

Scilab Code for  
Signals and Systems  
by Alan V. Oppenheim, Alan V. Willsky,  
S.Hamid Nawab<sup>1</sup>

Created by  
Prof. R. Senthilkumar  
Institute of Road and Transport Technology  
rsenthil\_signalprocess@in.com

Cross-Checked by  
Prof. Saravanan Vijayakumaran, IIT Bombay  
sarva@ee.iitb.ac.in

18 November 2010

<sup>1</sup>Funded by a grant from the National Mission on Education through ICT, <http://spoken-tutorial.org/NMEICT-Intro>. This text book companion and Scilab codes written in it can be downloaded from the website <http://scilab.in>

# Book Details

**Author:** Alan V. Oppenheim, Alan V. Willsky, S.Hamid Nawab

**Title:** Signals and Systems

**Publisher:** Prentice-Hall India

**Edition:** Second

**Year:** 1992

**Place:** New Delhi

**ISBN:** 978-81-203-1246-3

# Contents

|  |            |
|--|------------|
| <b>List of Scilab Code</b>                                       | <b>4</b>   |
| <b>1 Signals and Systems</b>                                     | <b>11</b>  |
| 1.1 Scilab Codes . . . . .                                       | 11         |
| <b>2 Linear Time Invariant Systems</b>                           | <b>26</b>  |
| 2.1 Scilab Codes . . . . .                                       | 26         |
| <b>3 Fourier Series Repeentation of Periodic Signals</b>         | <b>54</b>  |
| 3.1 Scilab Codes . . . . .                                       | 54         |
| <b>4 The Continuous Time Fourier Transform</b>                   | <b>90</b>  |
| 4.1 Scilab Codes . . . . .                                       | 90         |
| <b>5 The Discreet Time Fourier Transform</b>                     | <b>117</b> |
| 5.1 Scilab Codes . . . . .                                       | 117        |
| <b>6 Time and Frequency Characterization of Signals and Sys-</b> |            |
| <b>tems</b>  | <b>138</b> |
| 6.1 Scilab Codes . . . . .                                       | 138        |
| <b>7 Sampling</b>  | <b>148</b> |
| 7.1 Scilab Codes . . . . .                                       | 148        |
| <b>9 The Laplace Transform</b>                                   | <b>155</b> |
| 9.1 Scilab Codes . . . . .                                       | 155        |
| <b>10 The Z-Transform</b>  | <b>171</b> |
| 10.1 Scilab Codes . . . . .                                      | 171        |

|                                   |            |
|-----------------------------------|------------|
| <b>11 Linear Feedback Systems</b> | <b>181</b> |
| 11.1 Scilab Codes . . . . .       | 181        |

# List of Scilab Code

|       |                            |    |
|-------|----------------------------|----|
| 1.1   | Example1.1.sce . . . . .   | 11 |
| 1.2   | Example1.2.sce . . . . .   | 12 |
| 1.3   | Example1.3.sce . . . . .   | 13 |
| 1.4   | Example1.4.sce . . . . .   | 14 |
| 1.5   | Example1.5.sce . . . . .   | 14 |
| 1.6   | Example1.6.sce . . . . .   | 15 |
| 1.12  | Example1.12.sce . . . . .  | 16 |
| 1.13  | Example1.13.sce . . . . .  | 17 |
| 1.13b | Example1.13b.sce . . . . . | 18 |
| 1.14  | Example1.14.sce . . . . .  | 18 |
| 1.15  | Example1.15.sce . . . . .  | 20 |
| 1.16  | Example1.16.sce . . . . .  | 21 |
| 1.17  | Example1.17.sce . . . . .  | 22 |
| 1.18  | Example1.18.sce . . . . .  | 23 |
| 1.20  | Example1.20.sce . . . . .  | 24 |
| 2.1   | Example2.1.sce . . . . .   | 26 |
| 2.3   | Example2.3.sce . . . . .   | 29 |
| 2.4   | Example2.4.sce . . . . .   | 33 |
| 2.5   | Example2.5.sce . . . . .   | 37 |
| 2.6   | Example2.6.sce . . . . .   | 41 |
| 2.7   | Example2.7.sce . . . . .   | 45 |
| 2.8   | Example2.8.sce . . . . .   | 49 |
| 3.2   | Example3.2.sce . . . . .   | 54 |
| 3.3   | Example3.3.sce . . . . .   | 60 |
| 3.4   | Example3.4.sce . . . . .   | 61 |
| 3.5   | Example3.5.sce . . . . .   | 63 |
| 3.6   | Example3.6.sce . . . . .   | 66 |
| 3.7   | Example3.7.sce . . . . .   | 68 |

|      |  |     |
|------|--|-----|
| 3.8  | Example3.8.sce . . . . .                 | 71  |
| 3.10 | Example3.10.sce . . . . .                | 74  |
| 3.11 | Example3.11.sce . . . . .                | 76  |
| 3.12 | Example3.12.sce . . . . .                | 80  |
| 3.13 | Example3.13.sce . . . . .                | 82  |
| 3.14 | Example3.14.sce . . . . .                | 85  |
| 3.15 | Example3.15.sce . . . . .                | 87  |
| 4.1  | Example4.1.sce . . . . .                 | 90  |
| 4.2  | Example4.2.sce . . . . .                 | 93  |
| 4.4  | Example4.4.sce . . . . .                 | 95  |
| 4.5  | Example4.5.sce . . . . .                 | 97  |
| 4.6  | Example4.6.sce . . . . .                 | 99  |
| 4.7  | Example4.7.sce . . . . .                 | 101 |
| 4.8  | Example4.8.sce . . . . .                 | 104 |
| 4.9  | Example4.9.sce . . . . .                 | 106 |
| 4.12 | Example4.12.sce . . . . .                | 108 |
| 4.18 | Example4.18.sce . . . . .                | 110 |
| 4.23 | Example4.23.sce . . . . .                | 112 |
| 4.22 | Impulse response of LTI system . . . . . | 114 |
| 5.1  | Example5.1.sce . . . . .                 | 117 |
| 5.2  | Example5.2.sce . . . . .                 | 121 |
| 5.3  | Example5.3.sce . . . . .                 | 123 |
| 5.5  | Example5.5.sce . . . . .                 | 125 |
| 5.6  | Example5.6.sce . . . . .                 | 127 |
| 5.7  | Example5.7.sce . . . . .                 | 129 |
| 5.9  | Example5.9.sce . . . . .                 | 131 |
| 5.12 | Example5.12.sce . . . . .                | 133 |
| 5.15 | Example5.15.sce . . . . .                | 135 |
| 6.1  | Example6.1.sce . . . . .                 | 138 |
| 6.3  | Example6.3.sce . . . . .                 | 143 |
| 6.4  | Example6.4.sce . . . . .                 | 145 |
| 6.5  | Example6.5.sce . . . . .                 | 146 |
| 7.1  | Example7.1.sce . . . . .                 | 148 |
| 7.2  | Example7.2.sce . . . . .                 | 149 |
| 7.3  | Example7.3.sce . . . . .                 | 150 |
| 7.4  | Example7.4.sce . . . . .                 | 151 |
| 7.5  | Example7.5.sce . . . . .                 | 151 |
| 9.1  | Example9.1.sce . . . . .                 | 155 |

|       |                            |     |
|-------|----------------------------|-----|
| 9.2   | Example9.2.sce . . . . .   | 155 |
| 9.3   | Example9.3.sce . . . . .   | 155 |
| 9.4   | Example9.4.sce . . . . .   | 156 |
| 9.5   | Example9.5.sce . . . . .   | 156 |
| 9.6   | Example9.6.sce . . . . .   | 156 |
| 9.7   | Example9.7.sce . . . . .   | 157 |
| 9.8   | Example9.8.sce . . . . .   | 157 |
| 9.9   | Example9.9.sce . . . . .   | 158 |
| 9.10  | Example9.10.sce . . . . .  | 159 |
| 9.11  | Example9.11.sce . . . . .  | 159 |
| 9.12  | Example9.12.sce . . . . .  | 159 |
| 9.13  | Example9.13.sce . . . . .  | 160 |
| 9.14  | Example9.14.sce . . . . .  | 163 |
| 9.15  | Example9.15.sce . . . . .  | 164 |
| 9.16  | Example9.16.sce . . . . .  | 164 |
| 9.17  | Example9.17.sce . . . . .  | 165 |
| 9.18  | Example9.18.sce . . . . .  | 165 |
| 9.19  | Example9.19.sce . . . . .  | 165 |
| 9.20  | Example9.20.sce . . . . .  | 166 |
| 9.21  | Example9.21.sce . . . . .  | 166 |
| 9.25  | Example9.25.sce . . . . .  | 166 |
| 9.31  | Example9.31.sce . . . . .  | 167 |
| 9.33  | Example9.33.sce . . . . .  | 168 |
| 9.34  | Example9.34.sce . . . . .  | 168 |
| 9.35  | Example9.35.sce . . . . .  | 168 |
| 9.36  | Example9.36.sce . . . . .  | 169 |
| 9.37  | Example9.37.sce . . . . .  | 169 |
| 9.38  | Example9.38.sce . . . . .  | 169 |
| 10.1  | Example10.1.sce . . . . .  | 171 |
| 10.2  | Example10.2.sce . . . . .  | 171 |
| 10.3  | Example10.3.sce . . . . .  | 172 |
| 10.4  | Example10.4.sce . . . . .  | 172 |
| 10.5  | Example10.5.sce . . . . .  | 172 |
| 10.6  | Example10.6.sce . . . . .  | 173 |
| 10.7  | Example10.7.sce . . . . .  | 173 |
| 10.9  | Example10.9.sce . . . . .  | 173 |
| 10.10 | Example10.10.sce . . . . . | 174 |
| 10.11 | Example10.11.sce . . . . . | 175 |

|             |  |     |
|-------------|--|-----|
| 10.12       | Example10.12.sce . . . . .                               | 175 |
| 10.13       | Example10.13.sce . . . . .                               | 175 |
| 10.18       | Example10.18.sce . . . . .                               | 176 |
| 10.19       | Example10.19.sce . . . . .                               | 176 |
| 10.23       | Example10.23.sce . . . . .                               | 177 |
| 10.25       | Example10.25.sce . . . . .                               | 177 |
| 10.33       | Example10.33.sce . . . . .                               | 178 |
| 10.34       | Example10.34.sce . . . . .                               | 178 |
| 10.36       | Example10.36.sce . . . . .                               | 179 |
| 10.37       | Example10.37.sce . . . . .                               | 180 |
| 11.1        | Example11.1.sce . . . . .                                | 181 |
| 11.2        | Example11.2.sce . . . . .                                | 182 |
| 11.3        | Example11.3.sce . . . . .                                | 183 |
| 11.5Bode    | Example11.5Bode.sce . . . . .                            | 184 |
| 11.5Nyquist | Example11.5Nyquist.sce . . . . .                         | 185 |
| 11.6        | Example11.6.sce . . . . .                                | 187 |
| 11.7        | Example11.7.sce . . . . .                                | 188 |
| 11.8        | Example11.8.sce . . . . .                                | 188 |
| 11.9        | Example11.9.sce . . . . .                                | 189 |
| 11.9        | Root locus analysis of Linear feedback systems . . . . . | 192 |



# List of Figures

|      |                               |    |
|------|-------------------------------|----|
| 1.1  | Results of Exa 1.5 . . . . .  | 15 |
| 1.2  | Results of Exa 1.14 . . . . . | 20 |
| 2.1  | Results of Exa 2.1 . . . . .  | 27 |
| 2.2  | Results of Exa 2.1 . . . . .  | 28 |
| 2.3  | Results of Exa 2.1 . . . . .  | 29 |
| 2.4  | Results of Exa 2.3 . . . . .  | 31 |
| 2.5  | Results of Exa 2.3 . . . . .  | 32 |
| 2.6  | Results of Exa 2.3 . . . . .  | 33 |
| 2.7  | Results of Exa 2.4 . . . . .  | 35 |
| 2.8  | Results of Exa 2.4 . . . . .  | 36 |
| 2.9  | Results of Exa 2.4 . . . . .  | 37 |
| 2.10 | Results of Exa 2.5 . . . . .  | 39 |
| 2.11 | Results of Exa 2.5 . . . . .  | 40 |
| 2.12 | Results of Exa 2.5 . . . . .  | 41 |
| 2.13 | Results of Exa 2.6 . . . . .  | 43 |
| 2.14 | Results of Exa 2.6 . . . . .  | 44 |
| 2.15 | Results of Exa 2.6 . . . . .  | 45 |
| 2.16 | Results of Exa 2.7 . . . . .  | 47 |
| 2.17 | Results of Exa 2.7 . . . . .  | 48 |
| 2.18 | Results of Exa 2.7 . . . . .  | 49 |
| 2.19 | Results of Exa 2.8 . . . . .  | 51 |
| 2.20 | Results of Exa 2.8 . . . . .  | 52 |
| 2.21 | Results of Exa 2.8 . . . . .  | 53 |
| 3.1  | Results of Exa 3.2 . . . . .  | 57 |
| 3.2  | Results of Exa 3.2 . . . . .  | 58 |
| 3.3  | Results of Exa 3.2 . . . . .  | 59 |
| 3.4  | Results of Exa 3.2 . . . . .  | 60 |

|      |                               |     |
|------|-------------------------------|-----|
| 3.5  | Results of Exa 3.4 . . . . .  | 63  |
| 3.6  | Results of Exa 3.5 . . . . .  | 65  |
| 3.7  | Results of Exa 3.5 . . . . .  | 66  |
| 3.8  | Results of Exa 3.6 . . . . .  | 68  |
| 3.9  | Results of Exa 3.7 . . . . .  | 70  |
| 3.10 | Results of Exa 3.7 . . . . .  | 71  |
| 3.11 | Results of Exa 3.8 . . . . .  | 74  |
| 3.12 | Results of Exa 3.10 . . . . . | 76  |
| 3.13 | Results of Exa 3.11 . . . . . | 79  |
| 3.14 | Results of Exa 3.11 . . . . . | 80  |
| 3.15 | Results of Exa 3.12 . . . . . | 82  |
| 3.16 | Results of Exa 3.13 . . . . . | 85  |
| 3.17 | Results of Exa 3.14 . . . . . | 87  |
| 3.18 | Results of Exa 3.15 . . . . . | 89  |
| 4.1  | Results of Exa 4.1 . . . . .  | 92  |
| 4.2  | Results of Exa 4.1 . . . . .  | 93  |
| 4.3  | Results of Exa 4.2 . . . . .  | 95  |
| 4.4  | Results of Exa 4.4 . . . . .  | 97  |
| 4.5  | Results of Exa 4.5 . . . . .  | 99  |
| 4.6  | Results of Exa 4.6 . . . . .  | 101 |
| 4.7  | Results of Exa 4.7 . . . . .  | 103 |
| 4.8  | Results of Exa 4.7 . . . . .  | 104 |
| 4.9  | Results of Exa 4.8 . . . . .  | 106 |
| 4.10 | Results of Exa 4.9 . . . . .  | 108 |
| 4.11 | Results of Exa 4.12 . . . . . | 110 |
| 4.12 | Results of Exa 4.18 . . . . . | 112 |
| 4.13 | Results of Exa 4.23 . . . . . | 113 |
| 4.14 | Results of Exa 4.22 . . . . . | 115 |
| 4.15 | Results of Exa 4.22 . . . . . | 116 |
| 5.1  | Results of Exa 5.1 . . . . .  | 120 |
| 5.2  | Results of Exa 5.1 . . . . .  | 121 |
| 5.3  | Results of Exa 5.2 . . . . .  | 123 |
| 5.4  | Results of Exa 5.3 . . . . .  | 125 |
| 5.5  | Results of Exa 5.5 . . . . .  | 127 |
| 5.6  | Results of Exa 5.6 . . . . .  | 129 |
| 5.7  | Results of Exa 5.7 . . . . .  | 131 |

|       |                                      |     |
|-------|--------------------------------------|-----|
| 5.8   | Results of Exa 5.9 . . . . .         | 133 |
| 5.9   | Results of Exa 5.12 . . . . .        | 135 |
| 5.10  | Results of Exa 5.15 . . . . .        | 137 |
| 6.1   | Results of Exa 6.1 . . . . .         | 141 |
| 6.2   | Results of Exa 6.1 . . . . .         | 142 |
| 6.3   | Results of Exa 6.1 . . . . .         | 143 |
| 6.4   | Results of Exa 6.3 . . . . .         | 145 |
| 6.5   | Results of Exa 6.4 . . . . .         | 146 |
| 6.6   | Results of Exa 6.5 . . . . .         | 147 |
| 7.1   | Results of Exa 7.1 . . . . .         | 149 |
| 7.2   | Results of Exa 7.5 . . . . .         | 154 |
| 9.1   | Results of Exa 9.8 . . . . .         | 158 |
| 9.2   | Results of Exa 9.13 . . . . .        | 161 |
| 9.3   | Results of Exa 9.13 . . . . .        | 162 |
| 9.4   | Results of Exa 9.13 . . . . .        | 163 |
| 11.1  | Results of Exa 11.1 . . . . .        | 182 |
| 11.2  | Results of Exa 11.2 . . . . .        | 183 |
| 11.3  | Results of Exa 11.3 . . . . .        | 184 |
| 11.4  | Results of Exa 11.5Bode . . . . .    | 185 |
| 11.5  | Results of Exa 11.5Nyquist . . . . . | 186 |
| 11.6  | Results of Exa 11.6 . . . . .        | 187 |
| 11.7  | Results of Exa 11.8 . . . . .        | 189 |
| 11.8  | Results of Exa 11.9 . . . . .        | 191 |
| 11.9  | Results of Exa 11.9 . . . . .        | 192 |
| 11.10 | Results of Exa 11.9 . . . . .        | 193 |

# Chapter 1

## Signals and Systems

### 1.1 Scilab Codes

#### Example 1.1 Time Shifting

```
1 //Example 1.1: Time Shifting
2 //SIGNALS & SYSTEMS, Second Edition
3 //V.OPPENHEIM, S.WILLSKY, S.HAMID NAMWAB
4 //PHI, 2008 Edition
5 //Page 10
6 clear all;
7 clc;
8 close;
9 t = 0:1/100:1;
10 for i = 1:length(t)
11     x(i) = 1 ;
12 end
13 for i = length(t)+1:2*length(t)
14     x(i) = 1-t(i-length(t));
15 end
16 t1 = 0:1/100:2;
17 t2 = -1:1/100:1;
18 //t3 = 0:1/100:4/3;
19 //t4 = 0:1/length(t3):1;
20 //Mid =ceil(length(t3)/2);
```

```

21 //for i = 1:Mid
22 //  x3(i) = 1 ;
23 //end
24 //for i = Mid+1:length(t3)
25 //  x3(i) = 1-t4(i-Mid);
26 //end
27 figure
28 a=gca();
29 plot2d(t1,x(1:$-1))
30 a.thickness=2;
31 xtitle('The signal x(t)')
32 figure
33 a=gca();
34 plot2d(t2,x(1:$-1))
35 a.thickness=2;
36 a.y_location = "middle";
37 xtitle('The signal x(t+1)')
38 figure
39 a=gca();
40 plot2d(t2,x($:-1:2))
41 a.thickness=2;
42 a.y_location = "middle";
43 xtitle('The signal x(-t+1)')

```

---

### Example 1.2 Time Scaling

```

1 //Example 1.2:Time Scaling
2 //SIGNALS & SYSTEMS, Second Edition
3 //V.OPPENHEIM, S.WILLSKY, S.HAMID NAMWAB
4 //PHI, 2008 Edition
5 //Page 11
6 clear all;
7 clc;
8 close;
9 t3 = 0:1/100:4/3;
10 t4 = 0:1/length(t3):1;
11 Mid =ceil(length(t3)/2);
12 for i = 1:Mid

```

```

13     x3(i) = 1 ;
14 end
15 for i = Mid+1:length(t3)
16     x3(i) = 1-t4(i-Mid);
17 end
18 figure
19 a=gca();
20 plot2d(t3,x3)
21 a.thickness=2;
22 xtitle('Time Scaling x(3t/2)')

```

---

### Example 1.3 Time Scaling and Time Shifting

```

1 //Example 1.3:Time Scaling and Time Shifting
2 //SIGNALS & SYSTEMS, Second Edition
3 //V.OPPENHEIM, S.WILLSKY, S.HAMID NAMWAB
4 //PHI, 2008 Edition
5 //Page 11
6 clear all;
7 clc;
8 close;
9 t3 = 0:1/100:4/3;
10 t4 = 0:1/length(t3):1;
11 Mid =ceil(length(t3)/2);
12 for i = 1:Mid
13     x3(i) = 1 ;
14 end
15 for i = Mid+1:length(t3)
16     x3(i) = 1-t4(i-Mid);
17 end
18 t5 = -2/3:1/100:2/3;
19 figure
20 a=gca();
21 plot2d(t5,x3)
22 a.thickness=2;
23 a.y_location = "middle";
24 xtitle('Time Scaling and Time Shifting x((3t/2)+1)')

```

---

**Example 1.4** Combination two periodic signals Aperiodic signal

```
1 //Example 1.4:Combination two periodic signals
2 // Aperiodic signal
3 //Page 12
4 clear all;
5 clc;
6 close;
7 F=1; //Frequency = 1 Hz
8 t1 = 0:-1/100:-2*%pi;
9 x1 = cos(F*t1);
10 t2 = 0:1/100:2*%pi;
11 x2 = sin(F*t2);
12 a=gca();
13 plot(t2,x2);
14 plot(t1,x1);
15 a.y_location = "middle";
16 a.x_location = "middle";
17 xtitle('The signal  $x(t) = \cos t$  for  $t < 0$  and  $\sin t$ 
        for  $t > 0$ : Aperiodic Signal')
```

---

**Example 1.5** sum of two complex exponentials as a single sinusoid

```
1 //Example 1.5:To express sum of two complex
    exponentials
2 //as a single sinusoid
3 clear all;
4 clc;
5 close;
6 t =0:1/100:2*%pi;
7 x1 = exp(sqrt(-1)*2*t);
8 x2 = exp(sqrt(-1)*3*t);
9 x = x1+x2;
10 for i = 1:length(x)
11     X(i) = sqrt((real(x(i)).^2)+(imag(x(i)).^2));
12 end
13 plot(t,X);
```

```
14 xtitle('Full wave rectified sinusoid','time t','  
    Magnitude');
```

---

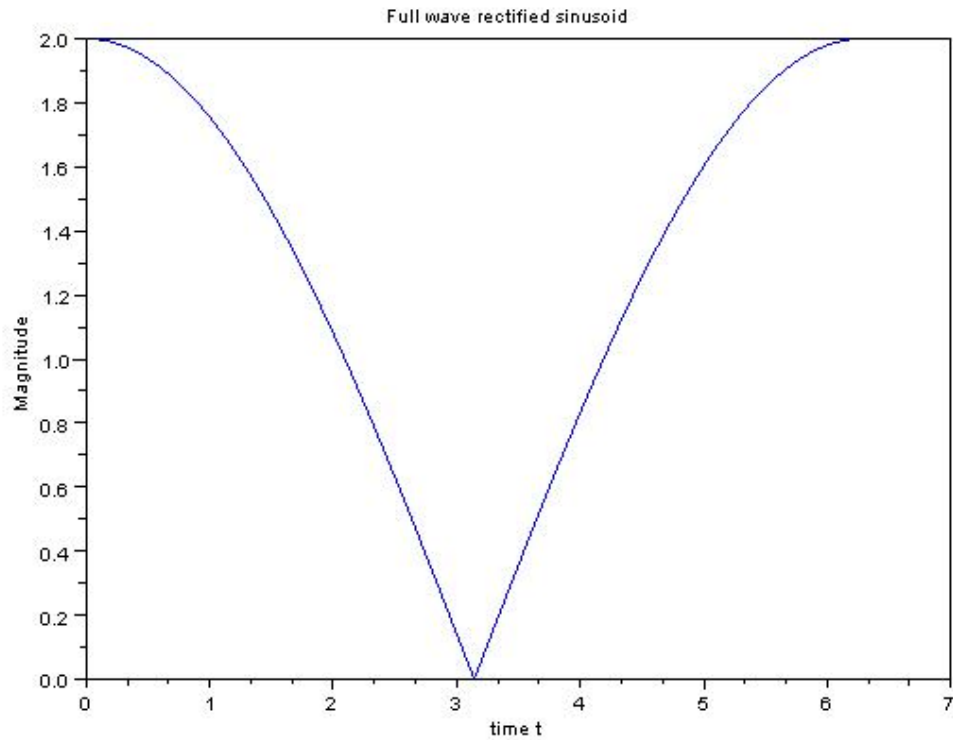


Figure 1.1: Results of Exa 1.5

**Example 1.6** Determining the fundamental period of composite discrete time signal

```
1 //Example 1.6:Determine the fundamental period of  
    composite  
2 // discrete time signal  
3 //x[n] = exp(j(2*%pi/3)n)+exp(j(3*%pi/4)n)  
4 clear all;
```



```

5  clc;
6  close;
7  Omega1 = 2*%pi/3;    //Angular frequency signal 1
8  Omega2 = 3*%pi/4;    //Angular frequency signal 2
9  N1 = (2*%pi)/Omega1; //Peirod of signal 1
10 N2 = (2*%pi)/Omega2; //Period of signal 2
11 //To find rational period of signal 1
12 for m1 = 1:100
13     period = N1*m1;
14     if(modulo(period,1)==0)
15         period1 = period;
16         integer_value = m1
17         break;
18     end
19 end
20 //To find rational period of signal 2
21 for m2 = 1:100
22     period = N2*m2;
23     if(modulo(period,1)==0)
24         period2 = period;
25         integer_value = m2
26         break;
27     end
28 end
29 disp(period1)
30 disp(period2)
31 //To determine the fundamental period N
32 N = period1*period2

