat="server" Text="Label"
       Font-Names="Arial" Font-Size="Small"></asp:Label>
        <br />
        <asp:TextBox ID="TextBox5" runat="server"></asp:TextBox>
        <cc1:SliderExtender ID="TextBox5_SliderExtender" runat="server"
          BehaviorID="TextBox5_SliderExtender" Maximum="5" Minimum="1"
          Steps = "5" BoundControlID="Label5" Length="100"
          TargetControlID="TextBox5">
        </cc1:SliderExtender>
      </td>
    </tr>
    <tr>
      <td class="style20">
       <asp:Label ID="lblSegments" runat="server" Text="Segments" CssClass="lblPadding"
       Font-Names="Arial" Font-Size="Small"></asp:Label>
       <asp:Label ID="Label7" runat="server" Text="Label"
       Font-Names="Arial" Font-Size="Small"></asp:Label>
       <br />
       <asp:TextBox ID="TextBox7" runat="server"></asp:TextBox>
        <cc1:SliderExtender ID="TextBox7_SliderExtender" runat="server"
          BehaviorID="TextBox7_SliderExtender" Maximum="30" Minimum="5"
          Steps ="25" BoundControlID="Label7" Length="100"
          TargetControlID="TextBox7">
        </cc1:SliderExtender>
       </td>
      <td class="style20">

       <asp:Label ID="lblHub" runat="server" Text="Hub Inertia (Kgm2)" CssClass="lblPadding"
       Font-Names="Arial" Font-Size="Small"></asp:Label>
        <asp:Label ID="Label10" runat="server" Text="Label"
        Font-Names="Arial" Font-Size="Small"></asp:Label>
        <br />
        <asp:TextBox ID="TextBox10" runat="server"></asp:TextBox>
        <cc1:SliderExtender ID="TextBox10_SliderExtender" runat="server"
          BehaviorID="TextBox10_SliderExtender" Maximum="0.001" Minimum="0"
          Steps="100000" BoundControlID="Label10" Length="100"
          Decimals="9"
          TargetControlID="TextBox10">
        </cc1:SliderExtender>
        </td>
      <td class="style20">
       <asp:Label ID="lblPayLoad" runat="server" Text="PayLoad(gms)" CssClass="lblPadding"
       Font-Names="Arial" Font-Size="Small"></asp:Label>
        <asp:Label ID="Label12" runat="server" Text="Label"
        Font-Names="Arial" Font-Size="Small"></asp:Label>
        <br />
        <asp:TextBox ID="TextBox12" runat="server"></asp:TextBox>
        <cc1:SliderExtender ID="TextBox12_SliderExtender" runat="server"
          BehaviorID="TextBox12_SliderExtender" Maximum="100" Minimum="0"
          Steps="10000" BoundControlID="Label12" Length="100"
          Decimals="5"
          TargetControlID="TextBox12">
        </cc1:SliderExtender>
        </td>
        <td class="style20">
        <asp:Label ID="lblStability" runat="server" Text="Stability" CssClass="lblPadding"
        Font-Names="Arial" Font-Size="Small"></asp:Label>
        <asp:Label ID="Label9" runat="server" Text="Label"
        Font-Names="Arial" Font-Size="Small"></asp:Label>
        <br />
        <asp:TextBox ID="TextBox9" runat="server"></asp:TextBox>
        <cc1:SliderExtender ID="TextBox9_SliderExtender" runat="server"
          BehaviorID="TextBox9_SliderExtender" Maximum="0.24" Minimum="0"
          Steps="100" BoundControlID="Label9" Length="100"
          Decimals="1"
          TargetControlID="TextBox9">
        </cc1:SliderExtender>
        </td>
    </tr>
    <tr>
      <td class="style20">
       <asp:Label ID="lblDampingFactor" runat="server" Text="Damping Factor"
          CssClass="lblPadding" Font-Names="Arial" Font-Size="Small"></asp:Label>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
        ErrorMessage="*" ControlToValidate="DropDownList1" ForeColor="Red"
          InitialValue="--Select--"></asp:RequiredFieldValidator>
          <br />