```

---

**Example 1.12** Classification of system:Causality property

```

1  //Example 1.12: Classification of system:Causality
   property
2  //Page 47
3  //To check whether the given discrete system is a
   Causal System (or) Non-Causal System
4  //Given discrete system  $y[n] = x[-n]$ 
5  clear;

```

```

6  clc;
7  x = [2,4,6,8,10,0,0,0,1]; //Assign some value to
    input
8  n = -length(x)/2:length(x)/2;
9  count = 0;
10 mid = ceil(length(x)/2);
11 y = zeros(1,length(x));
12 y(mid+1:$) = x($:-1:mid+1);
13 for n = -1:-1:-mid
14     y(n+1+mid) = x(-n);
15 end
16 for i = 1:length(x)
17     if (y(i)==x(i))
18         count = count+1;
19     end
20 end
21 if (count==length(x))
22     disp('The given system is a causal system')
23 else
24     disp('Since it depends on future input value')
25     disp('The given system is a non-causal system')
26 end

```

---

**Example 1.13** Determination of stability of a given system

```

1  //Example 1.13:Determination of stability of a
    given system
2  //Page 49
3  //given system  $y(t) = t.x(t)$ 
4  clear;
5  clc;
6  x = [1,2,3,4,0,2,1,3,5,8]; //Assign some input
7  Maximum_Limit = 10;
8  S = 0;
9  for t = 0:Maximum_Limit-1
10     S = S+t*x(t+1);
11 end
12 if (S >Maximum_Limit)

```

```

13     disp('Eventhough input is bounded output is
          unbounded')
14     disp('The given system is unstable');
15     disp('S =');
16     S
17 else
18     disp('The given system is stable');
19     disp('The value of S =');
20     S
21 end

```

---

**Example 1.13b** Determination of stability of a given system

```

1 //Example 1.13(b):Determination of stability of a
  given system
2 //Page 50
3 //given system  $y(t) = \exp(x(t))$ 
4 clear;
5 clc;
6 Maximum_Limit = 10;
7 S = 0;
8 for t = 0:Maximum_Limit-1
9     x(t+1)= -2^t;           //Input some bounded value
10    S = S+exp(x(t+1));
11 end
12 if (S >Maximum_Limit)
13     disp('Eventhough input is bounded output is
          unbounded')
14     disp('The given system is unstable');
15     disp('S =');
16     S
17 else
18     disp('The given system is stable');
19     disp(S);
20 end

```

---

**Example 1.14** Classification of a system:Time Invariance Property

```

1 //Example 1.14:classification of a system:Time
  Invariance Property
2 //Page 51
3 //To check whether the given system is a Time
  variant (or) Time In-variant
4 // The given discrete signal is  $y(t) = \sin(x(t))$ 
5 clear;
6 clc;
7 to = 2; //Assume the amount of time shift =2
8 T = 10; //Length of given signal
9 for t = 1:T
10     x(t) = (2*%pi/T)*t;
11     y(t) = sin(x(t));
12 end
13 //First shift the input signal only
14 Input_shift = sin(x(T-to));
15 Output_shift = y(T-to);
16 if(Input_shift == Output_shift)
17     disp('The given discrete system is a Time In-
        variant system');
18 else
19     disp('The given discrete system is a Time Variant
        system');
20 end

```

---

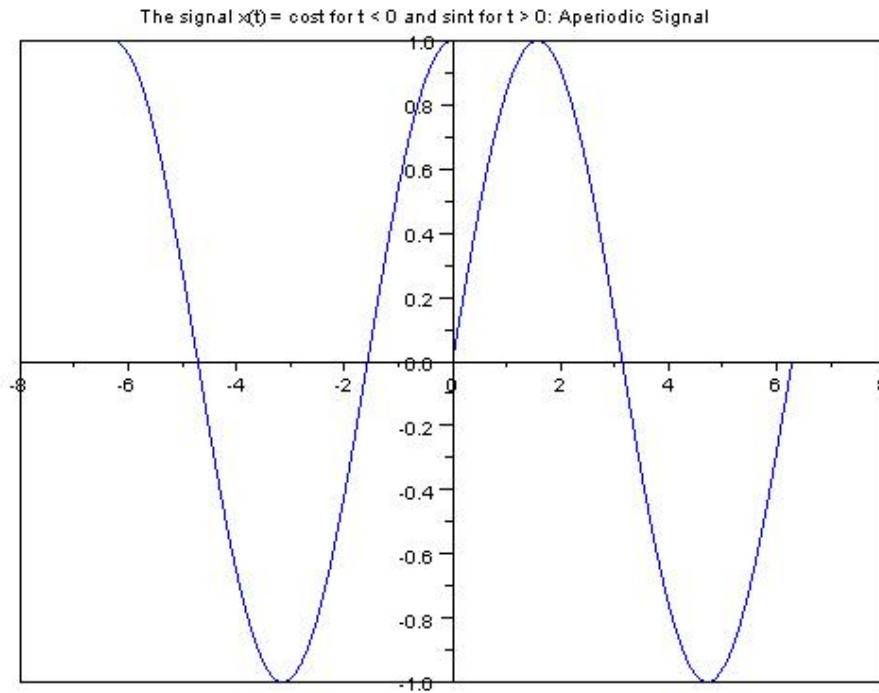


Figure 1.2: Results of Exa 1.14

**Example 1.15** Classification of a System:Time Invariance Property

```

1 //Example 1.15: Classification of a System:Time
  Invariance Property
2 //Page 51
3 //To check whether the given system is a Time
  variant (or) Time In-variant
4 // The given discrete signal is  $y[n] = n \cdot x[n]$ 
5 clear;
6 clc;
7 no = 2; //Assume the amount of time shift =2
8 L = 10; //Length of given signal

```

```

9  for n = 1:L
10     x(n) = n;
11     y(n) = n*x(n);
12 end
13 //First shift the input signal only
14 Input_shift = x(L-no);
15 Output_shift = y(L-no);
16 if(Input_shift == Output_shift)
17     disp('The given discrete system is a Time In-
        variant system');
18 else
19     disp('The given discrete system is a Time Variant
        system');
20 end

```

---

**Example 1.16** Classification of system:Time Invariance Property

```

1  //Example 1.16: Classification of system:Time
    Invariance Property
2  //Page 52
3  //To check whether the given system is a Time
    variant (or) Time In-variant
4  // The given discrete signal is  $y(t) = x(2t)$ 
5  clear;
6  clc;
7  to = 2; //Assume the amount of time shift =2
8  T = 10; //Length of given signal
9  x = [1,2,3,4,5,6,7,8,9,10];
10 y = zeros(1,length(x));
11 for t = 1:length(x)/2
12     y(t) = x(2*t);
13 end
14 //First shift the input signal only
15 Input_shift = x(T-to);
16 Output_shift = y(T-to);
17 if(Input_shift == Output_shift)
18     disp('The given discrete system is a Time In-
        variant system');

```

```

19 else
20     disp('The given discrete system is a Time Variant
        system');
21 end

```

---

**Example 1.17** Classification of system:Linearity Property

```

1 //Example 1.17: Classification of system:Linearity
  Property
2 //Page 54
3 //To check whether the given discrete system is a
  Linear System (or) Non-Linear System
4 //Given discrete system  $y(t) = t \cdot x(t)$ 
5 clear;
6 clc;
7 x1 = [1,1,1,1];
8 x2 = [2,2,2,2];
9 a = 1;
10 b = 1;
11 for t = 1:length(x1)
12     x3(t) = a*x1(t)+b*x2(t);
13 end
14 for t = 1:length(x1)
15     y1(t) = t*x1(t);
16     y2(t) = t*x2(t);
17     y3(t) = t*x3(t);
18 end
19 for t = 1:length(y1)
20     z(t) = a*y1(t)+b*y2(t);
21 end
22 count = 0;
23 for n =1:length(y1)
24     if(y3(t)== z(t))
25         count = count+1;
26     end
27 end
28 if(count == length(y3))

```

```

29     disp('Since It satisfies the superposition
        principle')
30     disp('The given system is a Linear system')
31     y3
32     z
33 else
34     disp('Since It does not satisfy the
        superposition principle')
35     disp('The given system is a Non-Linear system')
36 end

```

---

**Example 1.18** Classsification of a system:Linearity Property

```

1 //Example 1.18: Classsification of a system:Linearity
  Property
2 //Page 54
3 //To check whether the given discrete system is a
  Linear System (or) Non-Linear System
4 //Given discrete system  $y(t) = (x(t))^2$ 
5 clear;
6 clc;
7 x1 = [1,1,1,1];
8 x2 = [2,2,2,2];
9 a = 1;
10 b = 1;
11 for t = 1:length(x1)
12     x3(t) = a*x1(t)+b*x2(t);
13 end
14 for t = 1:length(x1)
15     y1(t) = (x1(t)^2);
16     y2(t) = (x2(t)^2);
17     y3(t) = (x3(t)^2);
18 end
19 for t = 1:length(y1)
20     z(t) = a*y1(t)+b*y2(t);
21 end
22 count = 0;
23 for n =1:length(y1)

```



```

24     if(y3(t)== z(t))
25         count = count+1;
26     end
27 end
28 if(count == length(y3))
29     disp('Since It satisfies the superposition
        principle')
30     disp('The given system is a Linear system')
31     y3
32     z
33 else
34     disp('Since It does not satisfy the
        superposition principle')
35     disp('The given system is a Non-Linear system')
36 end

```

---

**Example 1.20** Classsification of a system:Linearity Property

```

1 //Example 1.20: Classsification of a system:Linearity
  Property
2 //Page 55
3 //To check whether the given discrete system is a
  Linear System (or) Non-Linear System
4 //Given discrete system  $y[n] = 2x[n] + 3$ 
5 clear;
6 clc;
7 x1 = [1,1,1,1];
8 x2 = [2,2,2,2];
9 a = 1;
10 b = 1;
11 for n = 1:length(x1)
12     x3(n) = a*x1(n)+b*x2(n);
13 end
14 for n = 1:length(x1)
15     y1(n) = 2*x1(n)+3;
16     y2(n) = 2*x2(n)+3;
17     y3(n) = 2*x3(n)+3;
18 end

```

```

19 for n = 1:length(y1)
20     z(n) = a*y1(n)+b*y2(n);
21 end
22 count = 0;
23 for n =1:length(y1)
24     if(y3(n)== z(n))
25         count = count+1;
26     end
27 end
28 if(count == length(y3))
29     disp('Since It satisfies the superposition
        principle')
30     disp('The given system is a Linear system')
31     y3
32     z
33 else
34     disp('Since It does not satisfy the
        superposition principle')
35     disp('The given system is a Non-Linear system')
36 end

```

---

# Chapter 2

## Linear Time Invariant Systems

### 2.1 Scilab Codes

**Example 2.1** Linear Convolution Sum

```
1 //Example 2.1:Linear Convolution Sum
2 //page 80
3 clear all;
4 close;
5 clc;
6 h = [0,0,1,1,1,0,0];
7 N1 = -2:4;
8 x = [0,0,0.5,2,0,0,0];
9 N2 = -2:4;
10 y = convol(x,h);
11 for i = 1:length(y)
12     if (y(i)<=0.0001)
13         y(i)=0;
14     end
15 end
16 N = -4:8;
17 figure
18 a=gca();
19 plot2d3('gnn',N1,h)
20 xtitle('Impulse Response','n','h[n]');
```

```

21 a.thickness = 2;
22 figure
23 a=gca();
24 plot2d3('gnn',N2,x)
25 xtitle('Input Response','n','x[n]');
26 a.thickness = 2;
27 figure
28 a=gca();
29 plot2d3('gnn',N,y)
30 xtitle('Output Response','n','y[n]');
31 a.thickness = 2;

```

---

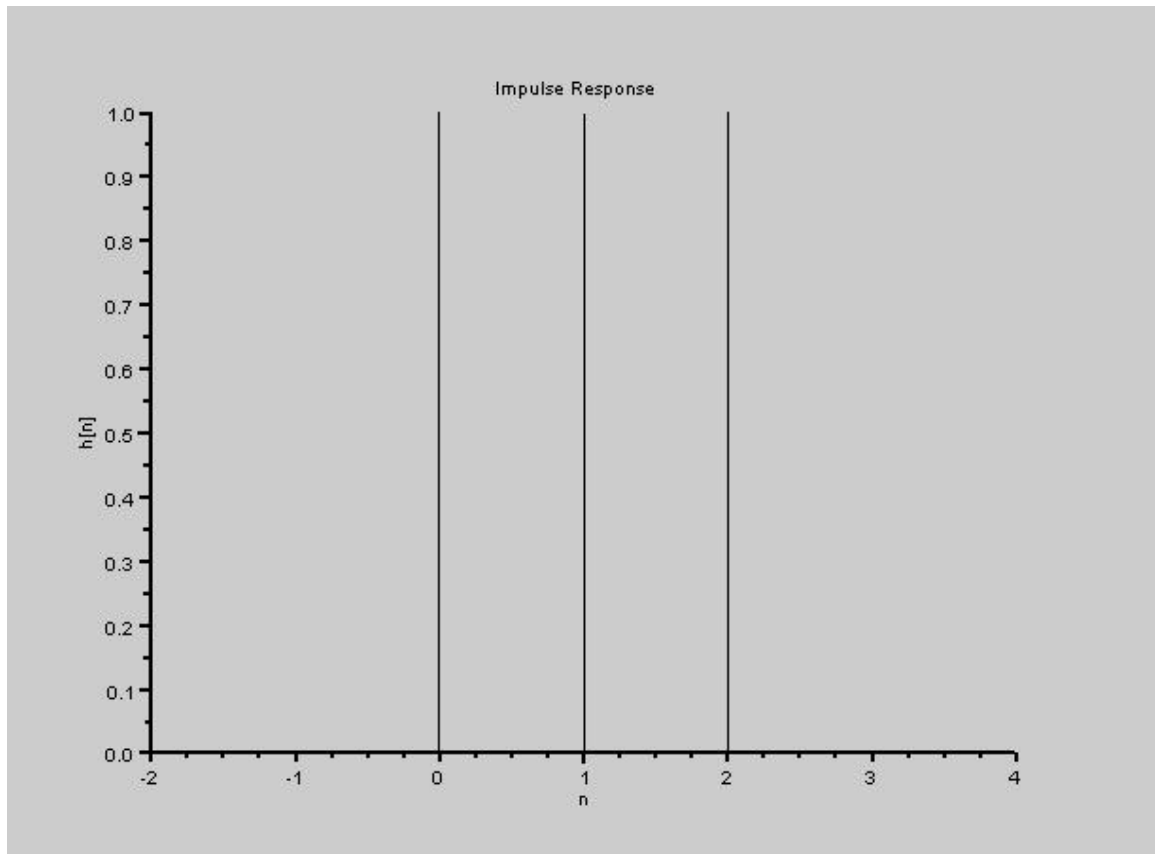


Figure 2.1: Results of Exa [2.1](#)

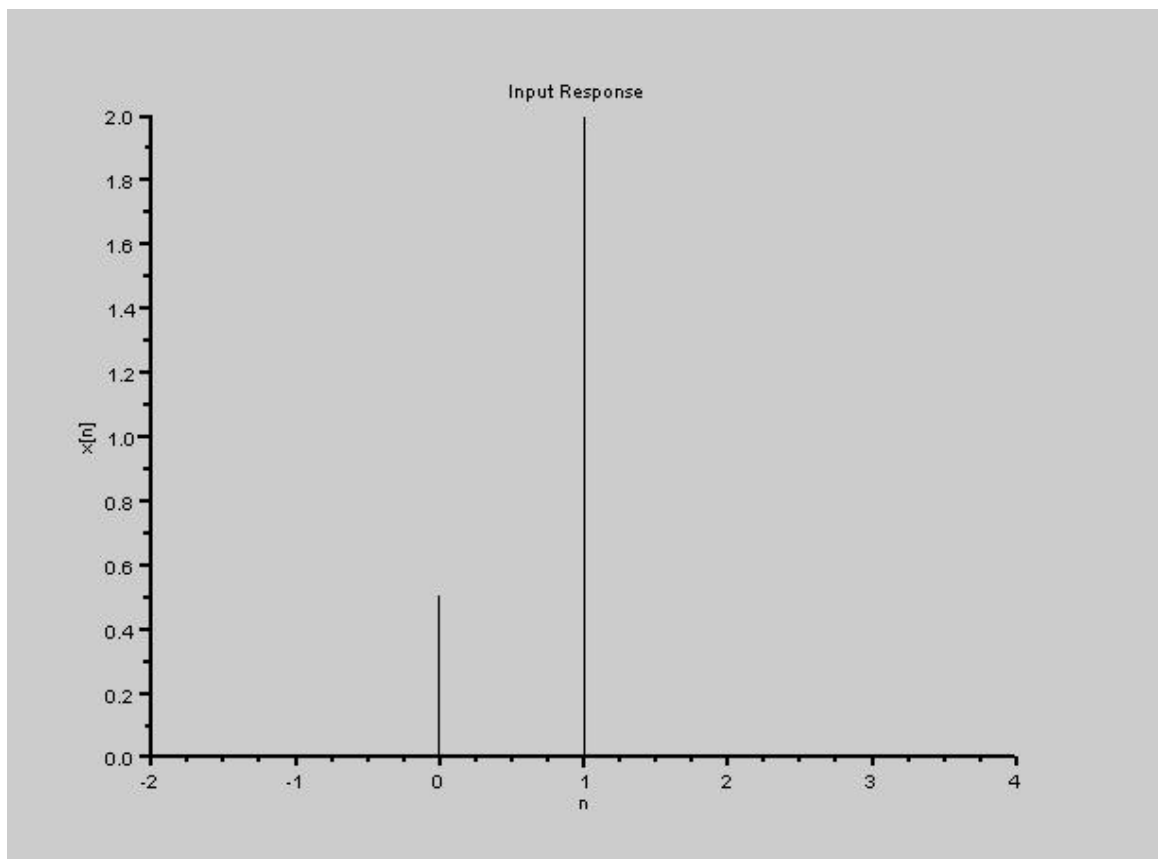


Figure 2.2: Results of Exa [2.1](#)

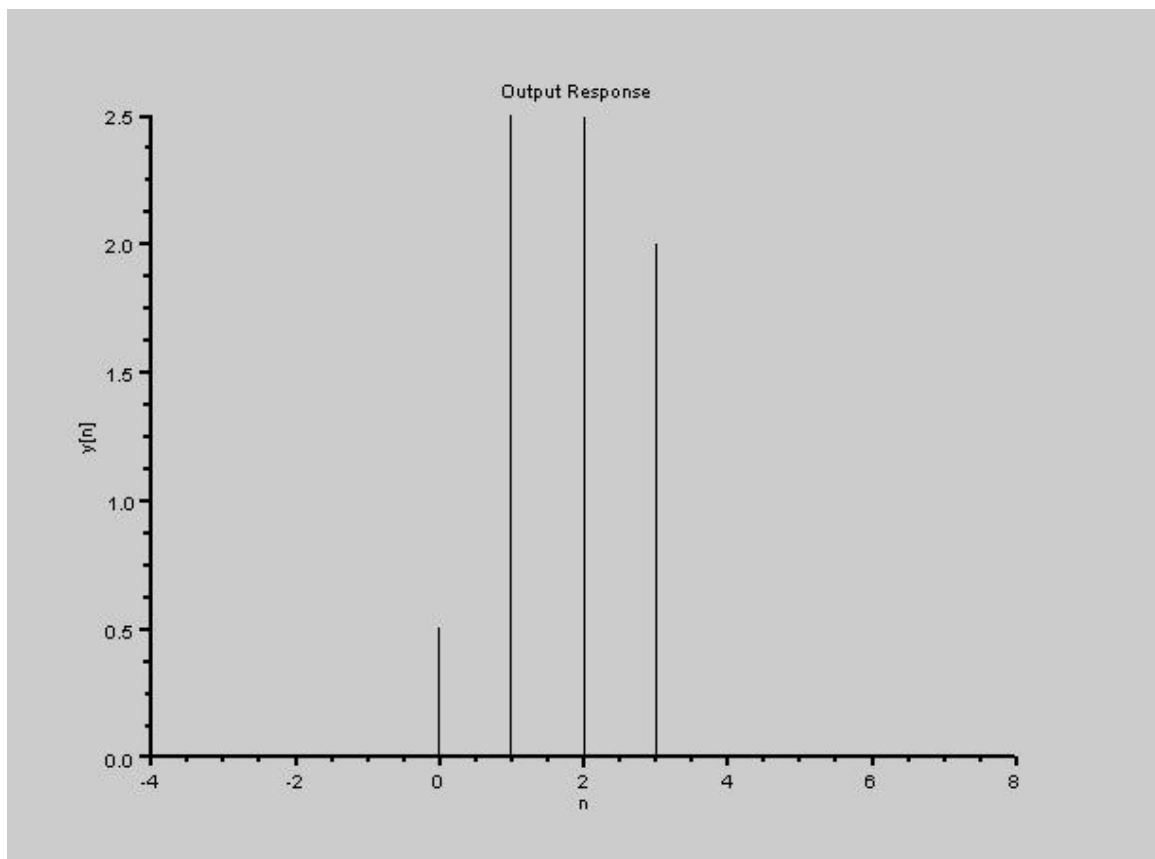


Figure 2.3: Results of Exa 2.1

**Example 2.3** Convolution Sum: Convolution of  $x[n]$  and Unit Impulse response  $h[n]$

```

1 //Example 2.3: Convolution Sum: Convolution of x[n]
  and
2 //Unit Impulse response h[n]
3 clear;
4 close;
5 clc;
6 Max_Limit = 10;
7 h = ones(1,Max_Limit);
8 N1 = 0:Max_Limit-1;

```

```

 9 Alpha = 0.5;      //alpha < 1
10 for n = 1:Max_Limit
11     x(n)= (Alpha^(n-1))*1;
12 end
13 N2 = 0:Max_Limit-1;
14 y = convol(x,h);
15 N = 0:2*Max_Limit-2;
16 figure
17 a=gca();
18 plot2d3('gnn',N1,h)
19 xtitle('Impulse Response Fig 2.5.(b)', 'n', 'h[n]');
20 a.thickness = 2;
21 figure
22 a=gca();
23 plot2d3('gnn',N2,x)
24 xtitle('Input Response Fig 2.5.(a)', 'n', 'x[n]');
25 a.thickness = 2;
26 figure
27 a=gca();
28 plot2d3('gnn',N(1:Max_Limit),y(1:Max_Limit),5)
29 xtitle('Output Response Fig 2.7', 'n', 'y[n]');
30 a.thickness = 2;

```

---

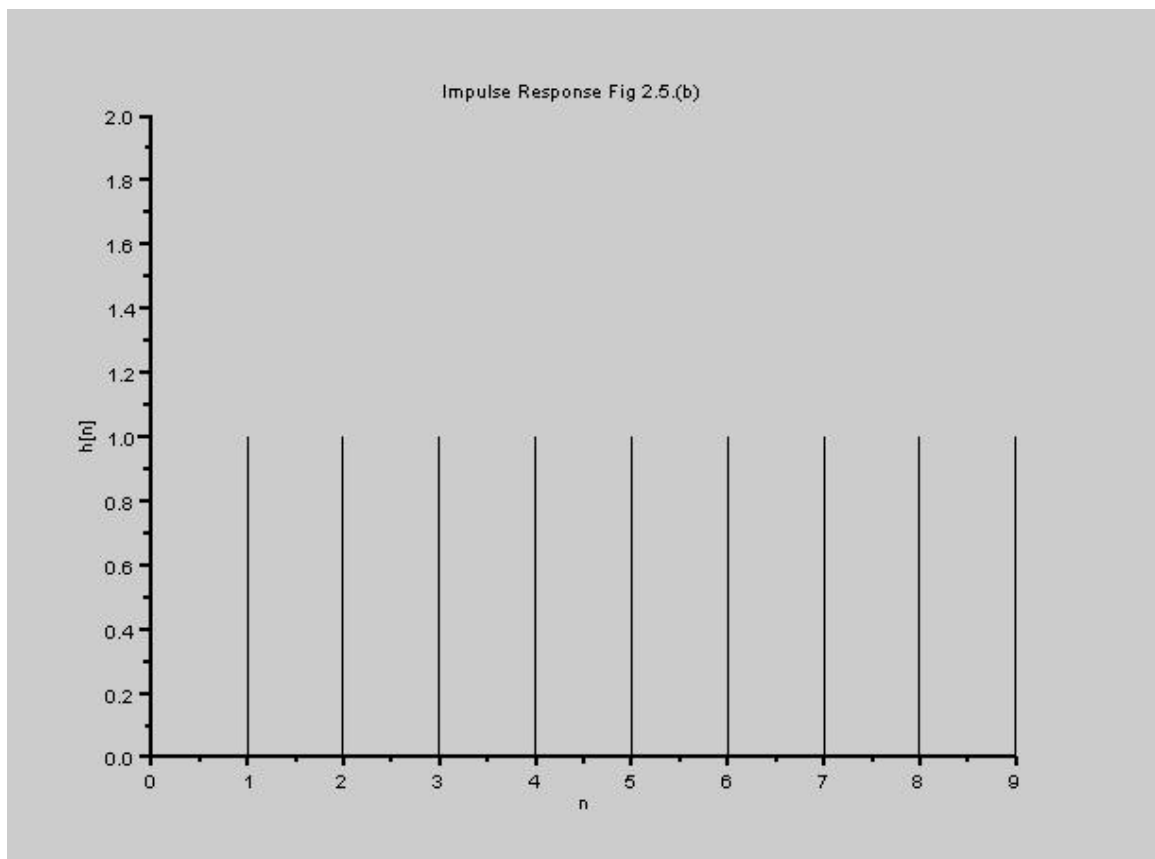


Figure 2.4: Results of Exa [2.3](#)



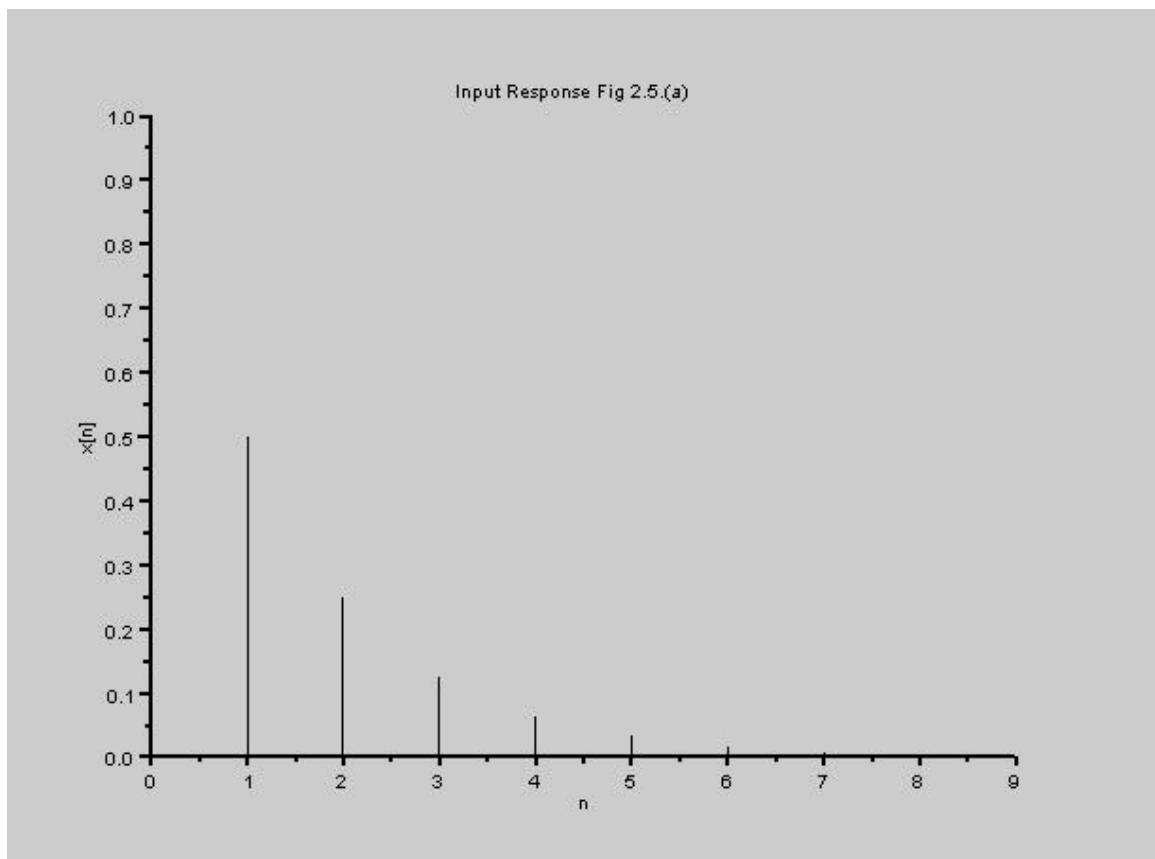


Figure 2.5: Results of Exa [2.3](#)

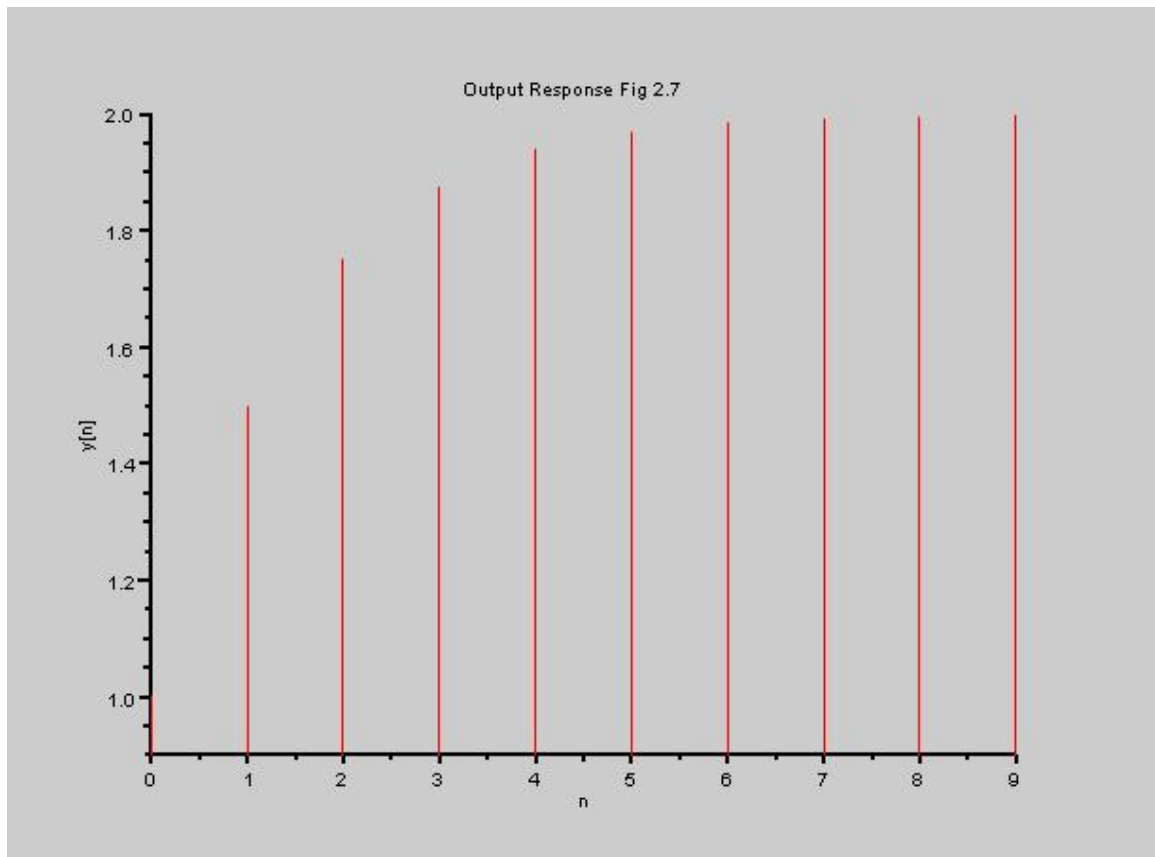


Figure 2.6: Results of Exa 2.3

**Example 2.4** Convolution Sum of finite duration sequences

```

1 //Example 2.4: Convolution Sum of finite duration
  sequences
2 clear;
3 close;
4 clc;
5 x = ones(1,5);
6 N1 = 0:length(x)-1;
7 Alpha = 1.4;    //alpha > 1
8 for n = 1:7
9     h(n) = (Alpha^(n-1))*1;

```

```

10 end
11 N2 = 0:length(h)-1;
12 y = convol(x,h);
13 N = 0:length(x)+length(h)-2;
14 figure
15 a=gca();
16 plot2d3('gnn',N2,h)
17 xtitle('Impulse Response','n','h[n]');
18 a.thickness = 2;
19 figure
20 a=gca();
21 plot2d3('gnn',N1,x)
22 xtitle('Input Response','n','x[n]');
23 a.thickness = 2;
24 figure
25 a=gca();
26 plot2d3('gnn',N,y)
27 xtitle('Output Response','n','y[n]');
28 a.thickness = 2;

```

---

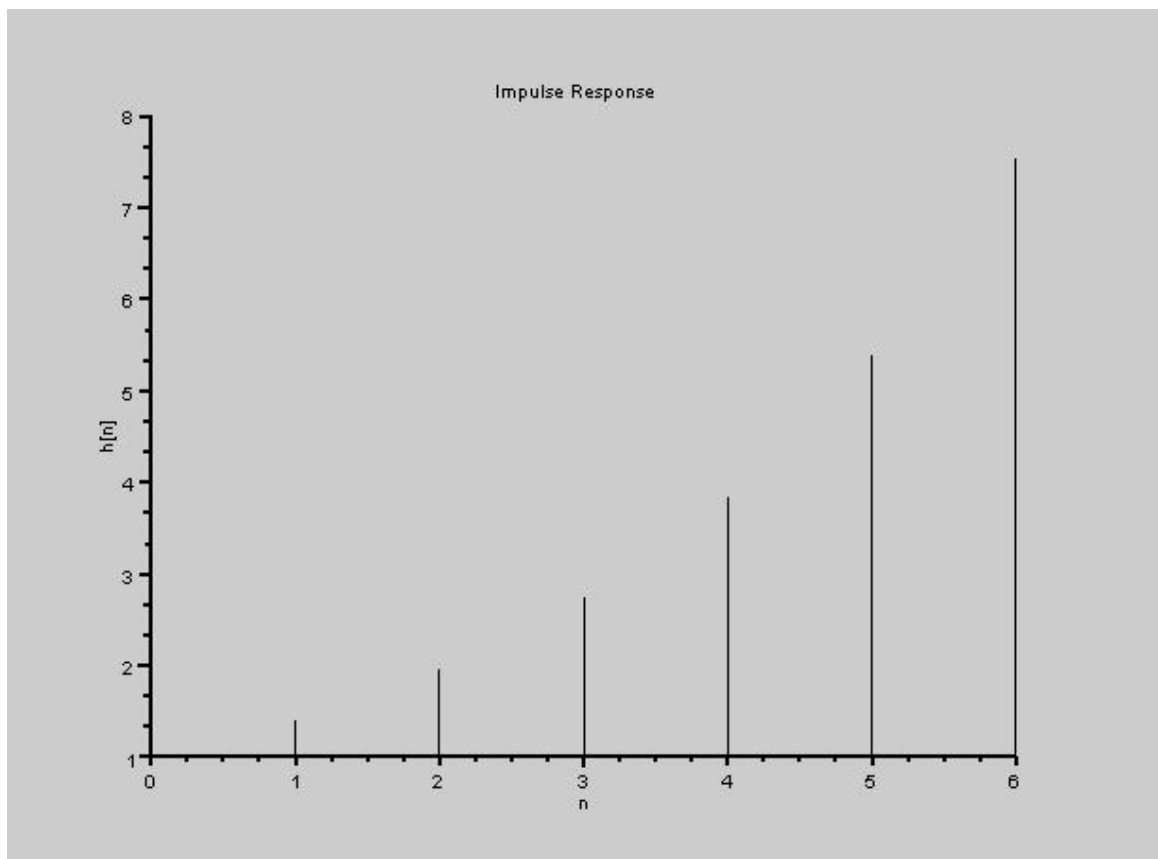


Figure 2.7: Results of Exa [2.4](#)

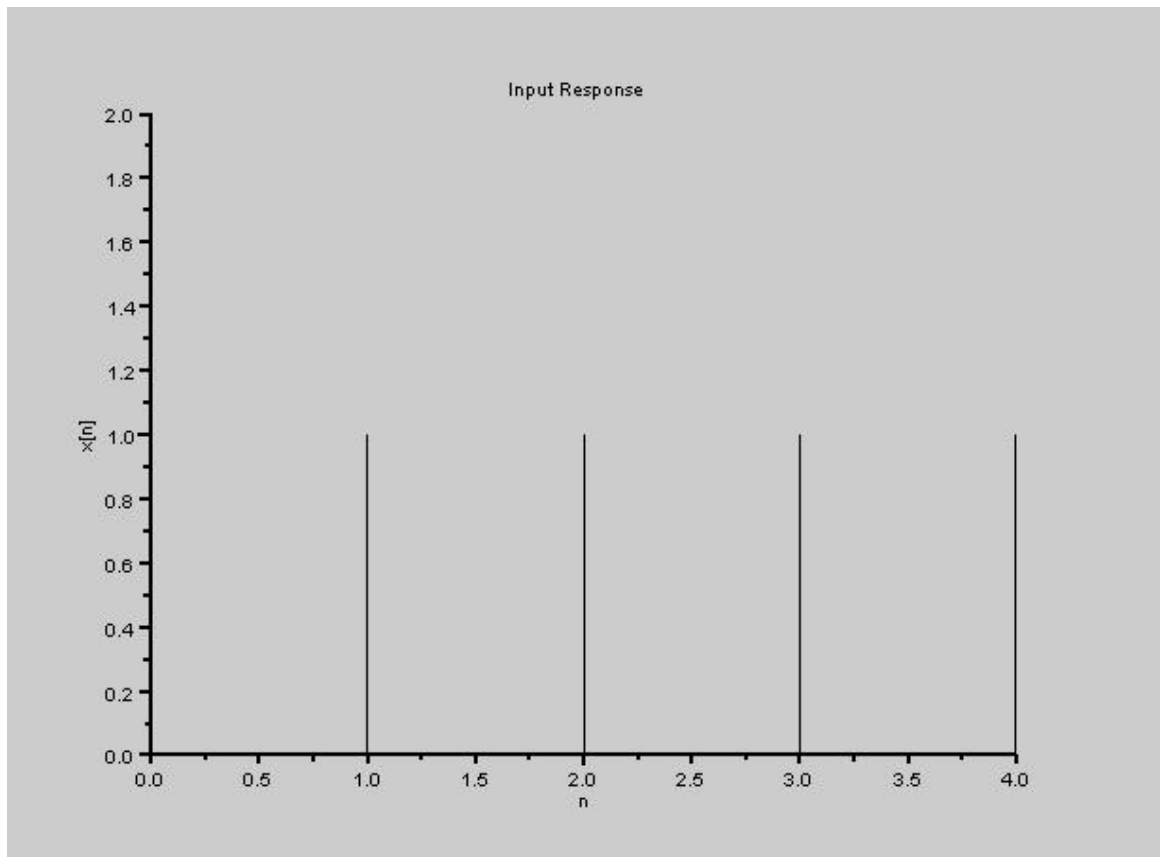


Figure 2.8: Results of Exa [2.4](#)

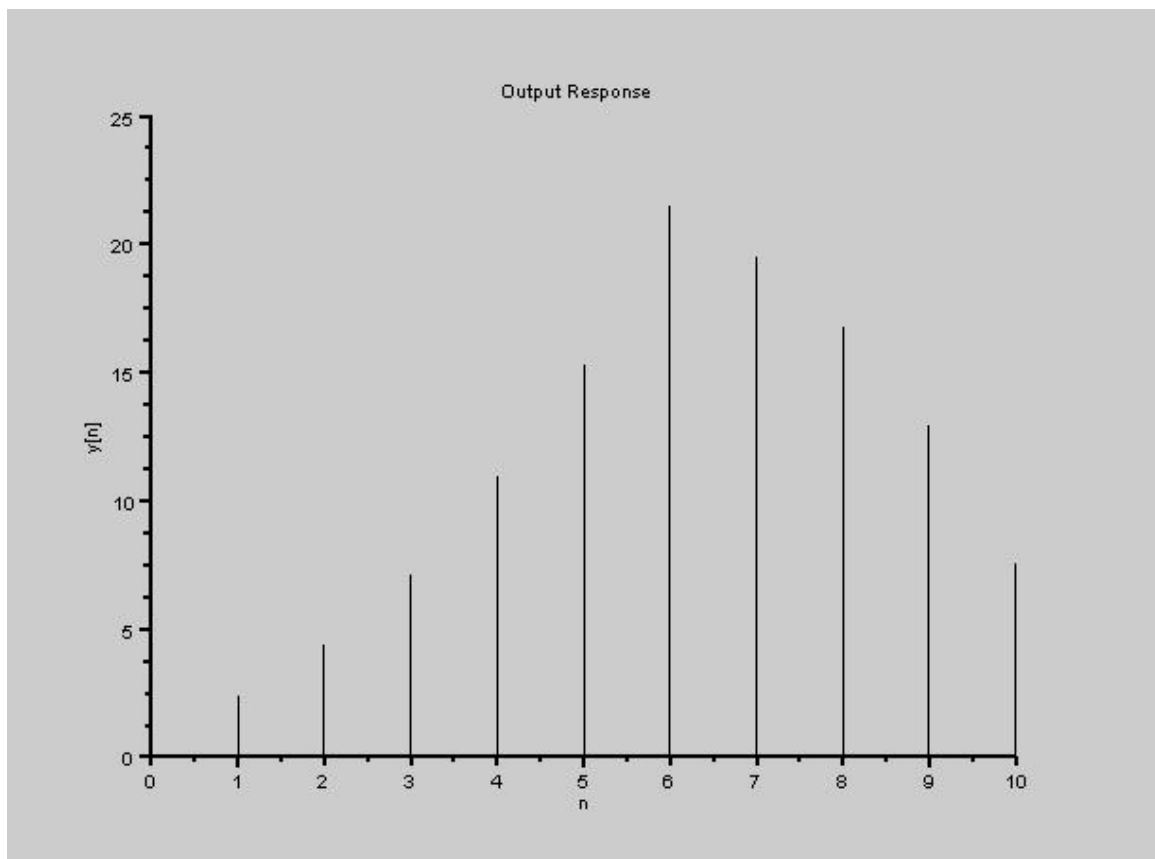


Figure 2.9: Results of Exa 2.4

**Example 2.5** Convolution Sum of input sequence  $x[n] = (2^n).u[-n]$  and  $h[n] = u[n]$

```

1 //Example 2.5: Convolution Sum of input sequence x[n
    ]=(2^n).u[-n]
2 //and h[n] = u[n]
3 clear;
4 close;
5 clc;
6 Max_Limit = 10;
7 h = ones(1,Max_Limit);
8 N2 =0:length(h)-1;

```

```

9  for n = 1:Max_Limit
10     x1(n)= (2^(-(n-1)))*1;
11 end
12 x = x1($:-1:1);
13 N1 = -length(x)+1:0;
14 y = convol(x,h);
15 N = -length(x)+1:length(h)-1;
16 figure
17 a=gca();
18 plot2d3('gnn',N2,h)
19 xtitle('Impulse Response','n','h[n]');
20 a.thickness = 2;
21 figure
22 a=gca();
23 a.y_location = "origin";
24 plot2d3('gnn',N1,x)
25 xtitle('Input Response Fig 2.11(a)','n','x[n]');
26 a.thickness = 2;
27 figure
28 a=gca();
29 a.y_location = "origin";
30 plot2d3('gnn',N,y)
31 xtitle('Output Response Fig 2.11(b)','n','y[n]');
32 a.thickness = 2;

```

---

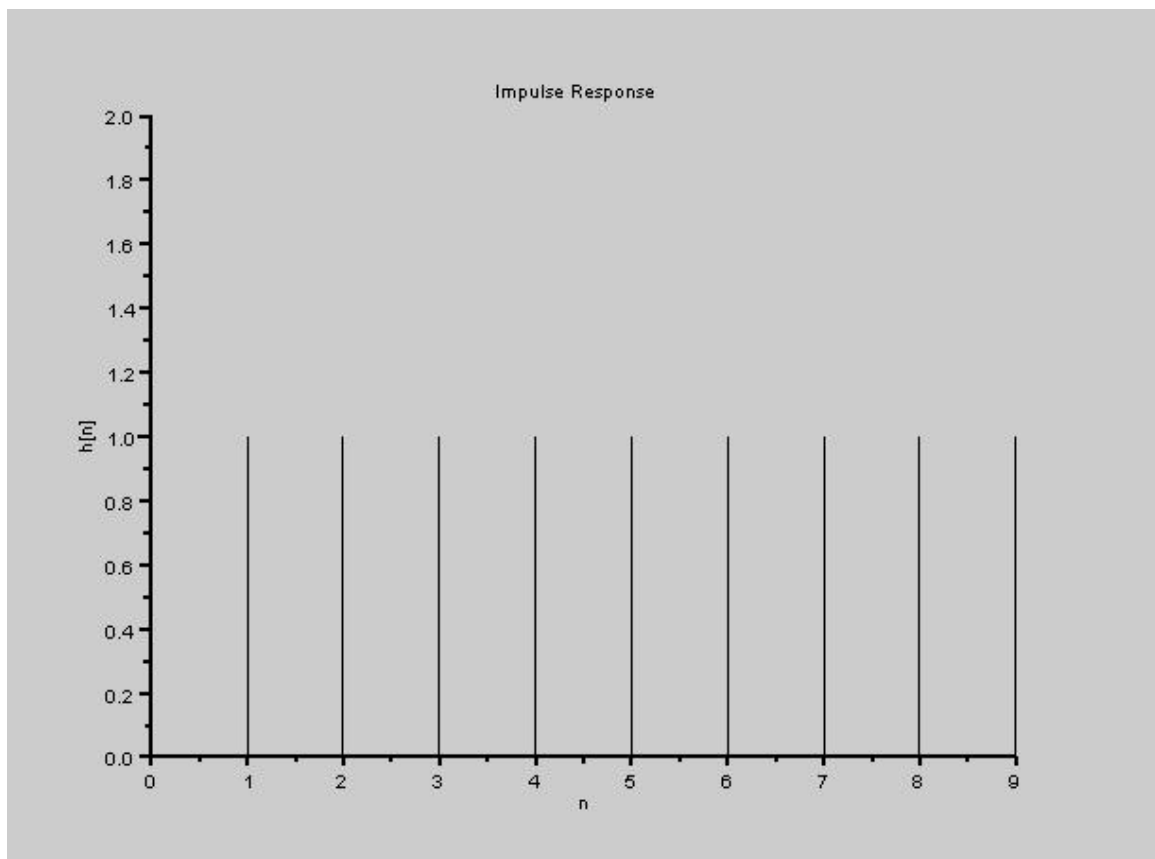


Figure 2.10: Results of Exa [2.5](#)



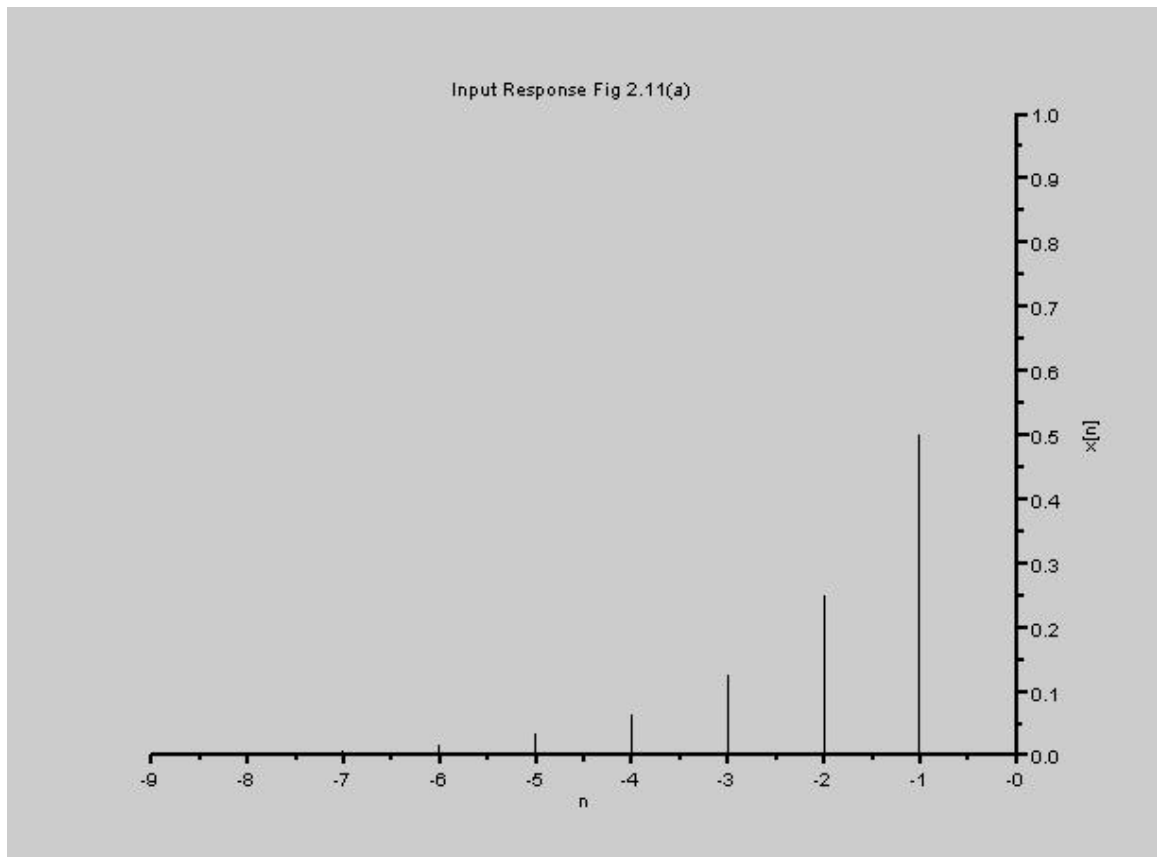


Figure 2.11: Results of Exa [2.5](#)