          <asp:DropDownList ID="DropDownList1" runat="server" Height="22px"
          Width="100px">
          <asp:ListItem Text="--Select--" Value="--Select--"></asp:ListItem>
        <asp:ListItem Text="0" Value="0"></asp:ListItem>
        <asp:ListItem Text="0.148" Value="0.148"></asp:ListItem>
        <asp:ListItem Text="0.036" Value="0.036"></asp:ListItem>
        <asp:ListItem Text="0.024" Value="0.024"></asp:ListItem>
        <asp:ListItem Text="0.012" Value="0.012"></asp:ListItem>
        <asp:ListItem Text="0.006" Value="0.006"></asp:ListItem>
        <asp:ListItem Text="0.005" Value="0.005"></asp:ListItem>
        <asp:ListItem Text="0.004" Value="0.004"></asp:ListItem>
        <asp:ListItem Text="0.003" Value="0.003"></asp:ListItem>
        <asp:ListItem Text="0.002" Value="0.002"></asp:ListItem>
        </asp:DropDownList>
        </td>
      <td class="style20">

       <asp:Label ID="lblYoungs" runat="server" Text="Young's Modulus"
          CssClass="lblPadding" Font-Names="Arial" Font-Size="Small"></asp:Label>
          <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
        ErrorMessage="*" ControlToValidate="DropDownList2" ForeColor="Red"
          InitialValue="--Select--"></asp:RequiredFieldValidator>
          <br />
          <asp:DropDownList ID="DropDownList2" runat="server" Height="22px"
          Width="100px">
          <asp:ListItem Text="--Select--" Value="--Select--"></asp:ListItem>
        <asp:ListItem Text="203E09" Value="203E09"></asp:ListItem>
        <asp:ListItem Text="71E09" Value="71E09"></asp:ListItem>
        <asp:ListItem Text="71E08" Value="71E08"></asp:ListItem>
        <asp:ListItem Text="68E07" Value="68E07"></asp:ListItem>
        <asp:ListItem Text="70E06" Value="70E06"></asp:ListItem>
        </asp:DropDownList>
          </td>
      <td class="style20">

       <asp:Label ID="lblDensity" runat="server" Text="Density(Kg/m3)"
          CssClass="lblPadding" Font-Names="Arial" Font-Size="Small"></asp:Label>
          <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server"
        ErrorMessage="*" ControlToValidate="DropDownList3" ForeColor="Red"
          InitialValue="--Select--"></asp:RequiredFieldValidator>
          <br />
          <asp:DropDownList ID="DropDownList3" runat="server" Height="22px"
          Width="100px">
          <asp:ListItem Text="--Select--" Value="--Select--"></asp:ListItem>
        <asp:ListItem Text="19210" Value="19210"></asp:ListItem>
        <asp:ListItem Text="7831" Value="7831"></asp:ListItem>
        <asp:ListItem Text="2710" Value="2710"></asp:ListItem>
        <asp:ListItem Text="2310" Value="2310"></asp:ListItem>
        <asp:ListItem Text="2100" Value="2100"></asp:ListItem>
        </asp:DropDownList>
          </td>
          <td class="style20">
        <br />
          </td>
    </tr>
    <tr>
      <td class="style20">
      <asp:Label ID="lblInput" runat="server" Text="Input"
          CssClass="lblPadding" Font-Names="Arial" Font-Size="Small"></asp:Label>
          <asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server"
        ErrorMessage="*" ControlToValidate="DropDownList4" ForeColor="Red"
          InitialValue="--Select--"></asp:RequiredFieldValidator>
          <br />
          <asp:DropDownList ID="DropDownList4" runat="server" Height="22px"
          Width="100px">
          <asp:ListItem Text="--Select--" Value="--Select--"></asp:ListItem>
        <asp:ListItem Text="Random" Value="1"></asp:ListItem>
        <asp:ListItem Text="Bang-Bang" Value="3"></asp:ListItem>
        <asp:ListItem Text="Composite PRBS" Value="2"></asp:ListItem>