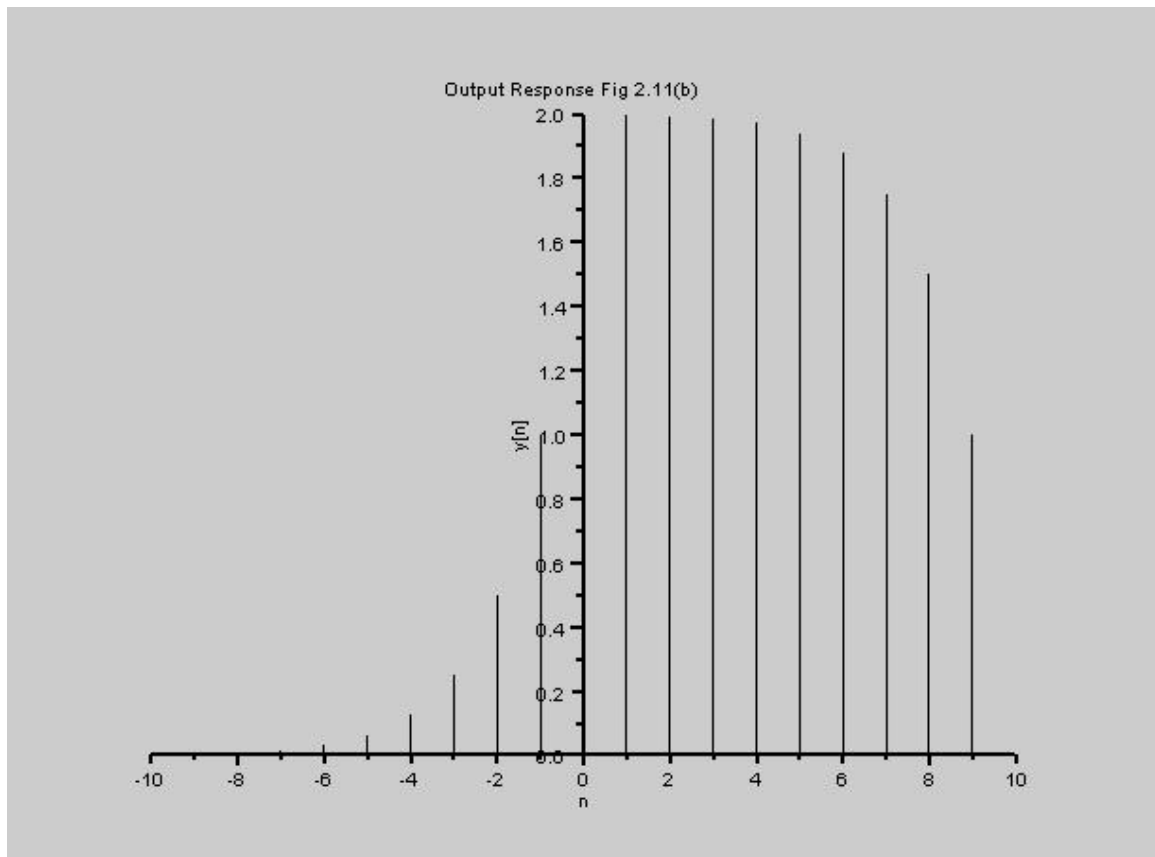


Figure 2.12: Results of Exa 2.5

**Example 2.6** onvolution Integral of input  $x(t) = (e^{-at}).u(t)$  and  $h(t) = u(t)$

```

1 //Example 2.6: Convolution Integral of input x(t) = (
    e^-at).u(t)
2 //and h(t) =u(t)
3 clear;
4 close;
5 clc;
6 Max_Limit = 10;
7 h = ones(1,Max_Limit);
8 N2 =0:length(h)-1;
9 a = 0.5; //constant a>0

```

```

10 for t = 1:Max_Limit
11     x(t)= exp(-a*(t-1));
12 end
13 N1 =0:length(x)-1;
14 y = convol(x,h)-1;
15 N = 0:length(x)+length(h)-2;
16 figure
17 a=gca();
18 plot2d(N2,h)
19 xtitle('Impulse Response','t','h(t)');
20 a.thickness = 2;
21 figure
22 a=gca();
23 plot2d(N1,x)
24 xtitle('Input Response','t','x(t)');
25 a.thickness = 2;
26 figure
27 a=gca();
28 plot2d(N(1:Max_Limit),y(1:Max_Limit))
29 xtitle('Output Response','t','y(t)');
30 a.thickness = 2;

```

---

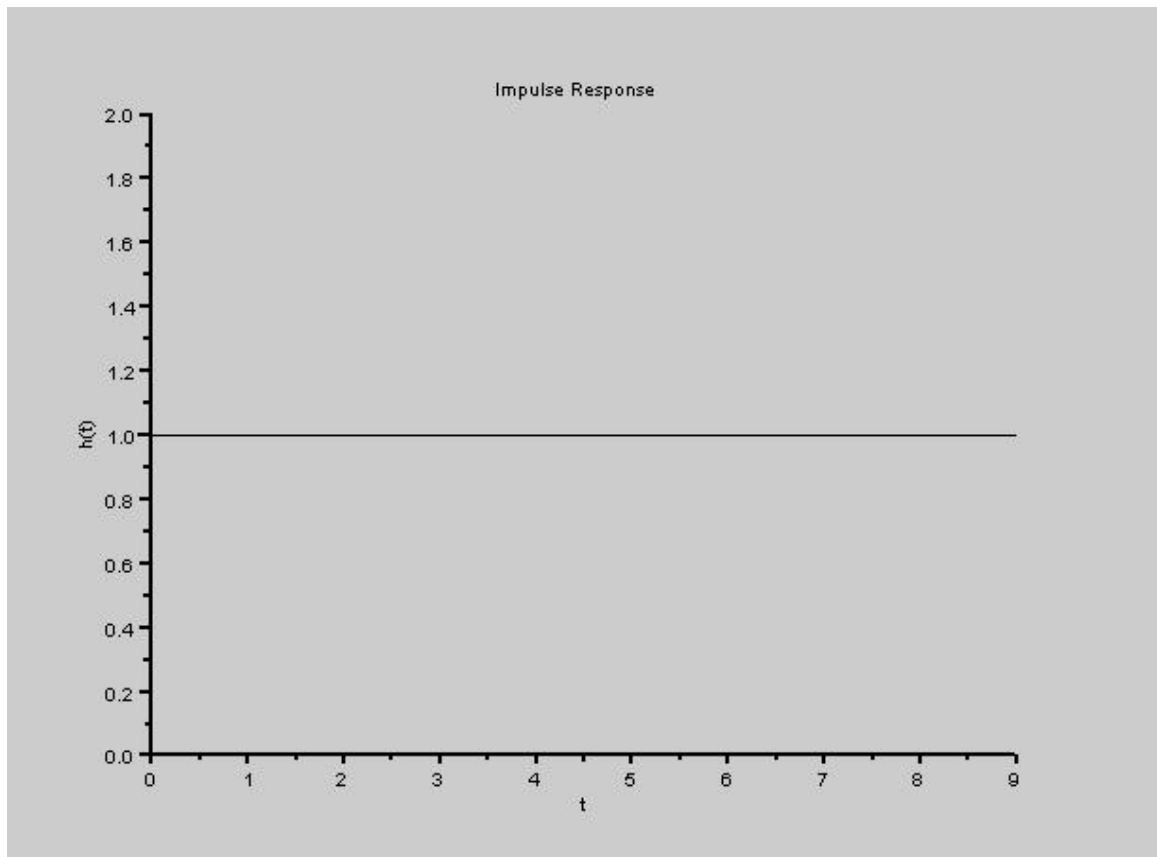


Figure 2.13: Results of Exa [2.6](#)

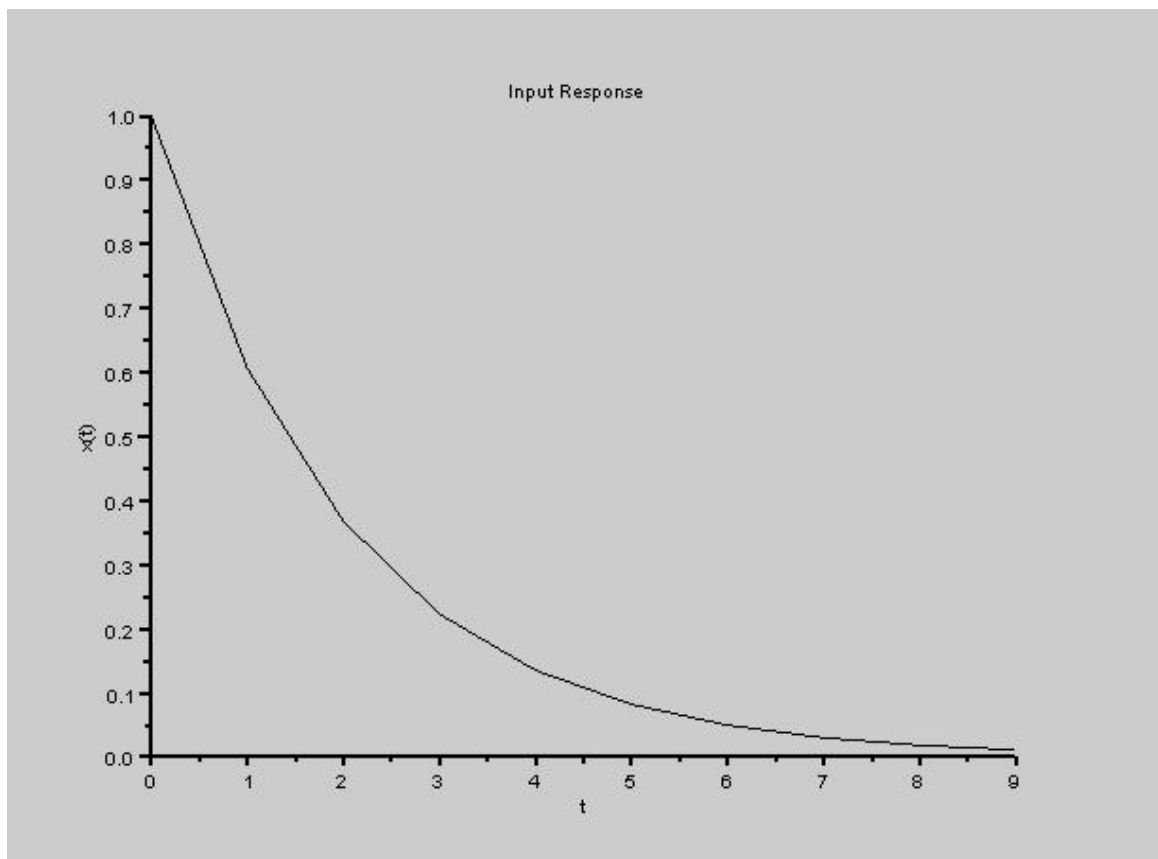


Figure 2.14: Results of Exa [2.6](#)

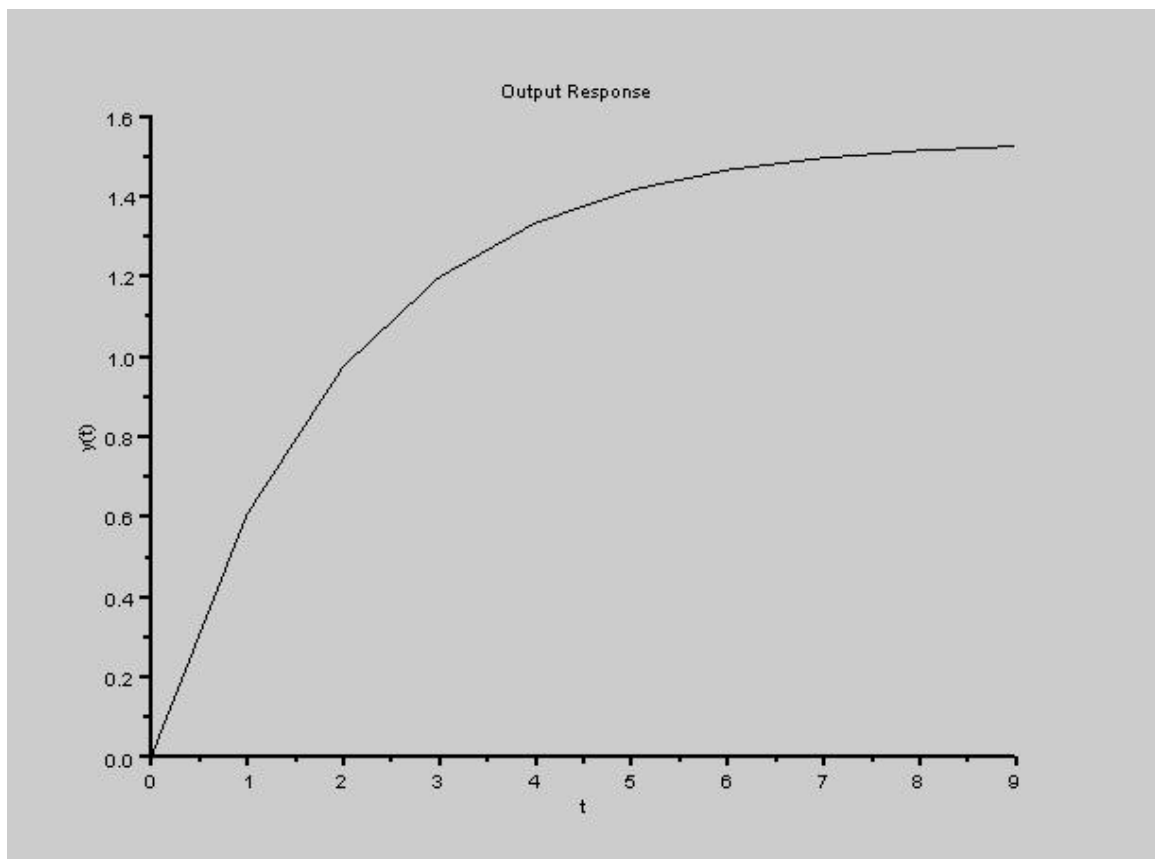


Figure 2.15: Results of Exa 2.6

**Example 2.7** Convolution Integral of finite duration signals

```

1 //Example 2.7: Convolution Integral of finite
  duration signals
2 //page99
3 clear;
4 close;
5 clc;
6 T = 10;
7 x = ones(1,T); //Input Response
8 for t = 1:2*T
9     h(t) = t-1; //Impulse Response

```

```

10 end
11 N1 = 0:length(x)-1;
12 N2 = 0:length(h)-1;
13 y = convol(x,h);
14 N = 0:length(x)+length(h)-2;
15 figure
16 a=gca();
17 a.x_location="origin";
18 plot2d(N2,h)
19 xtitle('Impulse Response','t','h(t)');
20 a.thickness = 2;
21 figure
22 a=gca();
23 plot2d(N1,x)
24 xtitle('Input Response','t','x(t)');
25 a.thickness = 2;
26 figure
27 a=gca();
28 plot2d(N,y)
29 xtitle('Output Response','t','y(t)');
30 a.thickness = 2;

```

---

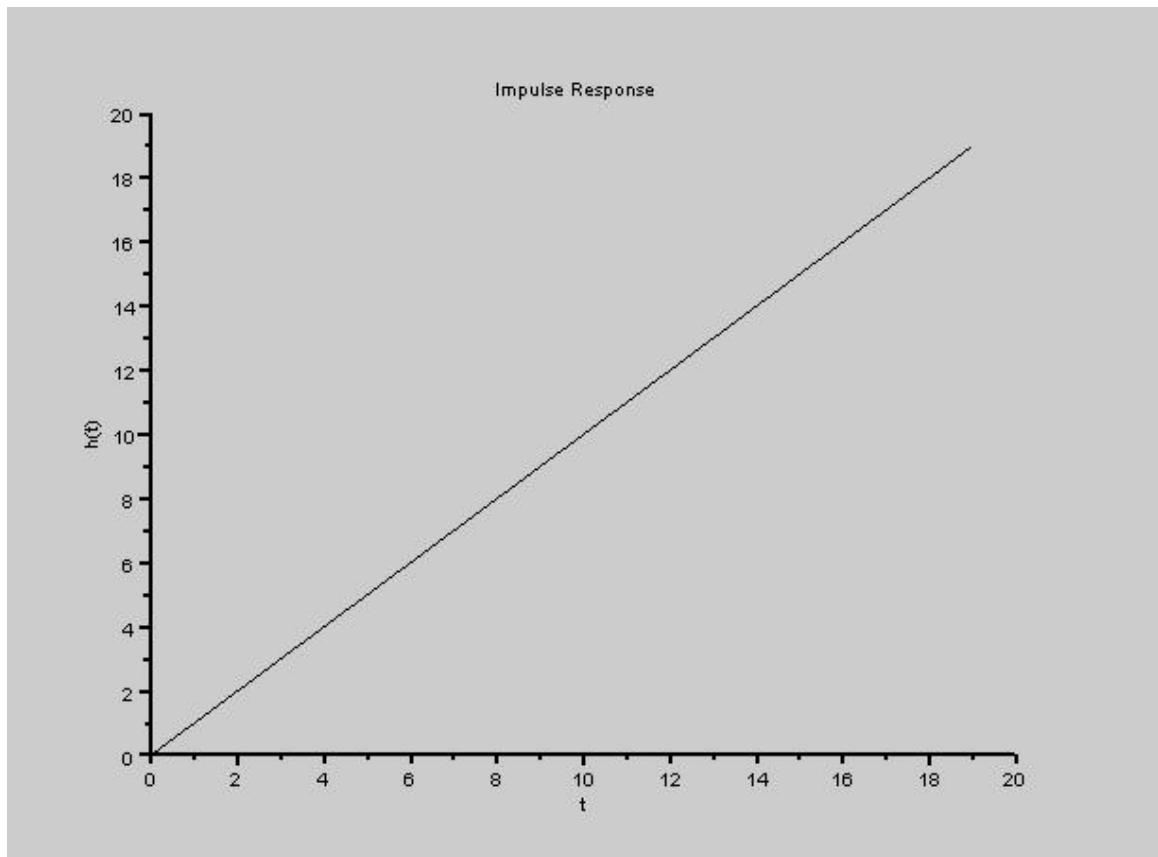


Figure 2.16: Results of Exa [2.7](#)



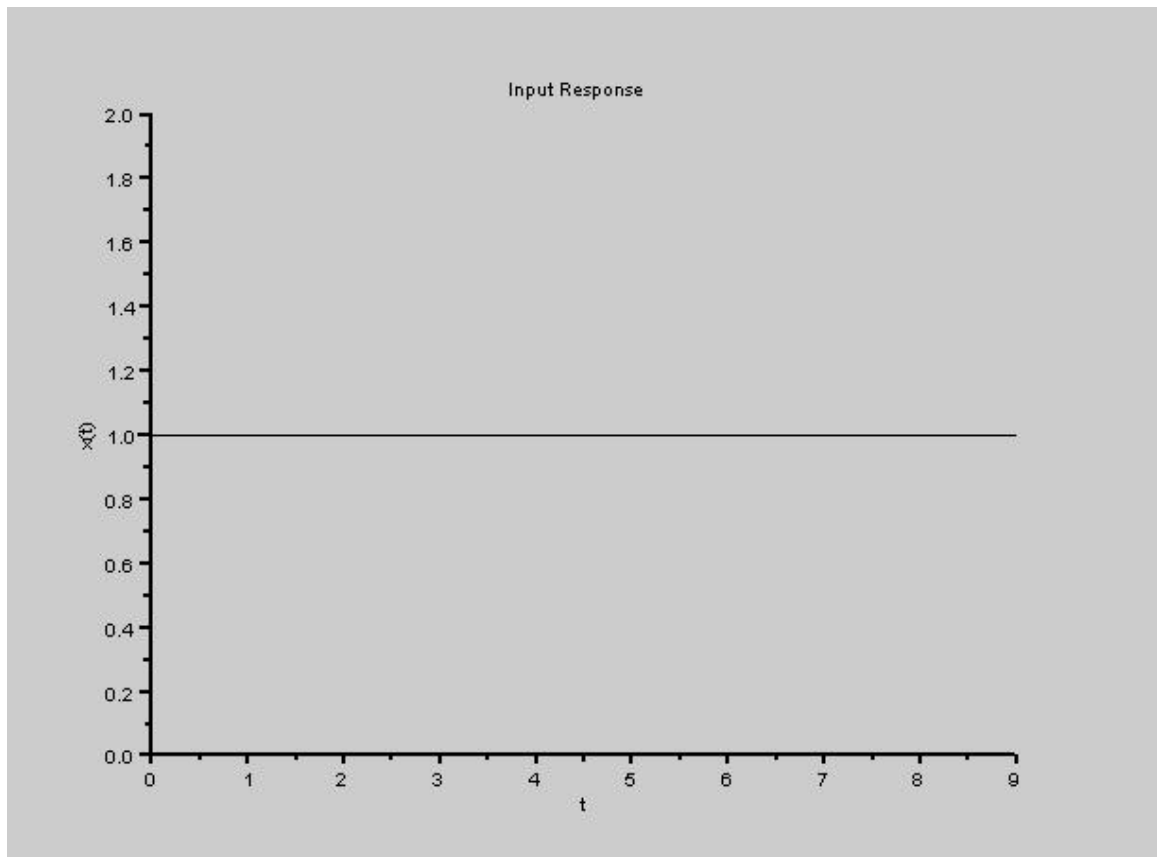


Figure 2.17: Results of Exa [2.7](#)

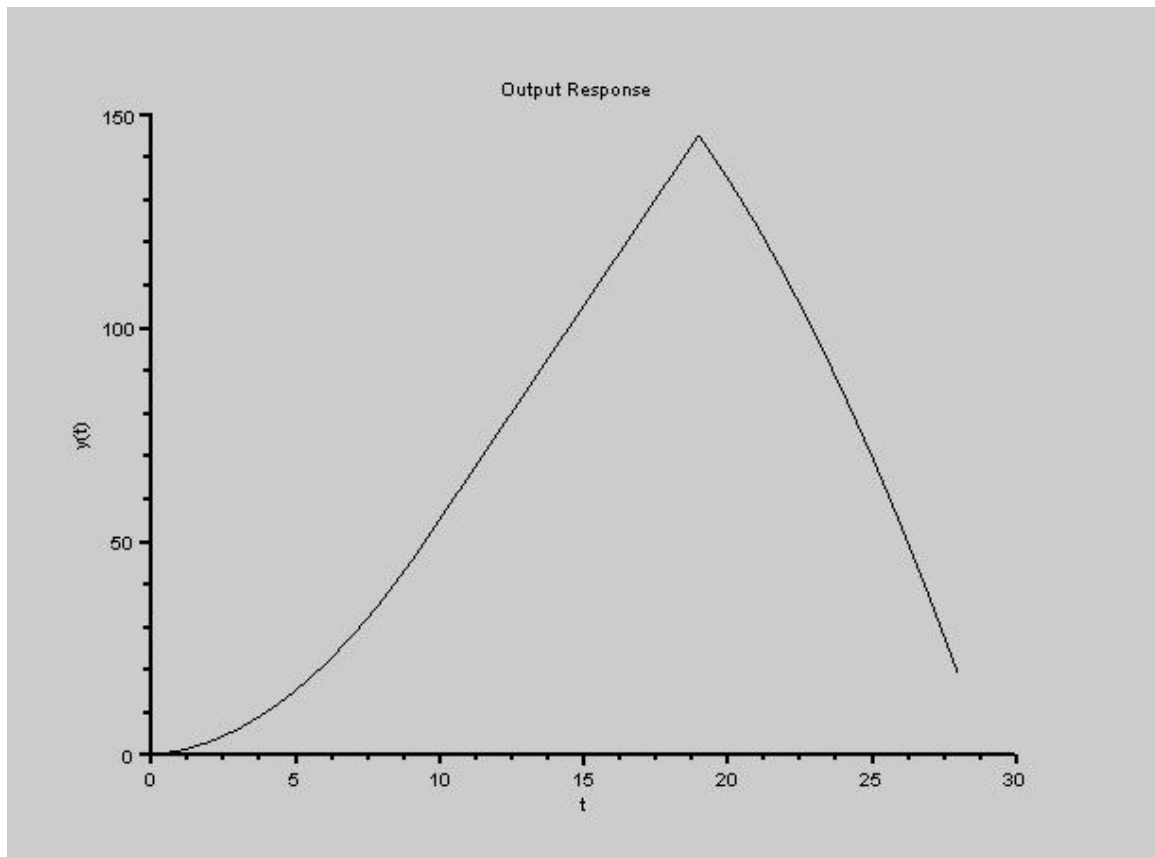


Figure 2.18: Results of Exa 2.7

**Example 2.8** Convolution Integral of input  $x(t) = (e^2t).u(-t)$  and  $h(t) = u(t-3)$

```

1 //Example 2.8: Convolution Integral of input x(t)=(e
    ^2t).u(-t) and
2 //h(t) = u(t-3)
3 clear;
4 close;
5 clc;
6 Max_Limit = 10;
7 h =[0,0,0,ones(1,Max_Limit-3)]; //h(n-3)
8 a = 2;

```

```

9  t = -9:0;
10 x= exp(a*t);
11 //x = x1($:-1:1)
12 N2 = 0:length(h)-1;
13 N1 = -length(x)+1:0;
14 t1 = -6:3;
15 y1 = (1/a)*exp(a*(t1-3));
16 y2 = (1/a)*ones(1,Max_Limit);
17 y = [y1 y2]
18 N = -length(h)+1:length(x)-1;
19 figure
20 a=gca();
21 a.x_location="origin";
22 a.y_location="origin";
23 plot2d(-Max_Limit+1:0,h($:-1:1))
24 xtitle('Impulse Response','t','h(t-T)');
25 a.thickness = 2;
26 figure
27 a=gca();
28 a.y_location = "origin";
29 plot2d(t,x)
30 xtitle('Input Response','t','x(t)');
31 a.thickness = 2;
32 figure
33 a=gca();
34 a.y_location = "origin";
35 a.x_location = "origin";
36 a.data_bounds=[-10,0;13,1];
37 plot2d(-Max_Limit+4:Max_Limit+3,y)
38 xtitle('Output Response','t','y(t)');
39 a.thickness = 2;

```

---

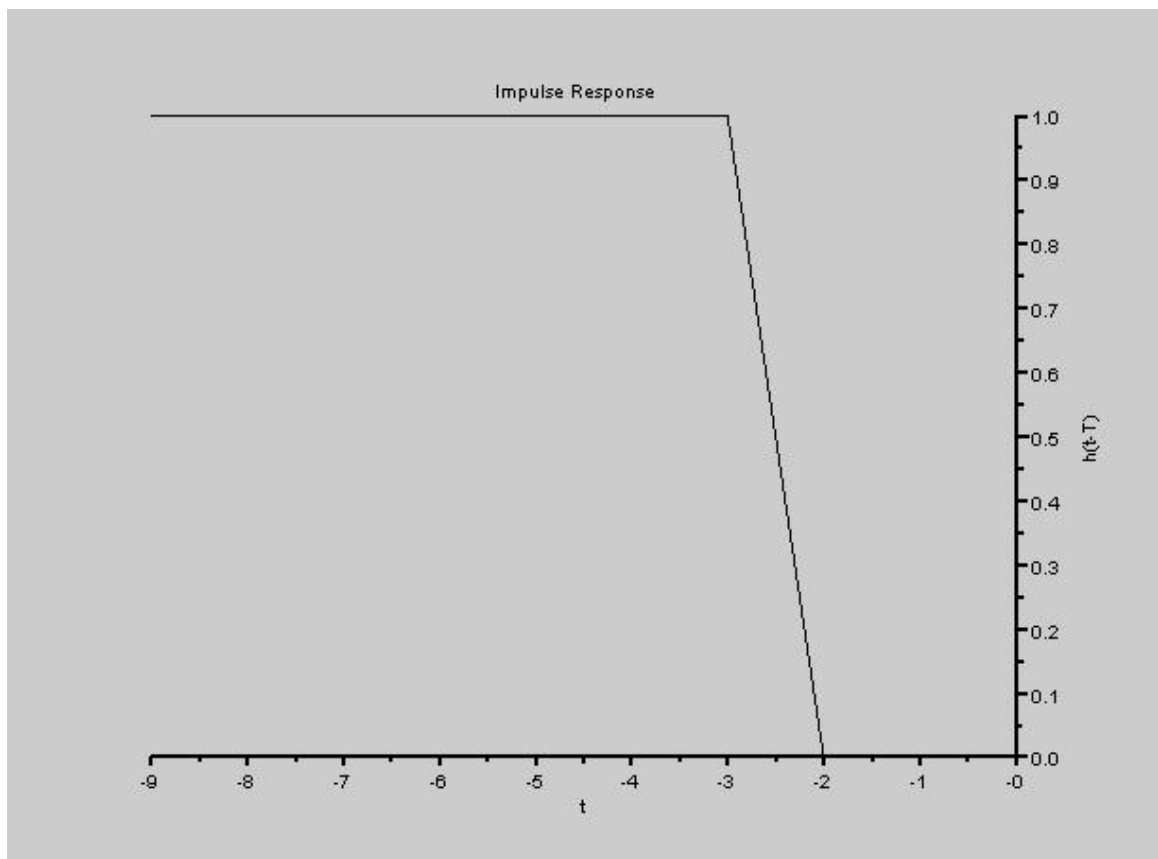


Figure 2.19: Results of Exa [2.8](#)

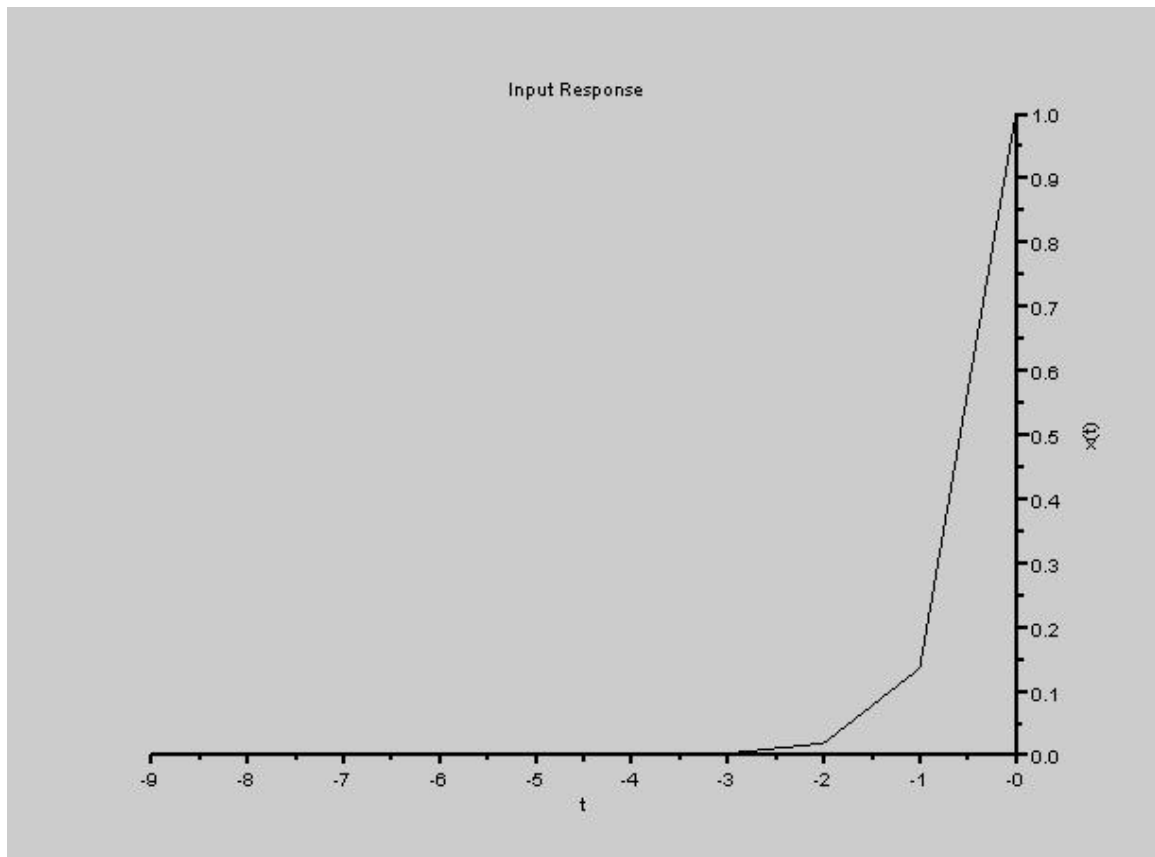


Figure 2.20: Results of Exa [2.8](#)

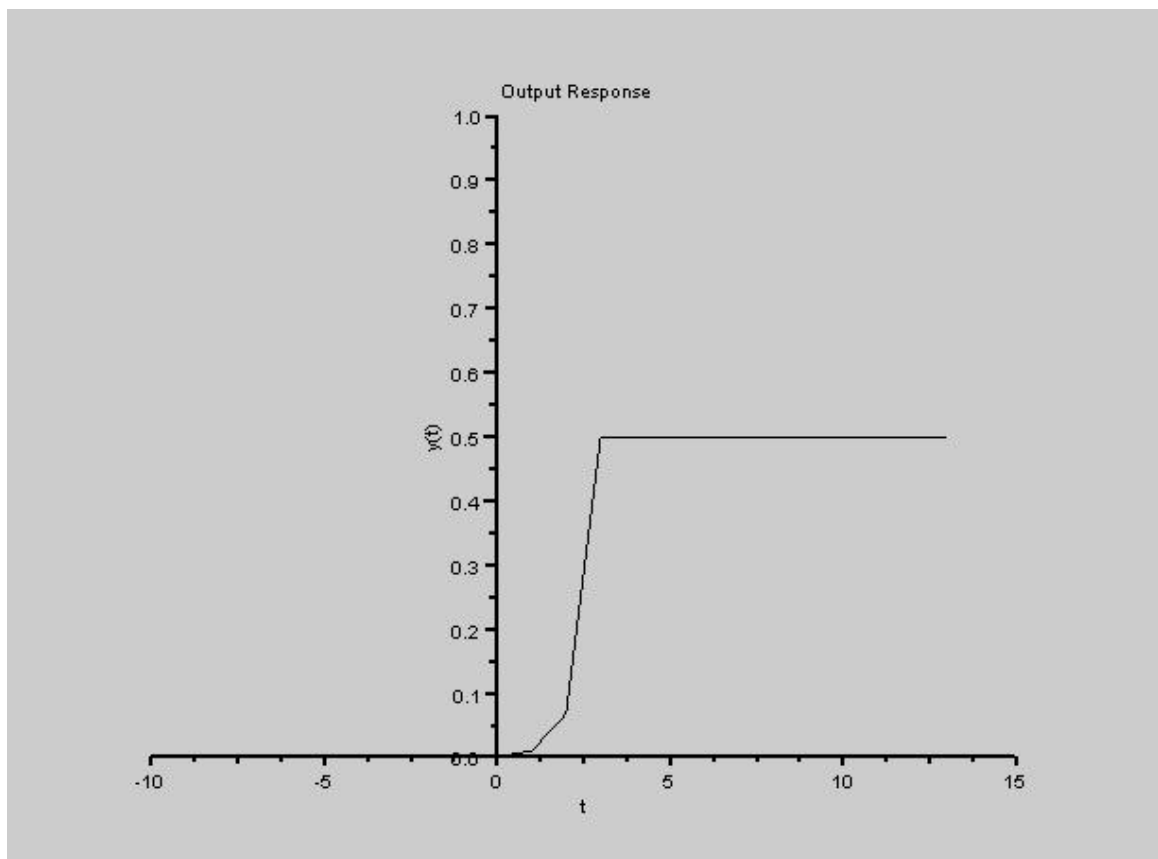


Figure 2.21: Results of Exa [2.8](#)

# Chapter 3

## Fourier Series Representation of Periodic Signals

### 3.1 Scilab Codes

**Example 3.2** CTFS of a periodic signal  $x(t)$  Expression of continuous time signal

```
1 //Example 3.2:CTFS of a periodic signal x(t)
2 //Expression of continuous time signal
3 //using continuous time fourier series
4 clear;
5 close;
6 clc;
7 t = -3:0.01:3;
8 //t1 = -%pi*4:(%pi*4)/100:%pi*4;
9 //t2 = -%pi*6:(%pi*6)/100:%pi*6;
10 xot = ones(1,length(t));
11 x1t = (1/2)*cos(%pi*2*t);
12 xot_x1t = xot+x1t;
13 x2t = cos(%pi*4*t);
14 xot_x1t_x2t = xot+x1t+x2t;
15 x3t = (2/3)*cos(%pi*6*t);
16 xt = xot+x1t+x2t+x3t;
17 //
```

```

18 figure
19 a = gca();
20 a.y_location = "origin";
21 a.x_location = "origin";
22 a.data_bounds=[-4,0;2 4];
23 plot(t,xot)
24 ylabel('t')
25 title('xot =1')
26 //
27 figure
28 subplot(2,1,1)
29 a = gca();
30 a.y_location = "origin";
31 a.x_location = "origin";
32 a.data_bounds=[-4,-3;2 4];
33 plot(t,x1t)
34 ylabel('t')
35 title('x1(t) =1/2*cos(2*pi*t)')
36 subplot(2,1,2)
37 a = gca();
38 a.y_location = "origin";
39 a.x_location = "origin";
40 a.data_bounds=[-4,0;2 4];
41 plot(t,xot_x1t)
42 ylabel('t')
43 title('xo(t)+x1(t)')
44 //
45 figure
46 subplot(2,1,1)
47 a = gca();
48 a.y_location = "origin";
49 a.x_location = "origin";
50 a.data_bounds=[-4,-2;4 2];
51 plot(t,x2t)
52 ylabel('t')
53 title('x2(t) =cos(4*pi*t)')
54 subplot(2,1,2)
55 a = gca();

```



```

56 a.y_location = "origin";
57 a.x_location = "origin";
58 a.data_bounds=[-4,0;4 4];
59 plot(t,xot_x1t_x2t)
60 ylabel('t')
61 title('xo(t)+x1(t)+x2(t)')
62 //
63 figure
64 subplot(2,1,1)
65 a = gca();
66 a.y_location = "origin";
67 a.x_location = "origin";
68 a.data_bounds=[-4,-3;4 3];
69 plot(t,x3t)
70 ylabel('t')
71 title('x1(t) = 2/3*cos(6*pi*t)')
72 subplot(2,1,2)
73 a = gca();
74 a.y_location = "origin";
75 a.x_location = "origin";
76 a.data_bounds=[-4,-3;4 3];
77 plot(t,xt)
78 ylabel('t')
79 title('x(t)=xo(t)+x1(t)+x2(t)+x3(t)')

```

---

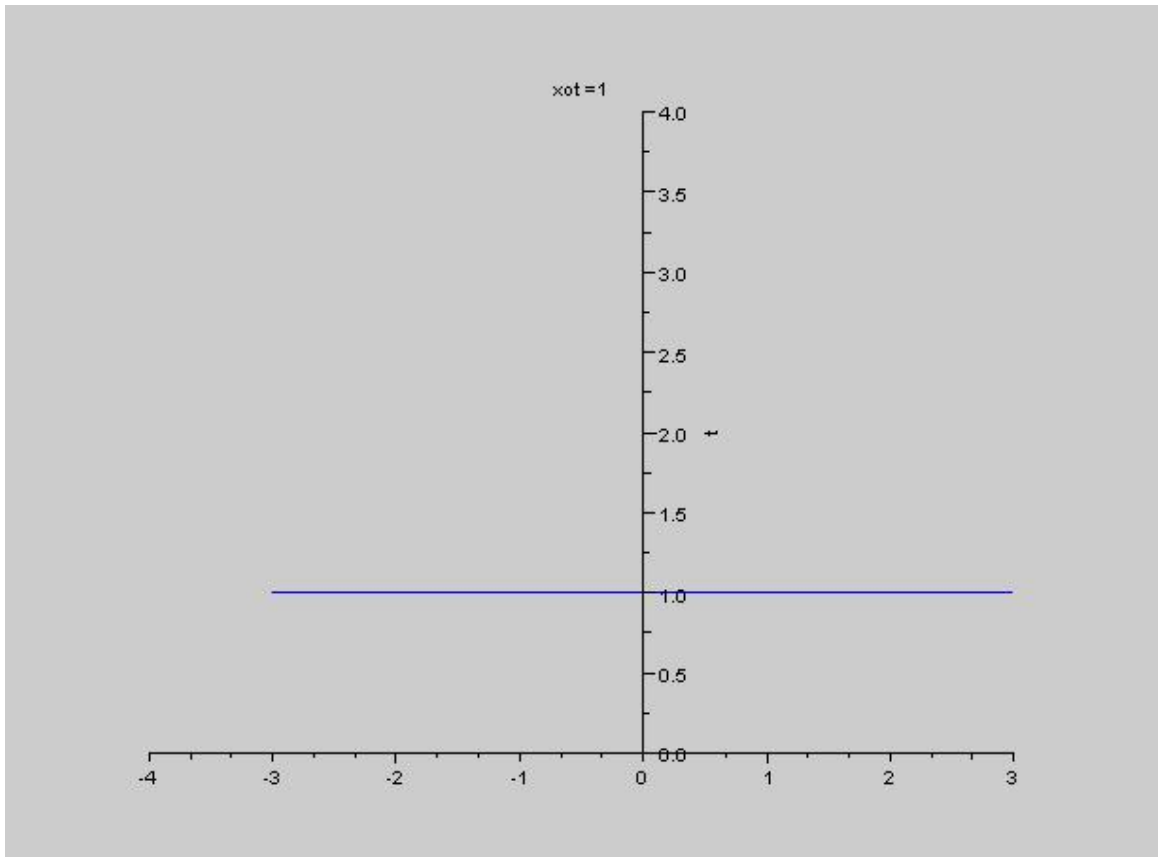


Figure 3.1: Results of Exa [3.2](#)

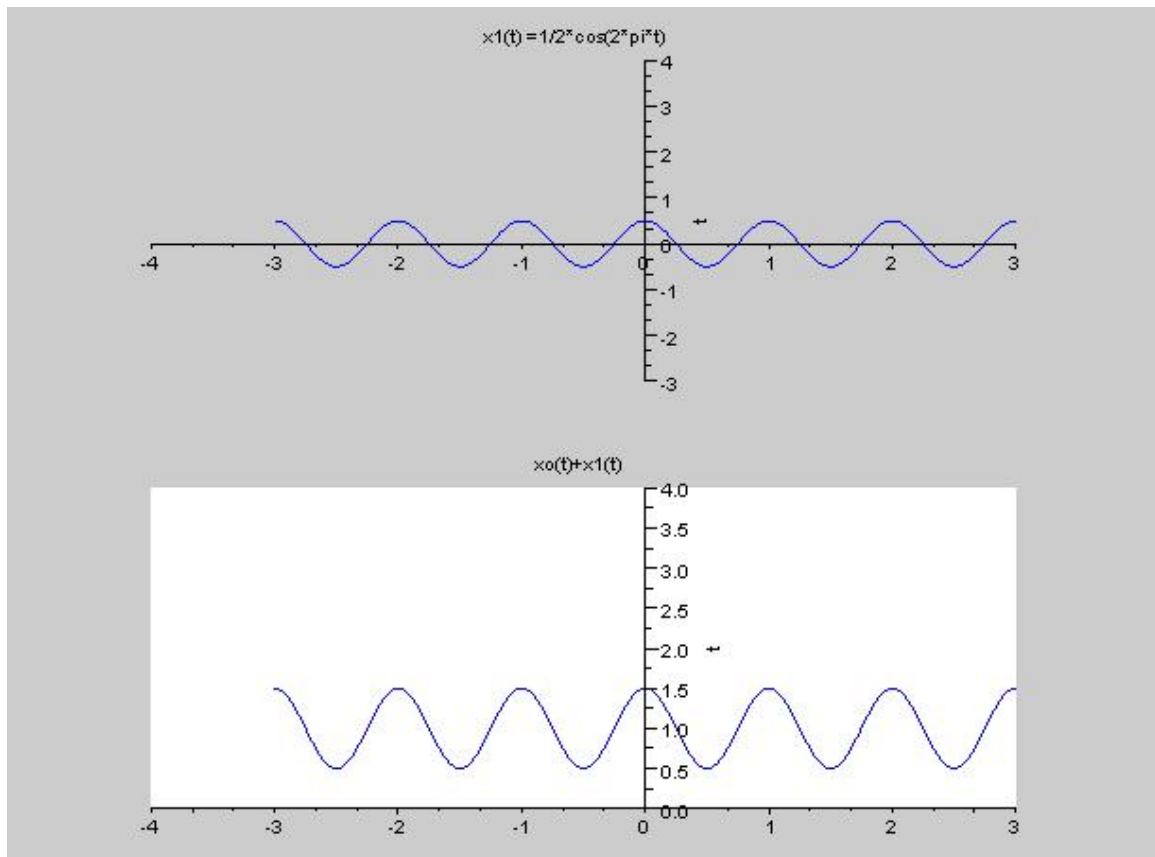


Figure 3.2: Results of Exa 3.2

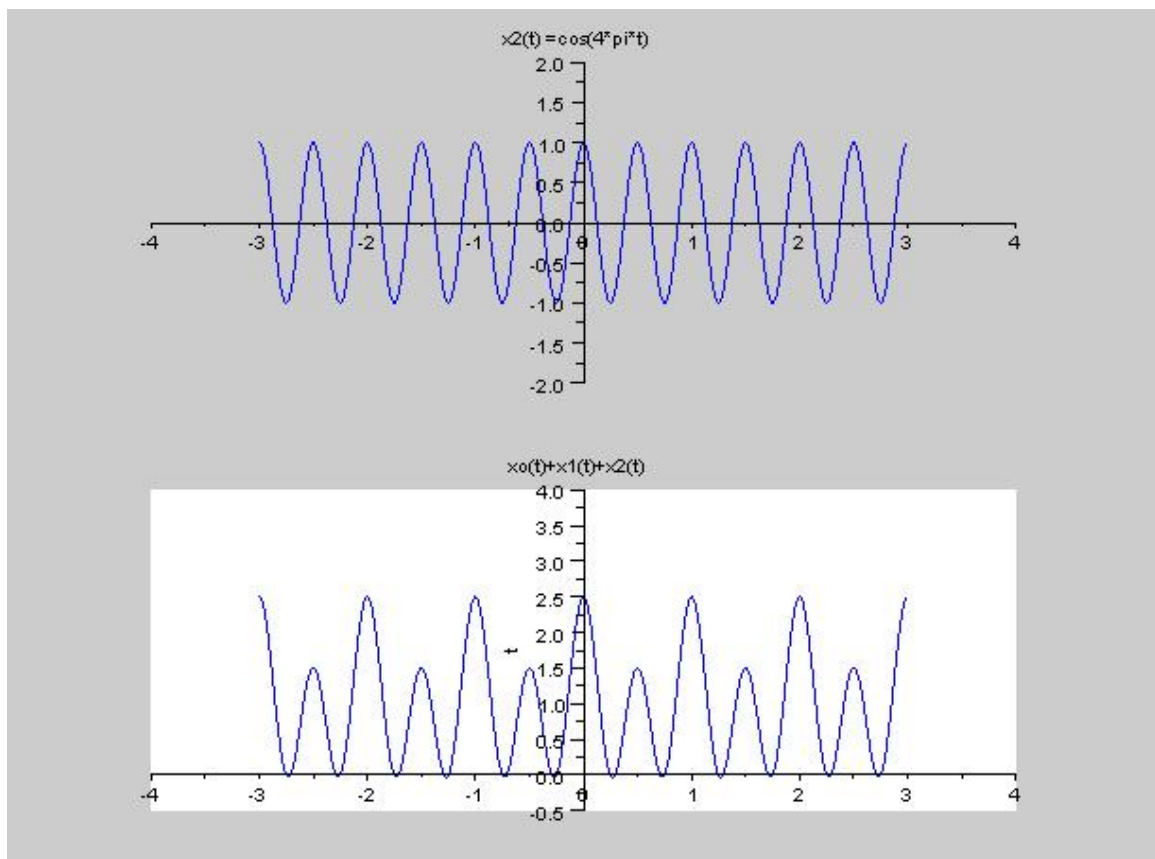


Figure 3.3: Results of Exa 3.2

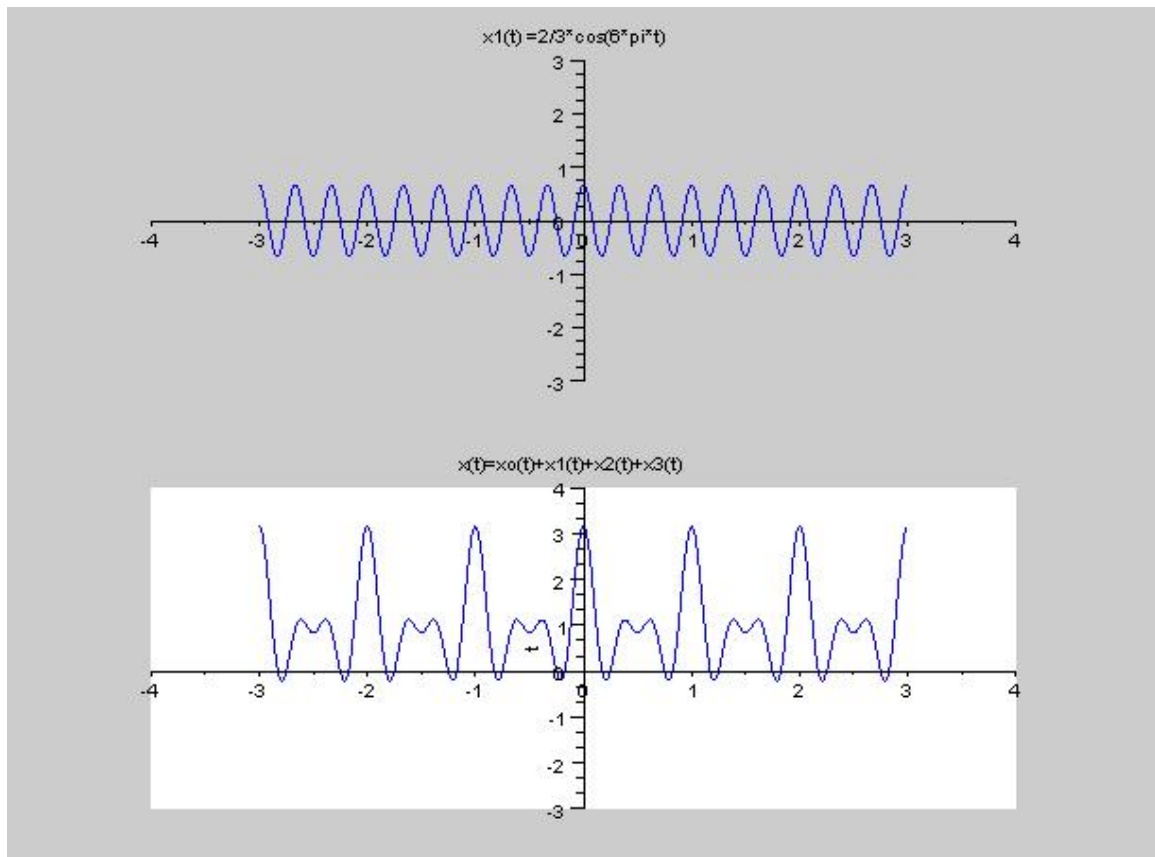


Figure 3.4: Results of Exa 3.2

**Example 3.3** Continuous Time Fourier Series Coefficients of a periodic signal  $x(t) = \sin(Wot)$

```

1 //Example3.3: Continuous Time Fourier Series
  Coefficients of
2 //a periodic signal x(t) = sin(Wot)
3 clear;
4 close;
5 clc;
6 t = 0:0.01:1;
7 T = 1;
8 Wo = 2*%pi/T;

```

```

9  xt = sin(Wo*t);
10 for k =0:5
11     C(k+1,:) = exp(-sqrt(-1)*Wo*t.*k);
12     a(k+1) = xt*C(k+1,:)' /length(t);
13     if(abs(a(k+1))<=0.01)
14         a(k+1)=0;
15     end
16 end
17 a =a';
18 ak = [-a,a(2:$)];

```

---

**Example 3.4** CTFS coefficients of a periodic signal  $x(t) = 1 + \sin(Wot) + 2\cos(Wot) + \cos(2Wot + \pi/4)$

```

1 //Example3.4:CTFS coefficients of a periodic signal
2 //x(t) = 1+sin(Wot)+2cos(Wot)+cos(2Wot+%pi/4)
3 clear;
4 close;
5 clc;
6 t = 0:0.01:1;
7 T = 1;
8 Wo = 2*%pi/T;
9 xt =ones(1,length(t))+sin(Wo*t)+2*cos(Wo*t)+cos(2*Wo
    *t+%pi/4);
10 for k =0:5
11     C(k+1,:) = exp(-sqrt(-1)*Wo*t.*k);
12     a(k+1) = xt*C(k+1,:)' /length(t);
13     if(abs(a(k+1))<=0.1)
14         a(k+1)=0;
15     end
16 end
17 a =a';
18 a_conj =conj(a);
19 ak = [a_conj($:-1:1),a(2:$)];
20 Mag_ak = abs(ak);
21 for i = 1:length(a)
22     Phase_ak(i) = atan(imag(ak(i))/(real(ak(i))
        +0.0001));