        </asp:DropDownList>
        </td>
      <td class="style20">
      <asp:Label ID="lblSimOptions" runat="server" Text="Simulation Options"
          CssClass="lblPadding" Font-Names="Arial" Font-Size="Small"></asp:Label>
          <asp:RequiredFieldValidator ID="RequiredFieldValidator5" runat="server"
        ErrorMessage="*" ControlToValidate="ddlSimOption" ForeColor="Red"
          InitialValue="--Select--"></asp:RequiredFieldValidator>
          <br />
          <asp:DropDownList ID="ddlSimOption" runat="server" Height="22px"
          Width="150px">
          <asp:ListItem Text="--Select--" Value="--Select--"></asp:ListItem>
        <asp:ListItem Text="Hub Angle" Value="Hub Angle"></asp:ListItem>
        <asp:ListItem Text="Hub Velocity" Value="Hub Velocity"></asp:ListItem>
        <asp:ListItem Text="Hub Accelaration" Value="Hub Accelaration"></asp:ListItem>
        <asp:ListItem Text="End point displacement" Value="End point displacement"></asp:ListItem>
        <asp:ListItem Text="End point velocity" Value="End point velocity"></asp:ListItem>
        <asp:ListItem Text="End point accelaration" Value="End point accelaration"></asp:ListItem>
        <asp:ListItem Text="Residual Movement" Value="Residual Movement"></asp:ListItem>
        </asp:DropDownList>
      </td>
      <td class="style20">
        <asp:Button ID="btnSubmit" runat="server" Text="Results"
          onclick="btnSubmit_Click" Height="30px" Width="75px" />
      </td>
      <td class="style20">
        <asp:Button ID="btnCancel" runat="server" Text="Back"
           Height="30px" Width="75px" CausesValidation="False"
          onclick="btnCancel_Click" />
      </td>
    </tr>
    </table>
    <br />
    <asp:Panel ID="graphs" runat="server" Width="800px" Visible="false">
      <table class="style1">
      <tr>
      <td colspan="2">
        <asp:Label ID="Label2" runat="server" Text="Simulation Results"
          Font-Size="Medium" ForeColor="#996600"></asp:Label>
      </td>
      </tr>
        <tr>
          <td class="style28">
          <asp:Chart ID="Chart1" runat="server" Width="350px" Height="350px">
          <series>
          <asp:Series ChartType="Spline" Name="Series1">
          </asp:Series>
          </series>
          <chartareas>
          <asp:ChartArea Name="ChartArea1">
          <AxisX Title="Index1"></AxisX>
          <AxisY Title="Val1"></AxisY>
          </asp:ChartArea>
          </chartareas>
        </asp:Chart>
          </td>
          <td class="style29">
          <asp:Chart ID="Chart2" runat="server" Width="350px" Height="350px">
          <Series>
          <asp:Series ChartArea="ChartArea1" ChartType="Spline" Name="Series1">
          </asp:Series>
          </Series>
          <ChartAreas>
          <asp:ChartArea Name="ChartArea1">
           <AxisX Title="Index2"></AxisX>
          <AxisY Title="Val2"></AxisY>
          </asp:ChartArea>
          </ChartAreas>
        </asp:Chart>
          </td>
        </tr>
      </table>
```

```
    </asp:Panel>
    </center>
</body>
</asp:Content>
```

D4 ASP.NET Back End (C# Code):

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using fd2;
using MathWorks.MATLAB.NET.Arrays;
using MathWorks.MATLAB.NET.Utility;
using System.Web.UI.DataVisualization.Charting;
using System.IO;
using System.Diagnostics;



public partial class _Default : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {
    if (!Page.IsPostBack)
    {
      Label1.Text = Session["SimulationType"].ToString();
    }
  }

  //Simulation options to values
  public int returnSimulationValue(String simOption)