```

```

23 end
24 Phase_ak = Phase_ak'
25 Phase_ak = [Phase_ak(1:$) -Phase_ak($-1:-1:1)];
26 figure
27 subplot(2,1,1)
28 a = gca();
29 a.y_location = "origin";
30 a.x_location = "origin";
31 plot2d3('gnn',[-k:k],Mag_ak,5)
32 poly1 = a.children(1).children(1);
33 poly1.thickness = 3;
34 title('abs(ak)')
35 xlabel('

        k')
36 subplot(2,1,2)
37 a = gca();
38 a.y_location = "origin";
39 a.x_location = "origin";
40 plot2d3('gnn',[-k:k],Phase_ak,5)
41 poly1 = a.children(1).children(1);
42 poly1.thickness = 3;
43 title('<(ak)')
44 xlabel('

        k')

```

---

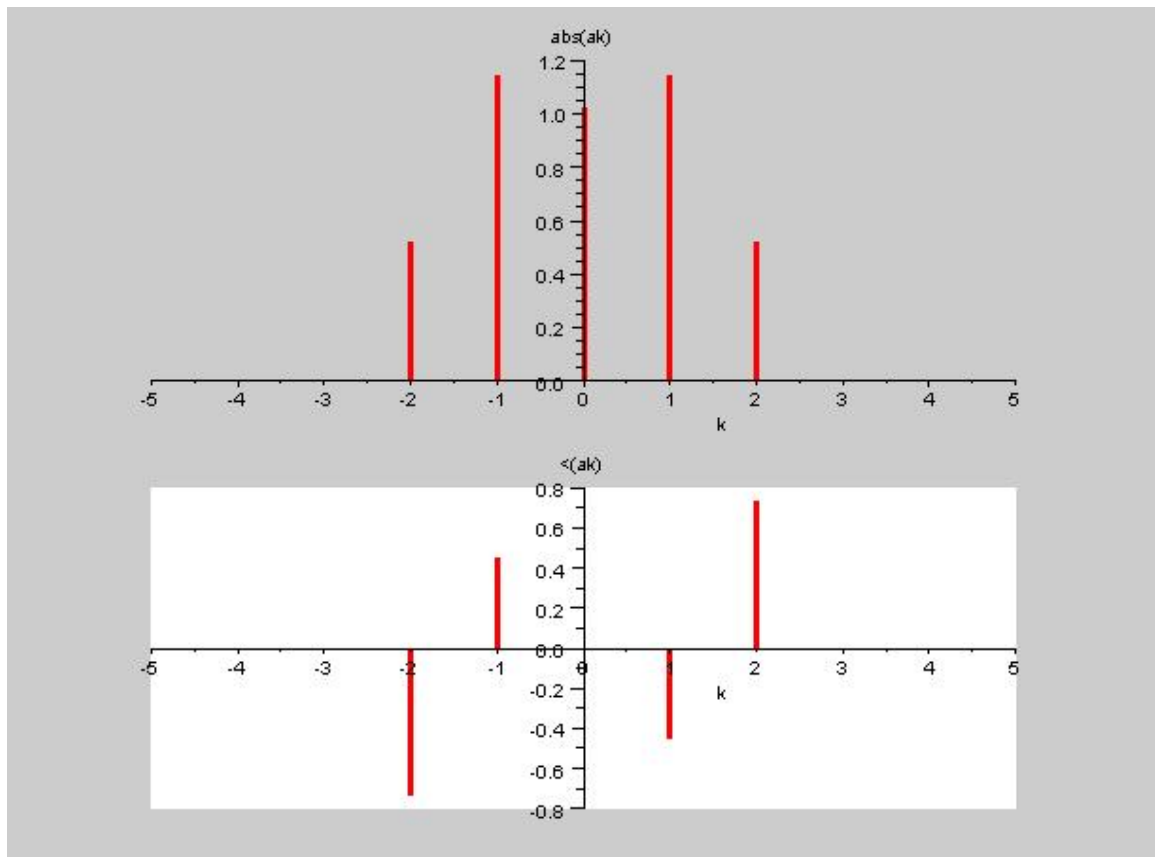


Figure 3.5: Results of Exa 3.4

**Example 3.5** CTFS coefficients of a periodic signal  $x(t) = 1, |t| < T_1$ , and  $0, T_1 < |t| < T/2$

```

1 //Example3.5:CTFS coefficients of a periodic signal
2 //x(t) = 1, |t|<T1, and 0, T1<|t|<T/2
3 clear;
4 close;
5 clc;
6 T =4;
7 T1 = T/4;
8 t = -T1:T1/100:T1;
9 Wo = 2*%pi/T;

```



```

10 xt =ones(1,length(t));
11 //
12 for k =0:5
13     C(k+1,:) = exp(-sqrt(-1)*Wo*t.*k);
14     a(k+1) = xt*C(k+1,:)/length(t);
15     if(abs(a(k+1))<=0.1)
16         a(k+1)=0;
17     end
18 end
19 a =a';
20 a_conj = real(a(:))-sqrt(-1)*imag(a(:));
21 ak = [a_conj($:-1:1)',a(2:$)];
22 k = 0:5;
23 k = [-k($:-1:1),k(2:$)];
24 Spectrum_ak = (1/2)*real(ak);
25 //
26 figure
27 a = gca();
28 a.y_location = "origin";
29 a.x_location = "origin";
30 a.data_bounds=[-2,0;2,2];
31 plot2d(t,xt,5)
32 poly1 = a.children(1).children(1);
33 poly1.thickness = 3;
34 title('x(t)')
35 xlabel('

    t')
36 //
37 figure
38 a = gca();
39 a.y_location = "origin";
40 a.x_location = "origin";
41 plot2d3('gnn',k,Spectrum_ak,5)
42 poly1 = a.children(1).children(1);
43 poly1.thickness = 3;
44 title('abs(ak)')
45 xlabel('

```

$k')$

---

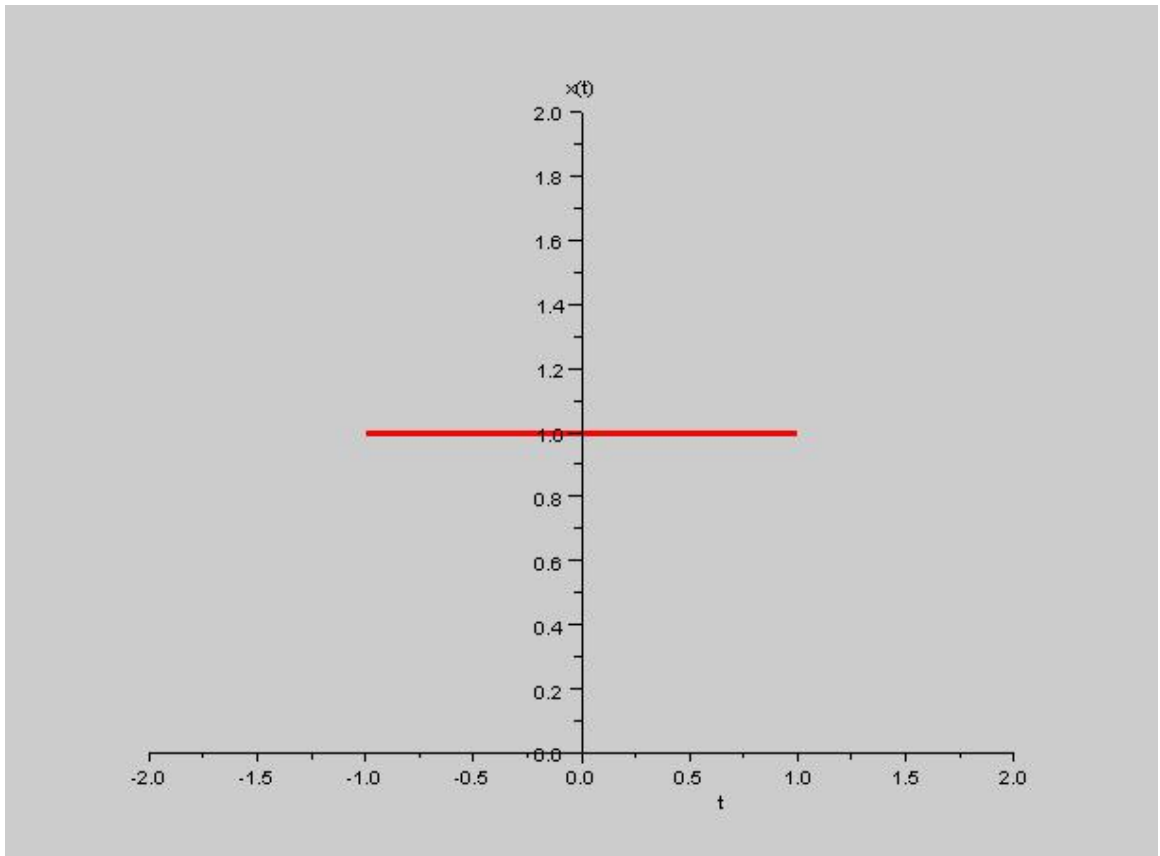


Figure 3.6: Results of Exa [3.5](#)

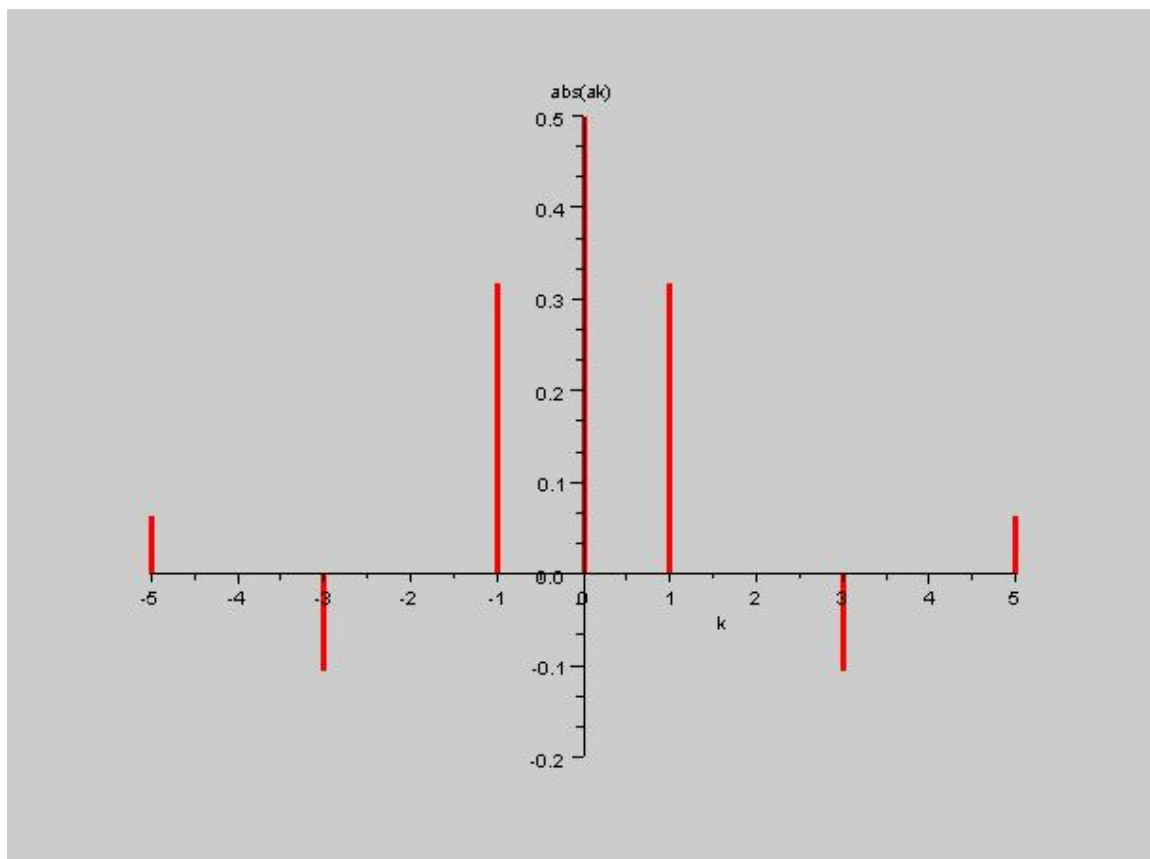


Figure 3.7: Results of Exa 3.5

### Example 3.6 Time Shift Property of CTFS

```

1 //Example3.6: Time Shift Property of CTFS
2 clear;
3 close;
4 clc;
5 T =4;
6 T1 = T/2;
7 t = 0:T1/100:T1;
8 Wo = 2*%pi/T;
9 gt =(1/2)*ones(1,length(t));
10 a(1)=0; //k=0, ak =0

```

```

11 d(1)=0;
12 for k =1:5
13     a(k+1) = (sin(%pi*k/2)/(k*%pi));
14     if(abs(a(k+1))<=0.01)
15         a(k+1)=0;
16     end
17     d(k+1) = a(k+1)*exp(-sqrt(-1)*k*%pi/2);
18 end
19 k = 0:5
20 disp('Fourier Series Coefficients of Square Wave')
21 a
22 disp('Fourier Series Coefficients of  $g(t)=x(t-1)-0.5$ 
    ')
23 d
24 //
25 figure
26 a = gca();
27 a.y_location = "origin";
28 a.x_location = "origin";
29 a.data_bounds=[-1,-2;1,4];
30 plot2d([-t($:-1:1),t(1:$)],[-gt,gt],5)
31 poly1 = a.children(1).children(1);
32 poly1.thickness = 3;
33 title('g(t)')
34 xlabel('
    t ')

```

---

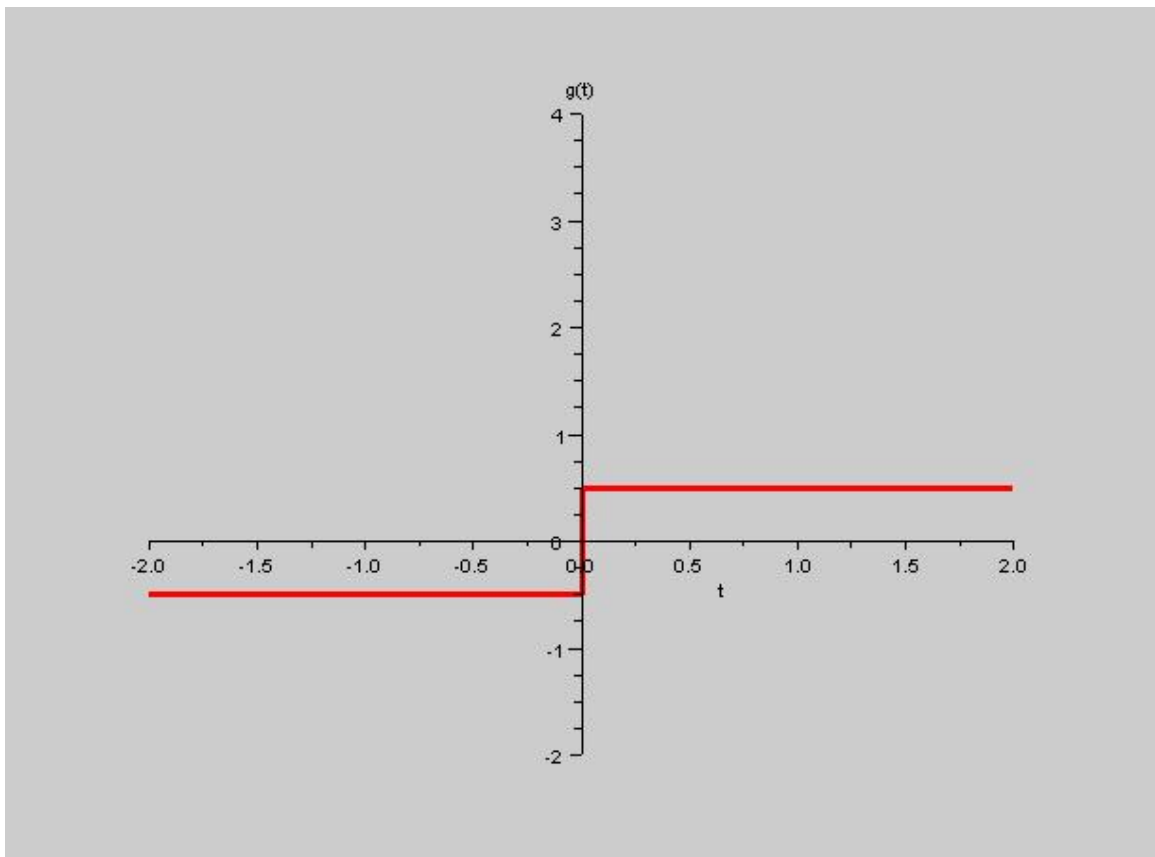


Figure 3.8: Results of Exa 3.6

**Example 3.7** Derivative Property of CTFS

```

1 //Example3.7: Derivative Property of CTFS
2 clear;
3 clc;
4 close;
5 T =4;
6 T1 = T/2;
7 t = 0:T1/100:T1;
8 xt = [t($:-1:1) t]/T1;
9 gt = (1/2)*ones(1,length(t));
10 e(1) = 1/2; //k =0, e0 = 1/2

```

```

11 for k = 1:5
12     a(k+1) = (sin(%pi*k/2)/(k*pi));
13     if(abs(a(k+1)) <= 0.01)
14         a(k+1) = 0;
15     end
16     d(k+1) = a(k+1)*exp(-sqrt(-1)*k*pi/2);
17     e(k+1) = 2*d(k+1)/(sqrt(-1)*k*pi);
18 end
19 k = 0:5
20 disp('Fourier Series Coefficients of Square Wave')
21 a
22 disp('Fourier Series Coefficients of  $g(t)=x(t-1)-0.5$ 
    ')
23 d
24 disp('Fourier Series Coefficients of Triangular Wave
    ')
25 e
26 //Plotting the time shifted square waveform
27 figure
28 a = gca();
29 a.y_location = "origin";
30 a.x_location = "origin";
31 a.data_bounds = [-1,-2;1,2];
32 plot2d([-t($:-1:1),t(1:$)],[-gt,gt],5)
33 poly1 = a.children(1).children(1);
34 poly1.thickness = 3;
35 title('g(t)')
36 xlabel('
    t')
37 //Plotting the Triangular waveform
38 figure
39 a = gca();
40 a.y_location = "origin";
41 a.x_location = "origin";
42 a.data_bounds = [-1,0;1,2];
43 plot2d([-t($:-1:1),t(1:$)],xt,5)
44 poly1 = a.children(1).children(1);

```

```
45 poly1.thickness = 3;  
46 title('x(t)')  
47 xlabel('t')
```

---

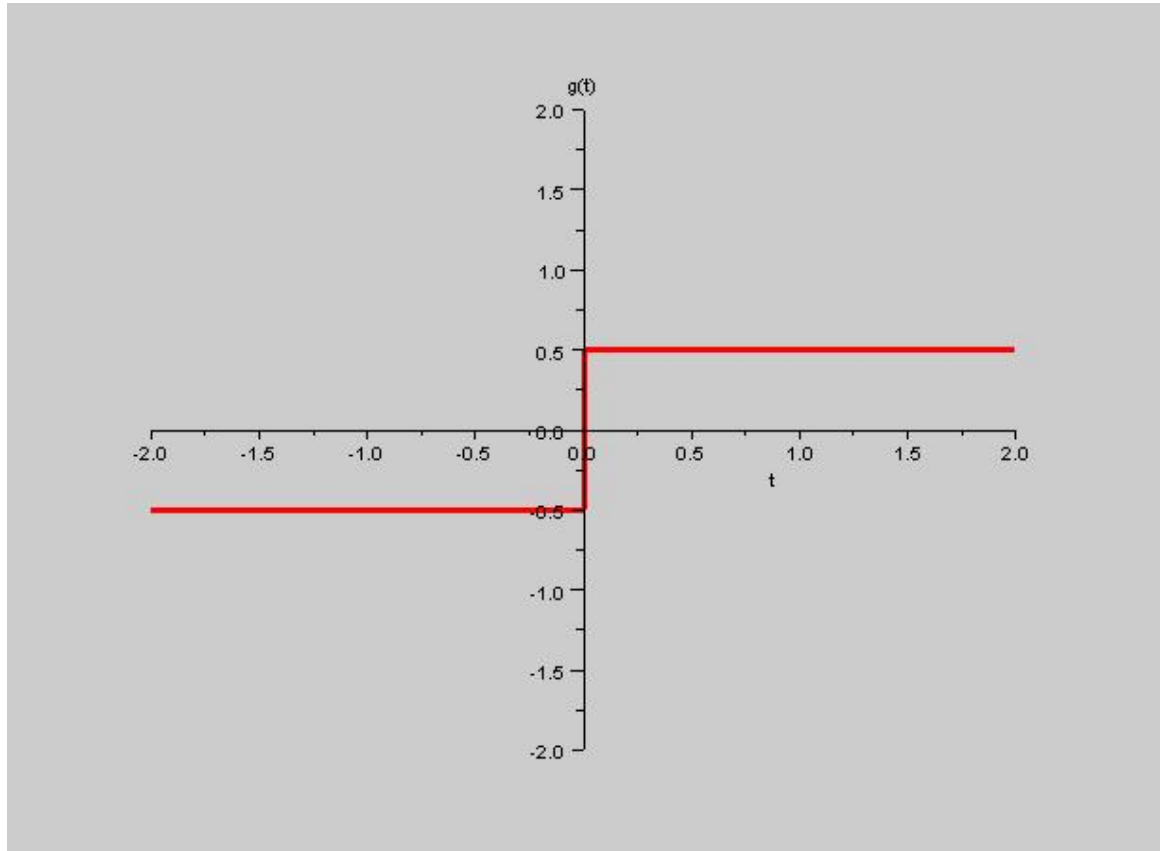


Figure 3.9: Results of Exa [3.7](#)

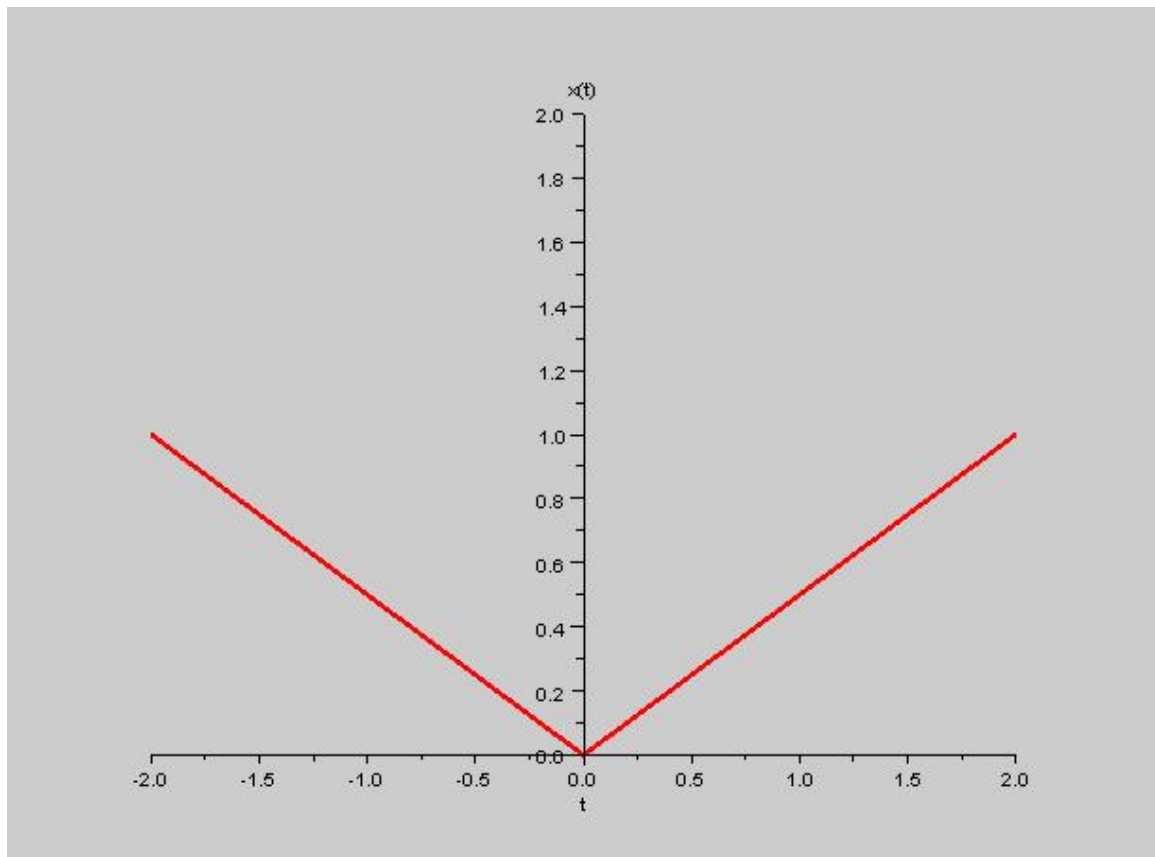


Figure 3.10: Results of Exa 3.7

**Example 3.8** Fourier Series Representation of Periodic Impulse Train

```

1 //Example3.8:Fourier Series Representation of
  Periodic Impulse Train
2 clear;
3 clc;
4 close;
5 T =4;
6 T1 = T/4;
7 t = [-T,0,T];
8 xt = [1,1,1]; //Generation of Periodic train of
  Impulses

```



```

9  t1 = -T1:T1/100:T1;
10 gt = ones(1,length(t1)); //Generation of periodic
    square wave
11 t2 = [-T1,0,T1];
12 qt = [1,0,-1]; //Derivative of periodic square wave
13 Wo = 2*%pi/T;
14 ak = 1/T;
15 b(1) = 0;
16 c(1) = 2*T1/T;
17 for k =1:5
18     b(k+1) = ak*(exp(sqrt(-1)*k*Wo*T1)-exp(-sqrt(-1)*k
        *Wo*T1));
19     if(abs(b(k+1))<=0.1)
20         b(k+1) =0;
21     end
22     c(k+1) = b(k+1)/(sqrt(-1)*k*Wo);
23     if(abs(c(k+1))<=0.1)
24         c(k+1) =0;
25     end
26 end
27 k = 0:5
28 disp('Fourier Series Coefficients of periodic Square
    Wave')
29 disp(b)
30 disp('Fourier Series Coefficients of derivative of
    periodic square wave')
31 disp(c)
32 //Plotting the periodic train of impulses
33 figure
34 subplot(3,1,1)
35 a = gca();
36 a.y_location = "origin";
37 a.x_location = "origin";
38 a.data_bounds=[-6,0;6,2];
39 plot2d3('gnn',t,xt,5)
40 poly1 = a.children(1).children(1);
41 poly1.thickness = 3;
42 title('x(t)')

```

```

43 //Plotting the periodic square waveform
44 subplot(3,1,2)
45 a = gca();
46 a.y_location = "origin";
47 a.x_location = "origin";
48 a.data_bounds=[-6,0;6,2];
49 plot2d(t1,gt,5)
50 poly1 = a.children(1).children(1);
51 poly1.thickness = 3;
52 plot2d(T+t1,gt,5)
53 poly1 = a.children(1).children(1);
54 poly1.thickness = 3;
55 plot2d(-T+t1,gt,5)
56 poly1 = a.children(1).children(1);
57 poly1.thickness = 3;
58 title('g(t)')
59 //Plotting the periodic square waveform
60 subplot(3,1,3)
61 a = gca();
62 a.y_location = "origin";
63 a.x_location = "origin";
64 a.data_bounds=[-6,-2;6,2];
65 poly1 = a.children(1).children(1);
66 poly1.thickness = 3;
67 plot2d3('gnn',t2,qt,5)
68 poly1 = a.children(1).children(1);
69 poly1.thickness = 3;
70 plot2d3('gnn',T+t2,qt,5)
71 poly1 = a.children(1).children(1);
72 poly1.thickness = 3;
73 plot2d3('gnn',-T+t2,qt,5)
74 poly1 = a.children(1).children(1);
75 poly1.thickness = 3;
76 title('q(t)')

```

---

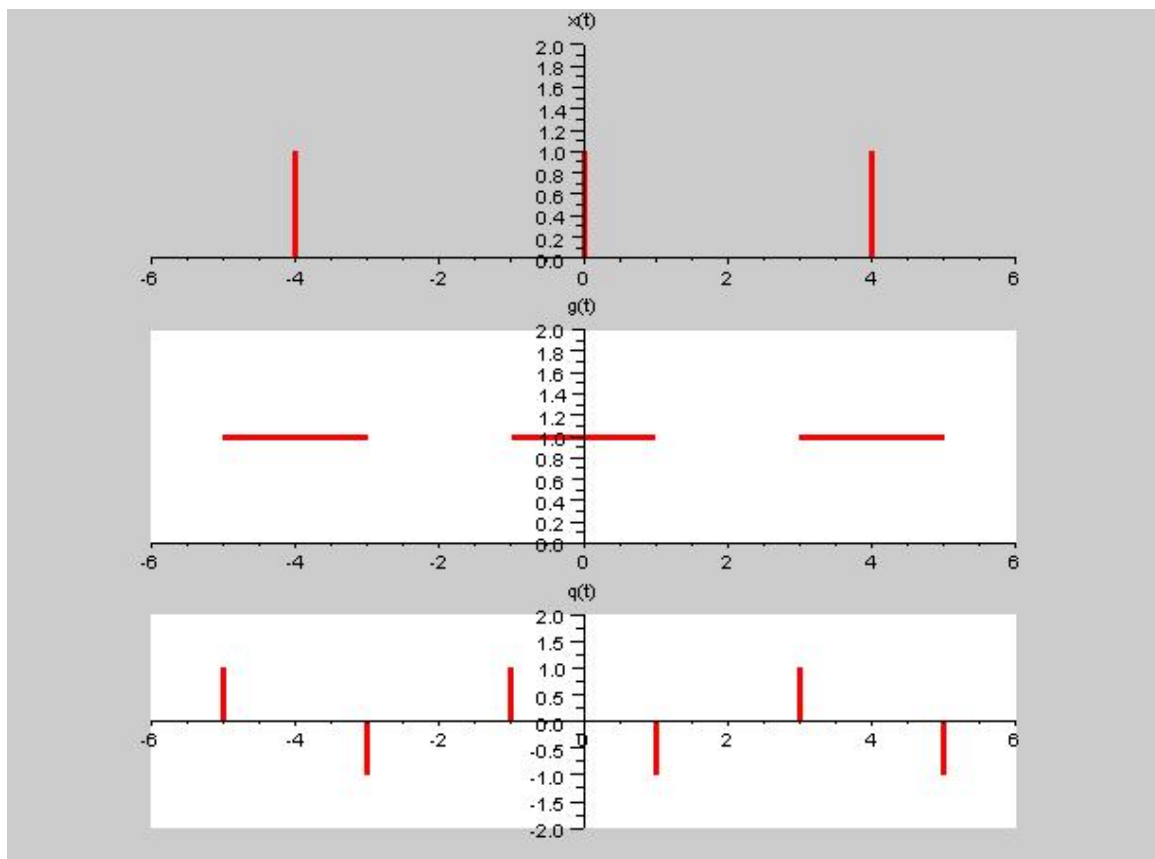


Figure 3.11: Results of Exa 3.8

**Example 3.10** DTFS of  $x(n) = \sin(W_0 n)$

```

1 //Example3.10:DTFS of x[n] =sin(Won)
2 clear;
3 close;
4 clc;
5 n = 0:0.01:5;
6 N = 5;
7 Wo = 2*%pi/N;
8 xn = sin(Wo*n);
9 for k =0:N-2
10     C(k+1,:) = exp(-sqrt(-1)*Wo*n.*k);

```

```

11     a(k+1) = xn*C(k+1,:)'/length(n);
12     if(abs(a(k+1))<=0.01)
13         a(k+1)=0;
14     end
15 end
16 a =a'
17 a_conj = conj(a);
18 ak = [a_conj($:-1:1),a(2:$)]
19 k = -(N-2):(N-2);
20 //
21 figure
22 a = gca();
23 a.y_location = "origin";
24 a.x_location = "origin";
25 a.data_bounds=[-8,-1;8,1];
26 poly1 = a.children(1).children(1);
27 poly1.thickness = 3;
28 plot2d3('gnn',k,-imag(ak),5)
29 poly1 = a.children(1).children(1);
30 poly1.thickness = 3;
31 plot2d3('gnn',N+k,-imag(ak),5)
32 poly1 = a.children(1).children(1);
33 poly1.thickness = 3;
34 plot2d3('gnn',-(N+k),-imag(ak($:-1:1)),5)
35 poly1 = a.children(1).children(1);
36 poly1.thickness = 3;
37 title('ak')

```

---

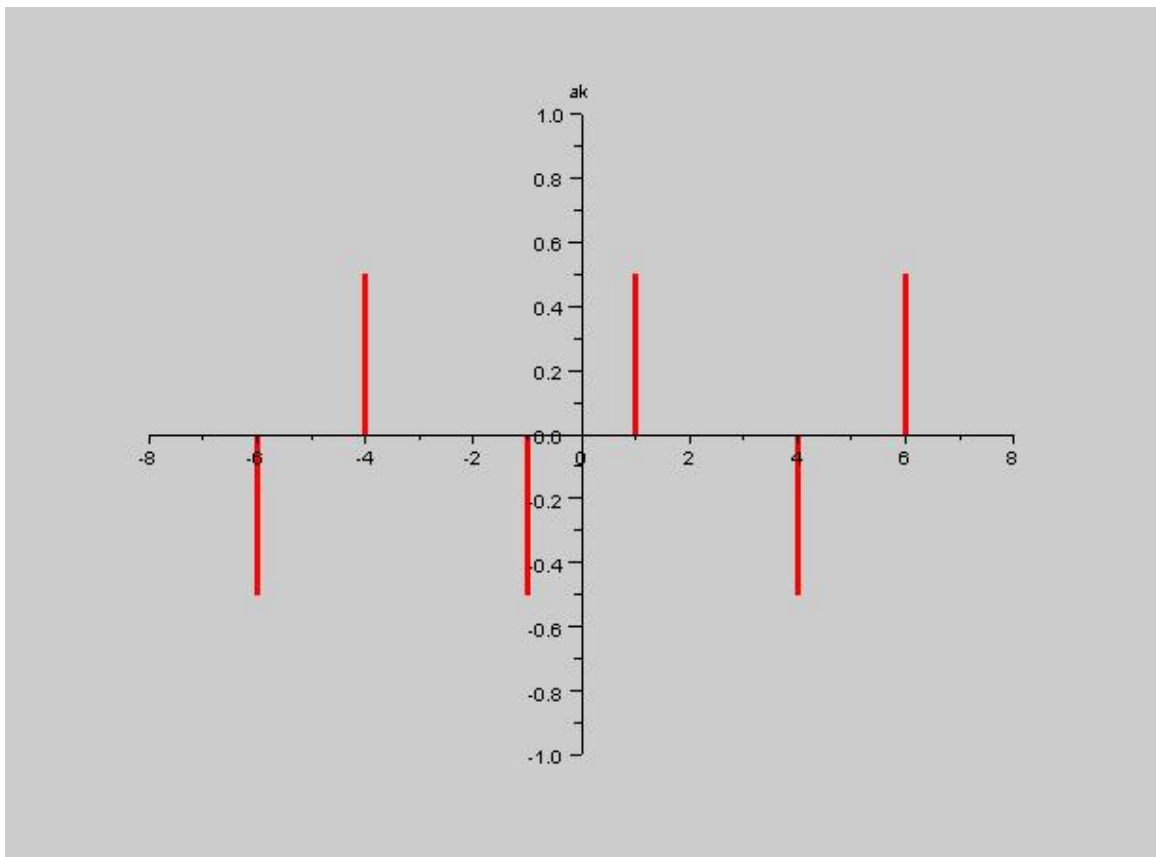


Figure 3.12: Results of Exa 3.10

**Example 3.11** DTFS of  $x(n) = 1 + \sin(2 * \pi / N) * n + 3 * \cos(2 * \pi / N) * n + \cos((4 * \pi / N) * n + \pi / 2)$

```

1 //Example3.11:DTFS of
2 //x[n] = 1+sin(2*%pi/N)n+3cos(2*%pi/N)n+cos[(4*%pi/N
   )n+%pi/2]
3 clear;
4 close;
5 clc;
6 N = 10;
7 n = 0:0.01:N;
8 Wo = 2*%pi/N;
```

```

9  xn =ones(1,length(n))+sin(Wo*n)+3*cos(Wo*n)+cos(2*Wo
    *n+%pi/2);
10 for k =0:N-2
11     C(k+1,:) = exp(-sqrt(-1)*Wo*n.*k);
12     a(k+1) = xn*C(k+1,:)/length(n);
13     if(abs(a(k+1))<=0.1)
14         a(k+1)=0;
15     end
16 end
17 a =a';
18 a_conj =conj(a);
19 ak = [a_conj($:-1:1),a(2:$)];
20 Mag_ak = abs(ak);
21 for i = 1:length(a)
22     Phase_ak(i) = atan(imag(ak(i))/(real(ak(i))
        +0.0001));
23 end
24 Phase_ak = Phase_ak';
25 Phase_ak = [Phase_ak(1:$-1) -Phase_ak($:-1:1)];
26 k = -(N-2):(N-2);
27 //
28 figure
29 subplot(2,1,1)
30 a = gca();
31 a.y_location = "origin";
32 a.x_location = "origin";
33 plot2d3('gnn',k,real(ak),5)
34 poly1 = a.children(1).children(1);
35 poly1.thickness = 3;
36 title('Real part of(ak)')
37 xlabel('

    k')
38 subplot(2,1,2)
39 a = gca();
40 a.y_location = "origin";
41 a.x_location = "origin";
42 plot2d3('gnn',k,imag(ak),5)

```

```

43 poly1 = a.children(1).children(1);
44 poly1.thickness = 3;
45 title('imaginary part of(ak)')
46 xlabel('

    k')
47 //
48 figure
49 subplot(2,1,1)
50 a = gca();
51 a.y_location = "origin";
52 a.x_location = "origin";
53 plot2d3('gnn',k,Mag_ak,5)
54 poly1 = a.children(1).children(1);
55 poly1.thickness = 3;
56 title('abs(ak)')
57 xlabel('

    k')
58 subplot(2,1,2)
59 a = gca();
60 a.y_location = "origin";
61 a.x_location = "origin";
62 plot2d3('gnn',k,Phase_ak,5)
63 poly1 = a.children(1).children(1);
64 poly1.thickness = 3;
65 title('<(ak)')
66 xlabel('

    k')

```

---

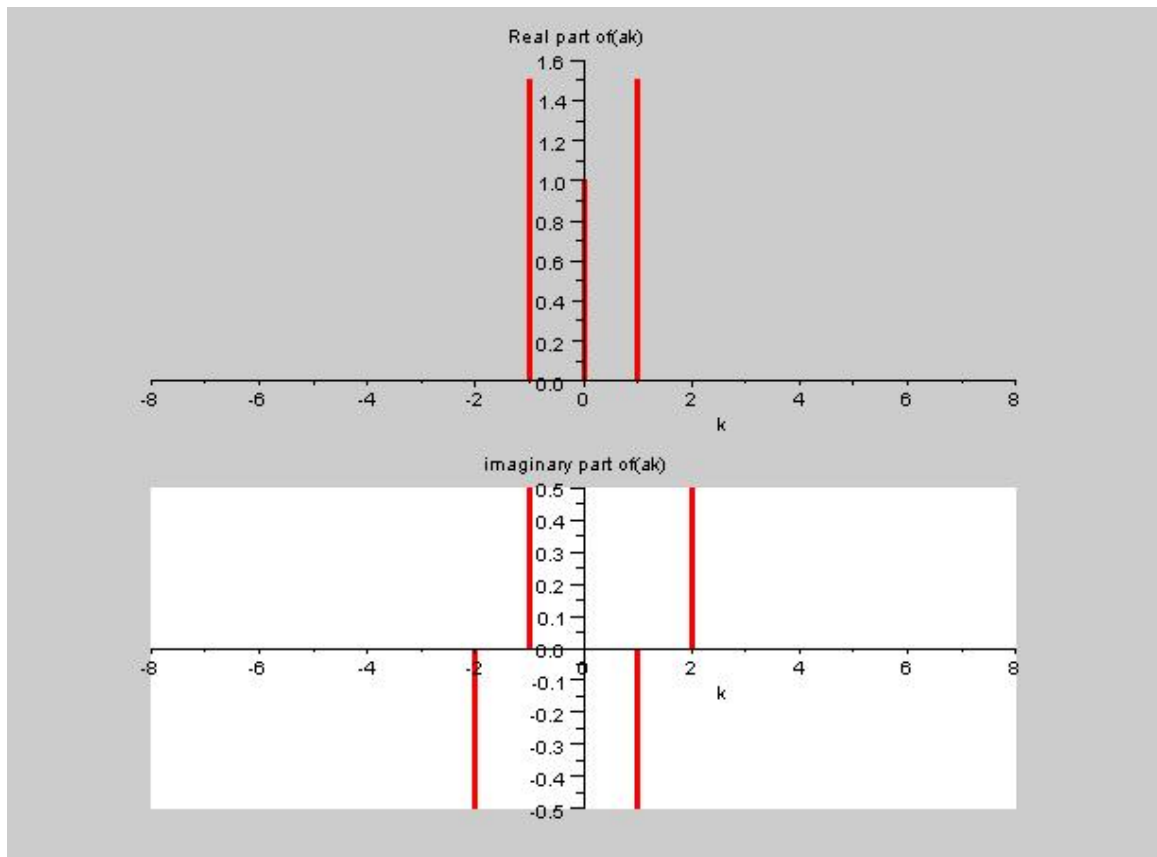


Figure 3.13: Results of Exa [3.11](#)



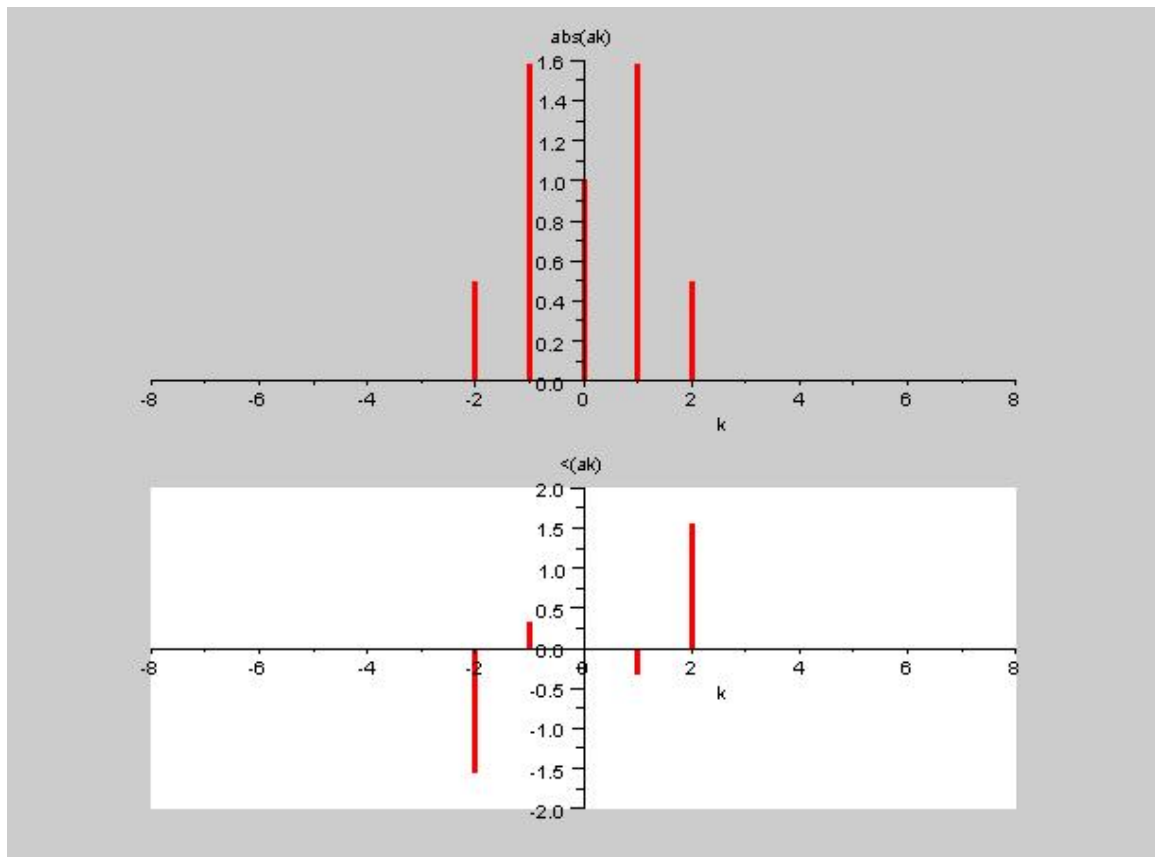


Figure 3.14: Results of Exa 3.11

**Example 3.12** DTFS coefficients of periodic square wave

```

1 //Example3.12:DTFS coefficients of periodic square
  wave
2 clear;
3 close;
4 clc;
5 N = 10;
6 N1 = 2;
7 Wo = 2*%pi/N;
8 xn = ones(1,length(N));
9 n = -(2*N1+1):(2*N1+1);

```

```

10 a(1) = (2*N1+1)/N;
11 for k = 1:2*N1
12     a(k+1) = sin((2*pi*k*(N1+0.5))/N)/sin(pi*k/N);
13     a(k+1) = a(k+1)/N;
14     if(abs(a(k+1))<=0.1)
15         a(k+1) = 0;
16     end
17 end
18 a = a';
19 a_conj = conj(a);
20 ak = [a_conj($:-1:1),a(2:$)];
21 k = -2*N1:2*N1;
22 //
23 figure
24 a = gca();
25 a.y_location = "origin";
26 a.x_location = "origin";
27 plot2d3('gnn',k,real(ak),5)
28 poly1 = a.children(1).children(1);
29 poly1.thickness = 3;
30 title('Real part of(ak)')
31 xlabel('

    k')

```

---

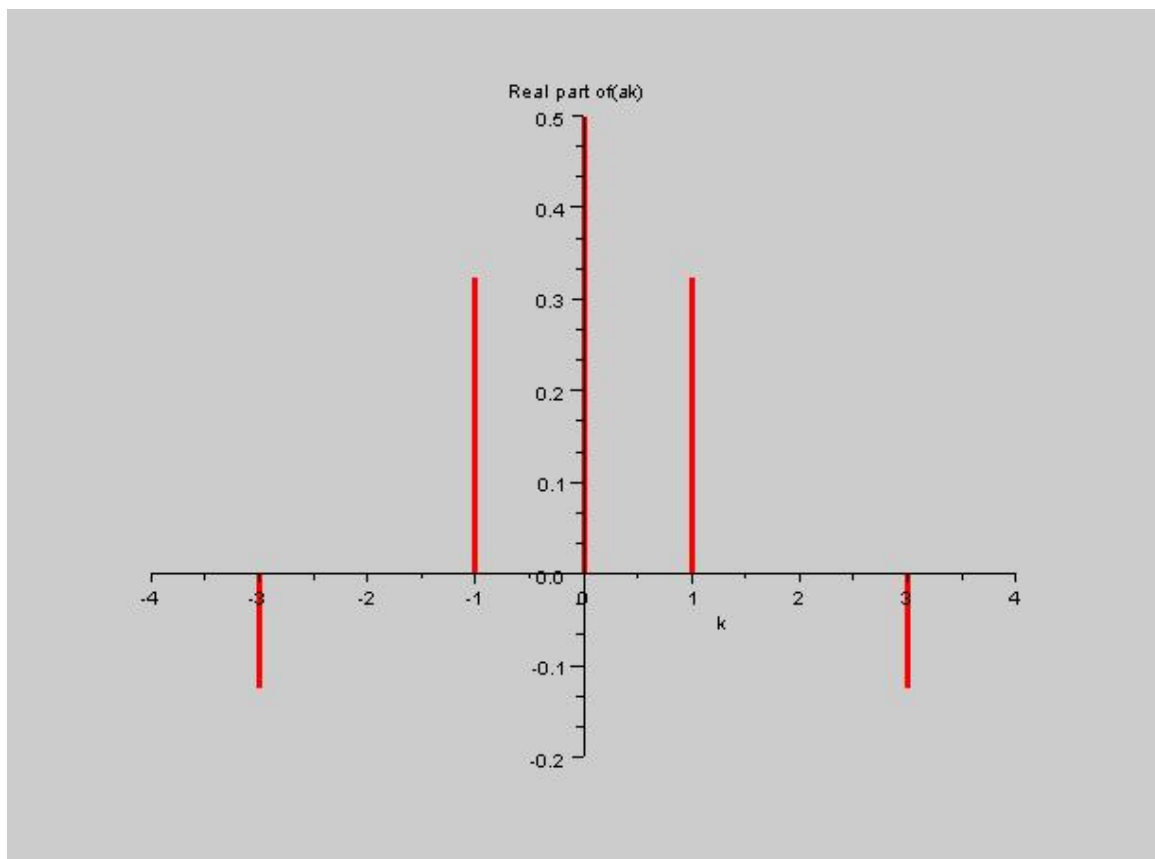


Figure 3.15: Results of Exa 3.12

**Example 3.13** TFS: Expression of periodic sequence using

```

1 //Example3.13:DTFS
2 //Expression of periodic sequence using
3 //the summation two different sequence
4 clear;
5 close;
6 clc;
7 N = 5;
8 n = 0:N-1;
9 x1 = [1,1,0,0,1];
10 x1 = [x1($:-1:1) x1(2:$)]; // Square Wave x1[n]

```

```

11 x2 = [1,1,1,1,1];
12 x2 = [x2($:-1:1) x2(2:$)]; //DC sequence of x2[n]
13 x = x1+x2; //sum of x1[n] & x2[n]
14 //Zeroth DTFS coefficient of dc sequence
15 c(1) = 1;
16 //Zeroth DTFS coefficient of square waveform
17 b(1) = 3/5;
18 //Zeroth DTFS coefficient of sum of x1[n] & x2[n]
19 a(1) = b(1)+c(1);
20 //
21 Wo = 2*%pi/N;
22 for k = 1:N-1
23     a(k+1) = sin((3*%pi*k)/N)/sin(%pi*k/N);
24     a(k+1) = a(k+1)/N;
25     if(abs(a(k+1))<=0.1)
26         a(k+1) = 0;
27     end
28 end
29 a = a';
30 a_conj = conj(a);
31 ak = [a_conj($:-1:1),a(2:$)];
32 k = -(N-1):(N-1);
33 n = -(N-1):(N-1);
34 //
35 figure
36 subplot(3,1,1)
37 a = gca();
38 a.y_location = "origin";
39 a.x_location = "origin";
40 plot2d3('gnn',n,x,5)
41 poly1 = a.children(1).children(1);
42 poly1.thickness = 3;
43 title('x[n]')
44 xlabel('

    n')
45 subplot(3,1,2)
46 a = gca();