  {
    Dictionary<String, int> btnVal = new Dictionary<String, int>();
    btnVal.Add("Hub Angle", 1);
    btnVal.Add("Hub Velocity", 2);
    btnVal.Add("Hub Accelaration", 3);
    btnVal.Add("End point displacement", 4);
    btnVal.Add("End point velocity", 5);
    btnVal.Add("End point accelaration", 6);
    btnVal.Add("Residual Movement", 7);
    return btnVal[simOption];
  }


  //Simulation option into Ranges
  public List<Double> returnSimulationRange(String simOption)
  {
    Dictionary<String, List<Double>> btnRange = new Dictionary<String, List<Double>>();
    btnRange.Add("Hub Angle", new List<Double> { 0, 50, -20, 20 });
    btnRange.Add("Hub Velocity", new List<Double> { -200, 300, -20, 20 });
    btnRange.Add("Hub Accelaration", new List<Double> { -1500, 1500, -20, 20 });
    btnRange.Add("End point displacement", new List<Double> { -0.5, 0.9, -20, 20 });
    btnRange.Add("End point velocity", new List<Double> { -20, 20, -20, 20 });
    btnRange.Add("End point accelaration", new List<Double> { -500, 500, -20, 20 });
    btnRange.Add("Residual Movement", new List<Double> { -0.005, 0.005, -20, 20 });
    return btnRange[simOption];
}

//Clearing text files
public void clearFiles()
{
    System.IO.File.WriteAllBytes("C:\\work\\Test1.txt", new byte[0]);
    System.IO.File.WriteAllBytes("C:\\work\\Test2.txt", new byte[0]);
    System.IO.File.WriteAllBytes("C:\\work\\Test3.txt", new byte[0]);
    System.IO.File.WriteAllBytes("C:\\work\\Test4.txt", new byte[0]);
    System.IO.File.WriteAllBytes("C:\\work\\Test5.txt", new byte[0]);
}

//Writing input values into Text
public void writeInputvalues(int simVal)
{
    try
    {
        StreamWriter sw = new StreamWriter("C:\\work\\Test1.txt", true, Encoding.ASCII);
        sw.Write(Convert.ToDouble(TextBox4.Text));
        sw.Write(";");
        double a = (Convert.ToDouble(TextBox6.Text));
        a = a / 100;
        sw.Write(a);
        sw.Write(";");
        double b = (Convert.ToDouble(TextBox8.Text));
        b = b / 100;
        sw.Write(b);
        sw.Write(";");
        sw.Write(Convert.ToDouble(TextBox10.Text));
        sw.Write(";");
        double c = (Convert.ToDouble(TextBox12.Text));
        c = c / 100;
        sw.Write(c);
        sw.Write(";");
        sw.Write(Convert.ToDouble(TextBox5.Text));
        sw.Write(";");
        sw.Write(Convert.ToDouble(TextBox7.Text));
        sw.Write(";");
        sw.Write(Convert.ToDouble(TextBox9.Text));
        sw.Write(";");
        sw.Write(Convert.ToDouble(DropDownList1.SelectedItem.Value));
        sw.Write(";");
        sw.Write(Convert.ToDouble(DropDownList2.SelectedItem.Value));
        sw.Write(";");
        sw.Write(Convert.ToDouble(DropDownList3.SelectedItem.Value));
        sw.Write(";");
        sw.Write(Convert.ToDouble(DropDownList4.SelectedItem.Value));
```

```
      sw.Write(";");
      sw.Write(simVal);
      sw.Close();
    }
    catch
    {

    }
  }

  public void getButton1Data()
  {
    String simOption = ddlSimOption.SelectedItem.Text;//Hubangle
    int simVal = returnSimulationValue(simOption);//1

    List<Double> simRange = returnSimulationRange(simOption);

    clearFiles();

    string[] arrText2;
    string[] arrText3;
    string[] arrText4;
    string[] arrText5;

    //Writing input data into Text1
    writeInputvalues(simVal);
    try
    {
      //Running the console process
      ProcessStartInfo startinfo = new ProcessStartInfo();
      startinfo.FileName =
@"C:\\Users\\Avinash\\Desktop\\ClassLibrary3\\ClassLibrary3\\bin\\debug\\ClassLibrary3.exe";
      startinfo.CreateNoWindow = true;
      startinfo.UseShellExecute = true;
      Process myProcess = Process.Start(startinfo);
      myProcess.WaitForExit();

      try
      {
        using (StreamReader sr = new StreamReader("C:\\work\\Test3.txt"))
        {
          String line = sr.ReadToEnd();
          arrText3 = line.Split(';');
          sr.Close();
        }
        using (StreamReader sr = new StreamReader("C:\\work\\Test2.txt"))
        {
          String line = sr.ReadToEnd();
          arrText2 = line.Split(';');
          sr.Close();
        }

        using (StreamReader sr = new StreamReader("C:\\work\\Test4.txt"))
        {
          String line = sr.ReadToEnd();
          arrText4 = line.Split(';');
    sr.Close();
}

using (StreamReader sr = new StreamReader("C:\\work\\Test5.txt"))
{
    String line = sr.ReadToEnd();
    arrText5 = line.Split(';');
    sr.Close();
}

arrText2 = arrText2.Take(arrText2.Count() - 1).ToArray();
arrText3 = arrText3.Take(arrText3.Count() - 1).ToArray();
arrText4 = arrText4.Take(arrText4.Count() - 1).ToArray();
arrText5 = arrText5.Take(arrText5.Count() - 1).ToArray();

double high = Double.MinValue;
for (int i = 0; i < arrText5.Length; i++)
{
    double val;
    bool foo = Double.TryParse(arrText5[i], out val);
    if (foo && (val > high))
    {
        high = val;
    }
}

double low = Double.MaxValue;
for (int i = 0; i < arrText5.Length; i++)
{
    double val;
    bool foo = Double.TryParse(arrText5[i], out val);
    if (foo && (val < low))
    {
        low = val;
    }
}

DataTable dt = new DataTable();
dt.Columns.Add("Time(Sec)");
dt.Columns.Add("Angle(Degree)");
DataRow dr;
for (int i = 0; i < arrText2.Length; i++)
{
    dr = dt.NewRow();
    dr["Time(Sec)"] = i;
    dr["Angle(Degree)"] = arrText2[i];
    dt.Rows.Add(dr);
}

DataTable dt1 = new DataTable();
dt1.Columns.Add("Frequency(Hz)");
dt1.Columns.Add("Magnitude");
DataRow dr1;
for (int i = 0; i < arrText4.Length; i++)
{
    if (i == 100)
```

```
                    break;
                dr1 = dt1.NewRow();
                dr1["Frequency(Hz)"] = i;
                dr1["Magnitude"] = arrText5[i];
                dt1.Rows.Add(dr1);
            }
            List<Double> range = simRange;
            Double xMaxRange = Convert.ToDouble(arrText3[arrText3.Length - 1]);
            Chart1.Series["Series1"].ChartType = SeriesChartType.Line;
            Chart1.Series["Series1"].Color = System.Drawing.Color.BurlyWood;
            Chart1.DataSource = dt;
            Chart1.Series["Series1"].XValueMember = dt.Columns[0].ColumnName;
            Chart1.Series["Series1"].YValueMembers = dt.Columns[1].ColumnName;
            Chart1.ChartAreas[0].AxisX.IsStartedFromZero = true;
            Chart1.ChartAreas[0].AxisY.IsStartedFromZero = true;
            Chart1.ChartAreas[0].AxisX.IsMarginVisible = false;
            Chart1.ChartAreas[0].AxisY.Minimum = range[0];
            Chart1.ChartAreas[0].AxisY.Maximum = range[1];
            Chart1.ChartAreas[0].AxisY.Interval = 10;

            List<Double> inter = new List<Double>();
            int len = arrText3.Length;
            double max = Convert.ToDouble(arrText3[len - 1]);
            double interval = max / 4;
            inter.Add(0);
            for (int i = 1; i < 4; i++)
            {
                double temp = inter[i - 1] + interval;
                temp = Math.Round(temp, 2);
                inter.Add(temp);
            }
            inter.Add(Math.Round(max, 2));
            int startOffset = 0;
            int endOffset = (int)max;
            ++endOffset;
            foreach (Double nums in inter)
            {
                CustomLabel lbl = new CustomLabel(startOffset, endOffset, nums.ToString(), 0, LabelMarkStyle.None);
                Chart1.ChartAreas[0].AxisX.CustomLabels.Add(lbl);
                startOffset = (startOffset + (int)(len/4));
                if (startOffset > len)
                    startOffset = len;
                endOffset = (endOffset + (int)(len/4));
                if (endOffset > len)
                    endOffset = len;
            }

            Chart1.ChartAreas[0].AxisX.MajorGrid.LineWidth = 0;
            Chart1.ChartAreas[0].AxisY.MajorGrid.LineWidth = 0;
            Chart1.ChartAreas[0].AxisX.LabelStyle.Font = new System.Drawing.Font("Comic Sans", 7F,
System.Drawing.FontStyle.Bold);
            Chart1.ChartAreas[0].BackColor = System.Drawing.Color.Azure;
            Chart1.DataBind();

            double first = Convert.ToDouble(arrText4[0]);
            first = Math.Round(first, 2);
```

```
                Chart2.Series["Series1"].ChartType = SeriesChartType.Line;
                Chart2.Series["Series1"].Color = System.Drawing.Color.BurlyWood;
                Chart2.DataSource = dt1;
                Chart2.Series["Series1"].XValueMember = dt1.Columns[0].ColumnName;
                Chart2.Series["Series1"].YValueMembers = dt1.Columns[1].ColumnName;
                Chart2.ChartAreas[0].AxisY.Minimum = Math.Round(low, 2);// range[3];
                Chart2.ChartAreas[0].AxisY.Maximum = Math.Round(high, 2);//range[4];
                Chart2.ChartAreas[0].AxisY.Interval = 0.5;

                List<Double> inter1 = new List<Double>();
                //int len1 = arrText4.Length;
                int len1 = 100;
                double max1 = Convert.ToDouble(arrText4[99]);
                double interval1 = max1 / 4;
                inter1.Add(first);
                for (int i = 1; i < 4; i++)
                {
                    double temp = inter1[i - 1] + interval1;
                    temp = Math.Round(temp, 2);
                    inter1.Add(temp);
                }
                inter1.Add(Math.Round(max1, 2));
                int startOffset1 = 0;
                int endOffset1 = 1;
                foreach (Double nums in inter1)
                {
                    CustomLabel lbl = new CustomLabel(startOffset1, endOffset1, Math.Round(nums,0).ToString(), 0,
LabelMarkStyle.None);
                    Chart2.ChartAreas[0].AxisX.CustomLabels.Add(lbl);
                    startOffset1 = (startOffset1 + (int)(len1 / 4));
                    if (startOffset1 > len1)
                        startOffset1 = len1;
                    endOffset1 = (endOffset1 + (int)(len1 / 4));
                    if (endOffset1 > len1)
                        endOffset1 = len1;
                }
                //Chart2.ChartAreas[0].AxisX.LabelStyle.Angle = 90;
                Chart2.ChartAreas[0].AxisX.MajorGrid.LineWidth = 0;
                Chart2.ChartAreas[0].AxisY.MajorGrid.LineWidth = 0;
                Chart2.ChartAreas[0].AxisX.LabelStyle.Font = new System.Drawing.Font("Comic Sans", 7F,
System.Drawing.FontStyle.Bold);
                Chart2.ChartAreas[0].BackColor = System.Drawing.Color.Azure;
                Chart2.DataBind();

            }
            catch (Exception ex)
            {

            }
        }
        catch (Exception ex)
        {

        }
    }
    protected void btnSubmit_Click(object sender, EventArgs e)
    {
        getButton1Data();
        graphs.Visible = true;
    }
    protected void btnCancel_Click(object sender, EventArgs e)
    {
        Response.Redirect("Home.aspx");
    }
}
```