```

```

47 a.y_location = "origin";
48 a.x_location = "origin";
49 plot2d3('gnn',n,x1,5)
50 poly1 = a.children(1).children(1);
51 poly1.thickness = 3;
52 title('x1[n]')
53 xlabel('

        n')
54 subplot(3,1,3)
55 a = gca();
56 a.y_location = "origin";
57 a.x_location = "origin";
58 plot2d3('gnn',n,x2,5)
59 poly1 = a.children(1).children(1);
60 poly1.thickness = 3;
61 title('x2[n]')
62 xlabel('

        n')

```

---

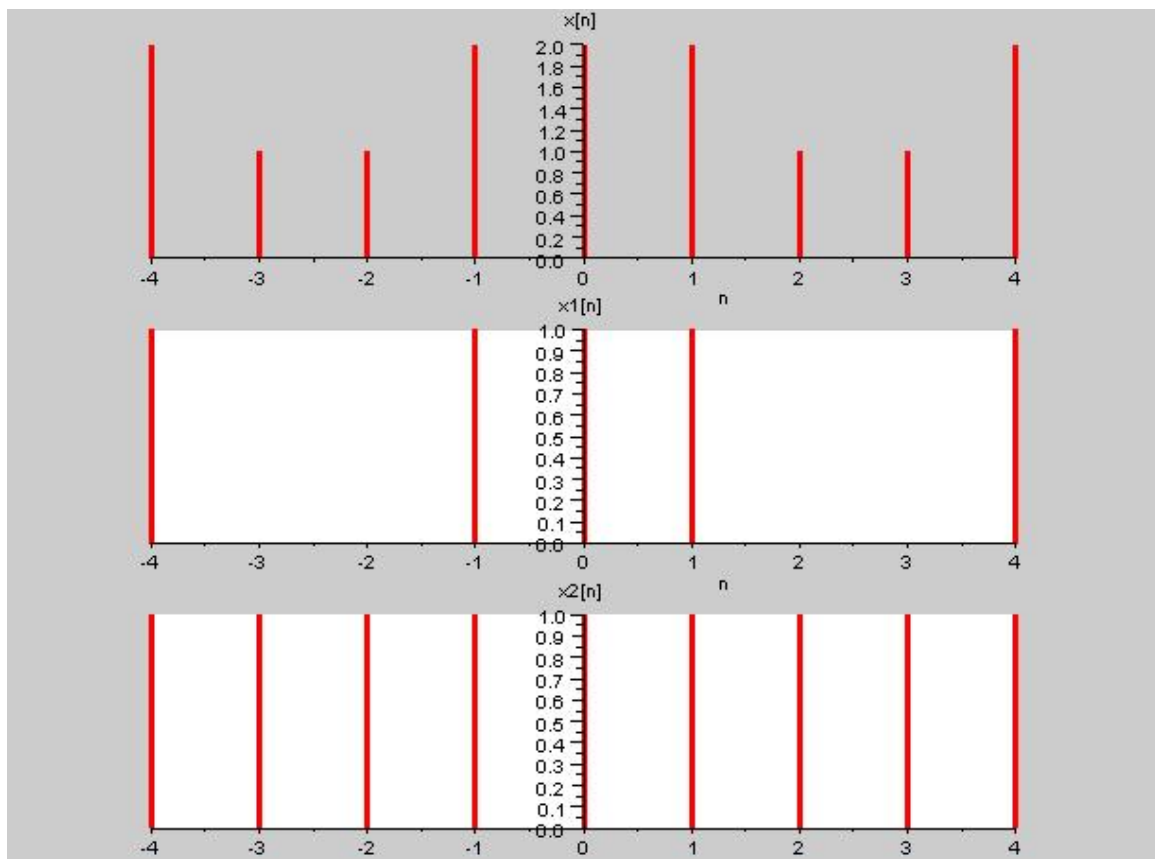


Figure 3.16: Results of Exa 3.13

**Example 3.14** DTFS: Finding  $x[n]$  using parseval's relation of DTFS

```

1 //Example3.14:DTFS
2 //Finding x[n] using parseval's relation of DTFS
3 clear;
4 close;
5 clc;
6 N = 6;
7 n = 0:N-1;
8 a(1) = 1/3;
9 a(2)=0;
10 a(4)=0;
```

```

11 a(5)=0;
12 a1 = (1/6)*((-1)^n);
13 x =0;
14 for k = 0:N-2
15     if(k==2)
16         x = x+a1;
17     else
18         x = x+a(k+1);
19     end
20 end
21 x = [x($:-1:1),x(2:$)];
22 n = -(N-1):(N-1);
23 //
24 figure
25 a = gca();
26 a.y_location = "origin";
27 a.x_location = "origin";
28 plot2d3('gnn',n,x,5)
29 poly1 = a.children(1).children(1);
30 poly1.thickness = 3;
31 title('x[n]')
32 xlabel('

```

---

n')

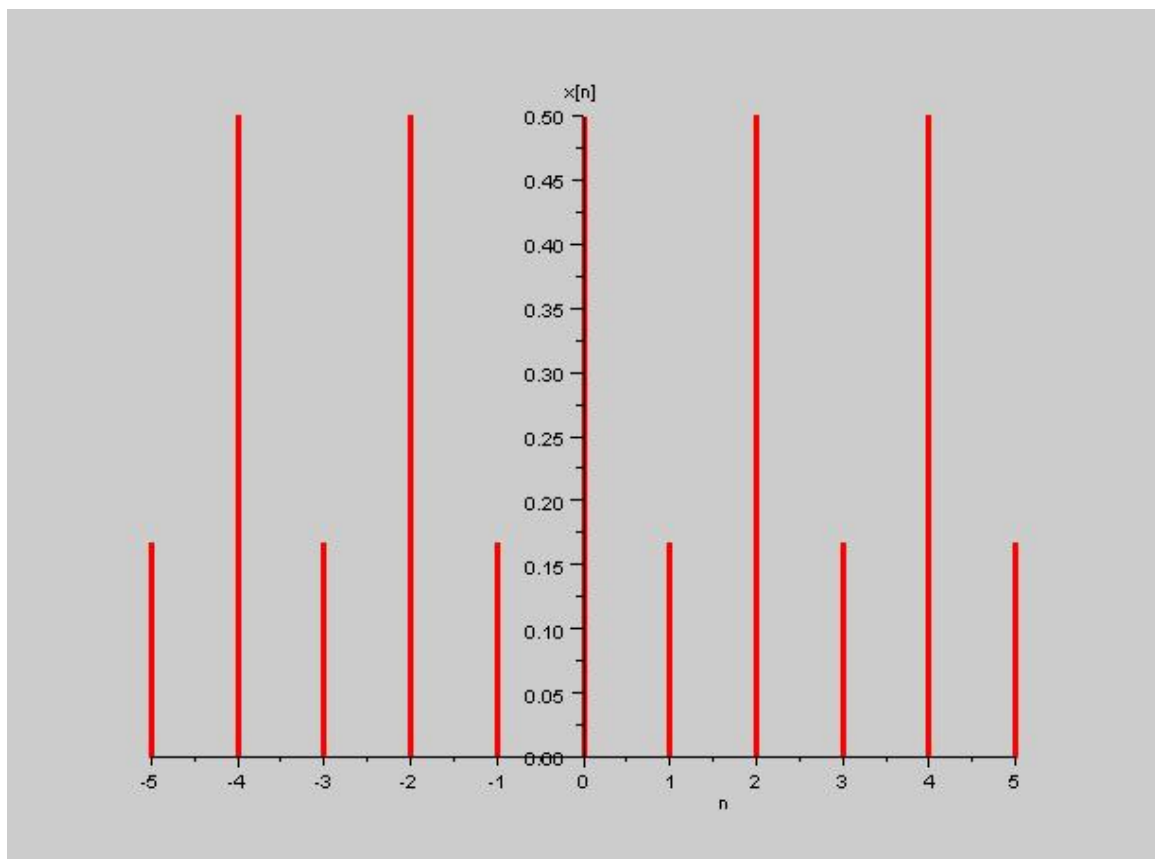


Figure 3.17: Results of Exa 3.14

**Example 3.15** DTFS: Periodic Convolution Property

```

1 //Example3.15:DTFS: Periodic Convolution Property
2 clear;
3 clc;
4 close;
5 x = [1,1,0,0,0,0,1];
6 X = fft(x);
7 W = X.*X;
8 w = ifft(W);
9 w = abs(w);
10 for i =1:length(x)

```



```

11     if (abs(w(i)) <= 0.1)
12         w(i) = 0;
13     end
14 end
15 w = [w($:-1:1) w(2:$)];
16 N = length(x);
17 figure
18 a = gca();
19 a.y_location = "origin";
20 a.x_location = "origin";
21 plot2d3('gnn',[-(N-1):0,1:N-1],w,5)
22 poly1 = a.children(1).children(1);
23 poly1.thickness = 3;
24 title('w[n]')
25 xlabel('

```

---

```

        n')

```

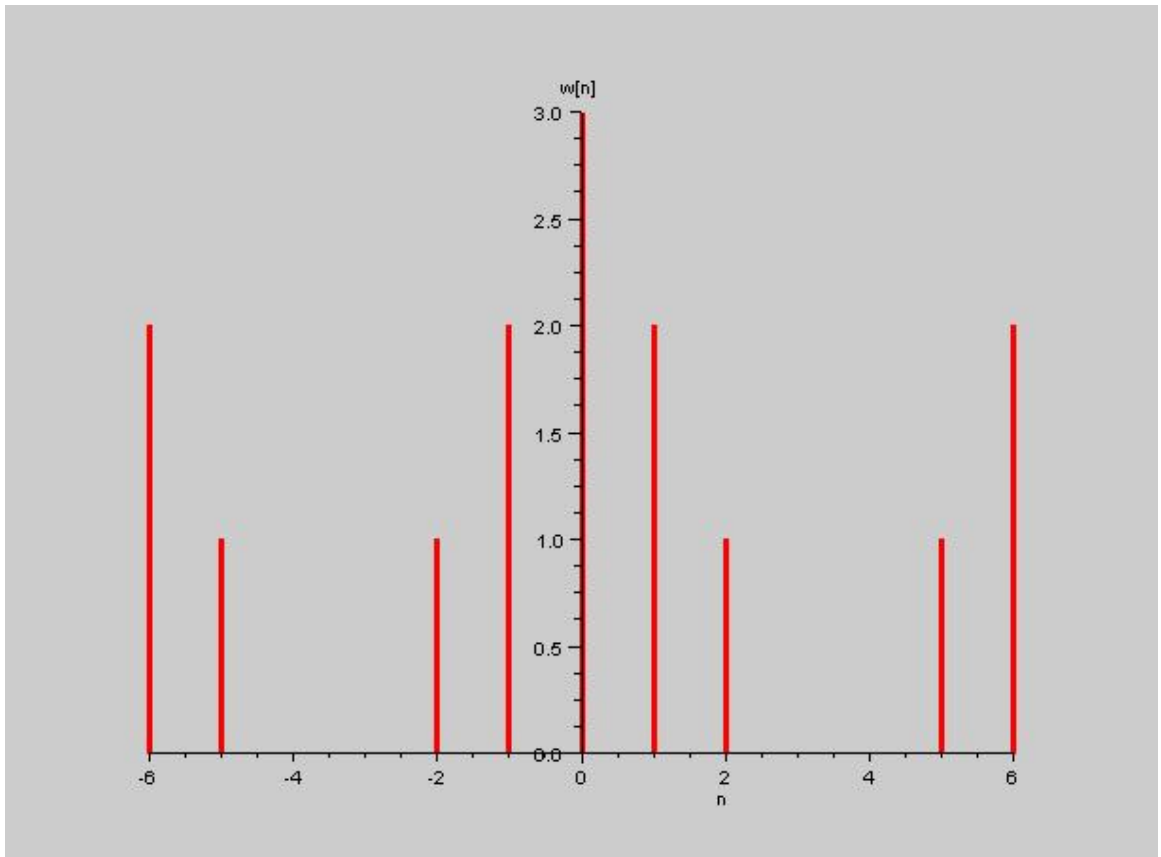


Figure 3.18: Results of Exa [3.15](#)

# Chapter 4

## The Continuous Time Fourier Transform

### 4.1 Scilab Codes

**Example 4.1** Continuous Time Fourier Transform of a Continuous Time Signal  $x(t) = \exp(-A * t)u(t), t > 0$

```
1 //Example 4.1: Continuous Time Fourier Transform of a
2 //Continuous Time Signal x(t)= exp(-A*t)u(t), t>0
3 clear;
4 clc;
5 close;
6 // Analog Signal
7 A =1; //Amplitude
8 Dt = 0.005;
9 t = 0:Dt:10;
10 xt = exp(-A*t);
11 //
12 // Continuous-time Fourier Transform
13 Wmax = 2*%pi*1; //Analog Frequency = 1Hz
14 K = 4;
15 k = 0:(K/1000):K;
16 W = k*Wmax/K;
17 XW = xt* exp(-sqrt(-1)*t'*W) * Dt;
```

```

18 XW_Mag = abs(XW);
19 W = [-mtlbfliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
20 XW_Mag = [mtlbfliplr(XW_Mag), XW_Mag(2:1001)];
21 [XW_Phase,db] = phasemag(XW);
22 XW_Phase = [-mtlbfliplr(XW_Phase),XW_Phase(2:1001)
    ];
23 //Plotting Continuous Time Signal
24 figure
25 a = gca();
26 a.y_location = "origin";
27 plot(t,xt);
28 xlabel('t in sec. ');
29 ylabel('x(t)')
30 title('Continuous Time Signal')
31 figure
32 //Plotting Magnitude Response of CTS
33 subplot(2,1,1);
34 a = gca();
35 a.y_location = "origin";
36 plot(W,XW_Mag);
37 xlabel('Frequency in Radians/Seconds—> W');
38 ylabel('abs(X(jW))')
39 title('Magnitude Response (CTFT)')
40 //Plotting Phase Reponse of CTS
41 subplot(2,1,2);
42 a = gca();
43 a.y_location = "origin";
44 a.x_location = "origin";
45 plot(W,XW_Phase*%pi/180);
46 xlabel('Frequency in
    Radians/Seconds—> W');
47 ylabel('
    (jW)')
48 title('Phase Response(CTFT) in Radians')

```

---

<X

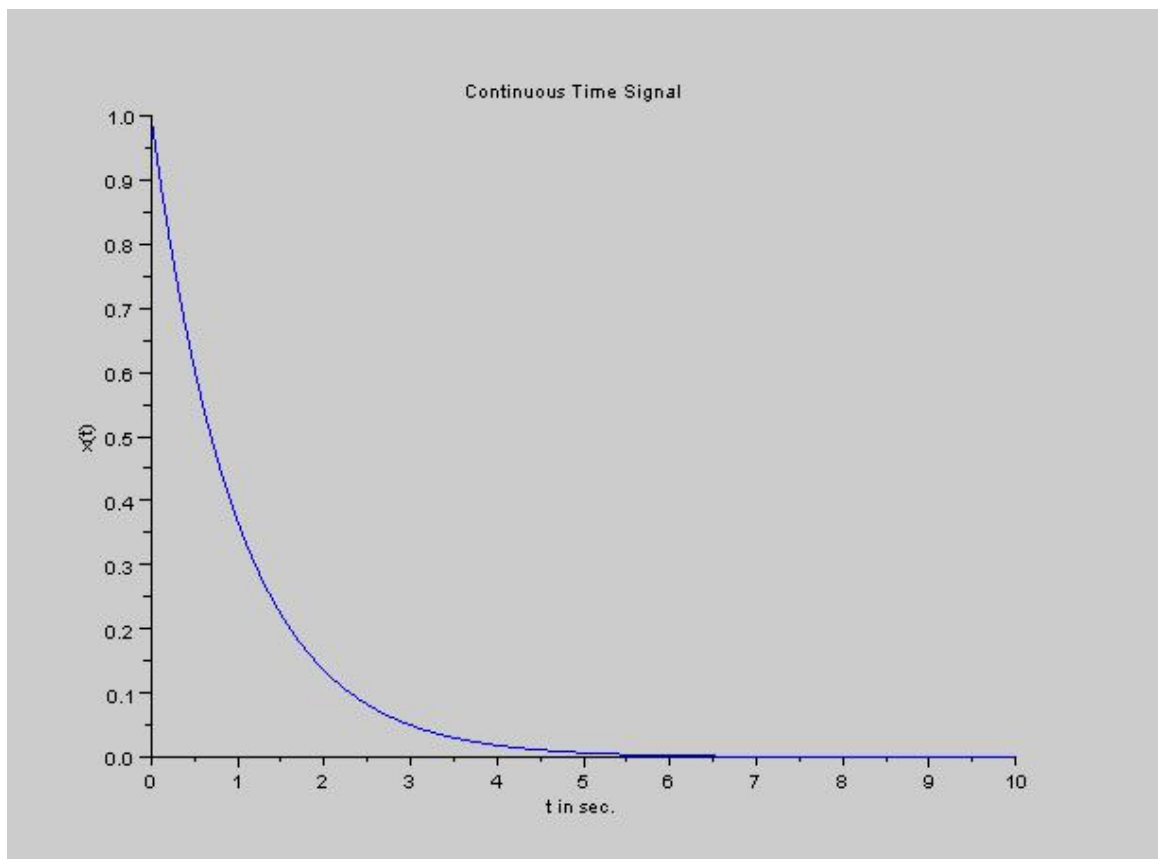


Figure 4.1: Results of Exa [4.1](#)

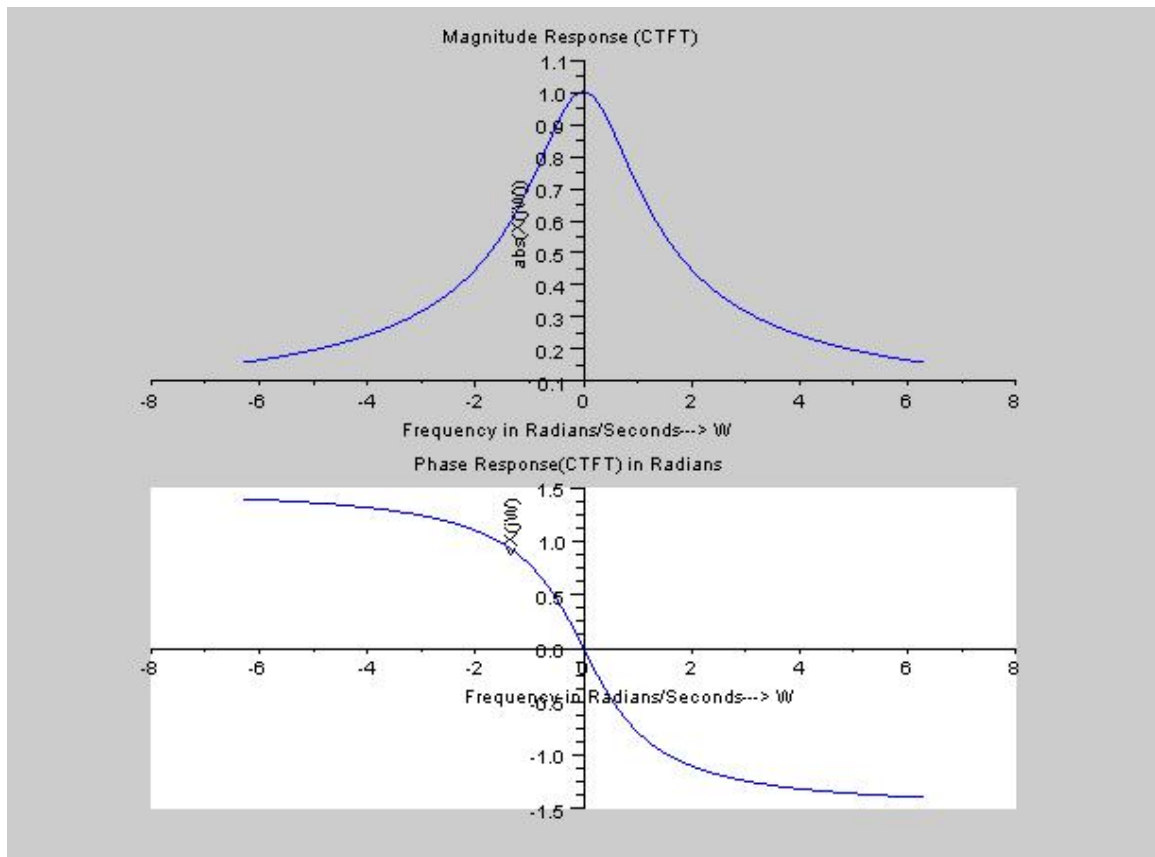


Figure 4.2: Results of Exa 4.1

**Example 4.2** Continuous Time Fourier Transform of a Continuous Time Signal  $x(t) = \exp(-A * \text{abs}(t))$

```

1 //Example 4.2:Continuous Time Fourier Transform of a
2 //Continuous Time Signal x(t)= exp(-A*abs(t))
3 clear;
4 clc;
5 close;
6 // Analog Signal
7 A =1; //Amplitude
8 Dt = 0.005;
9 t = -4.5:Dt:4.5;

```

```

10 xt = exp(-A*abs(t));
11 //
12 // Continuous-time Fourier Transform
13 Wmax = 2*pi*1;           //Analog Frequency = 1Hz
14 K = 4;
15 k = 0:(K/1000):K;
16 W = k*Wmax/K;
17 XW = xt* exp(-sqrt(-1)*t'*W) * Dt;
18 XW = real(XW);
19 W = [-mtlbfliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
20 XW = [mtlbfliplr(XW), XW(2:1001)];
21 subplot(1,1,1)
22 subplot(2,1,1);
23 a = gca();
24 a.y_location = "origin";
25 plot(t,xt);
26 xlabel('t in sec. ');
27 ylabel('x(t)')
28 title('Continuous Time Signal')
29 subplot(2,1,2);
30 a = gca();
31 a.y_location = "origin";
32 plot(W,XW);
33 xlabel('Frequency in Radians/Seconds W');
34 ylabel('X(jW)')
35 title('Continuous-time Fourier Transform')

```

---

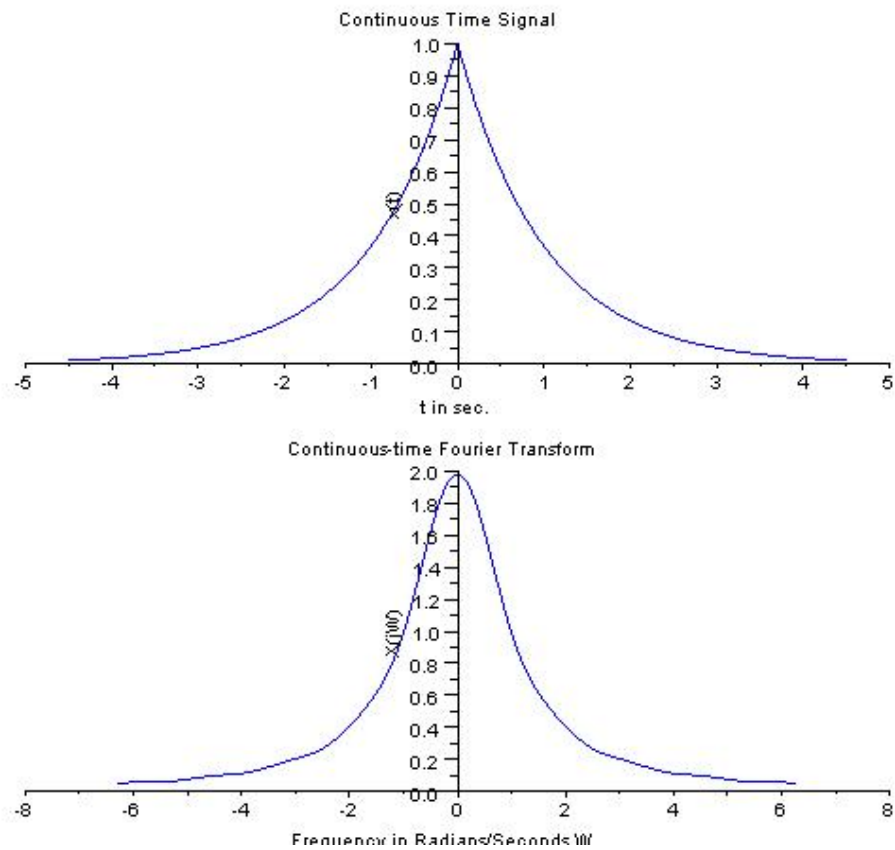


Figure 4.3: Results of Exa 4.2

**Example 4.4** Continuous Time Fourier Transform and Frequency Response of a Square Waveform  $x(t) = A$ , from  $-T_1$  to  $T_1$

```

1 //Example 4.4
2 // Continuous Time Fourier Transform
3 //and Frequency Response of a Square Waveform
4 // x(t)= A, from -T1 to T1
5 clear;
6 clc;
7 close;
8 // CTS Signal
9 A =1; //Amplitude

```



```

10 Dt = 0.005;
11 T1 = 4; //Time in seconds
12 t = -T1/2:Dt:T1/2;
13 for i = 1:length(t)
14     xt(i) = A;
15 end
16 //
17 // Continuous-time Fourier Transform
18 Wmax = 2*pi*1; //Analog Frequency = 1Hz
19 K = 4;
20 k = 0:(K/1000):K;
21 W = k*Wmax/K;
22 xt = xt';
23 XW = xt* exp(-sqrt(-1)*t'*W) * Dt;
24 XW_Mag = real(XW);
25 W = [mtlbfliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
26 XW_Mag = [mtlbfliplr(XW_Mag), XW_Mag(2:1001)];
27 //
28 subplot(2,1,1);
29 a = gca();
30 a.data_bounds=[-4,0;4,2];
31 a.y_location = "origin";
32 plot(t,xt);
33 xlabel('t in msec. ');
34 title('Contiuous Time Signal x(t)')
35 subplot(2,1,2);
36 a = gca();
37 a.y_location = "origin";
38 plot(W,XW_Mag);
39 xlabel('Frequency in Radians/Seconds');
40 title('Continuous-time Fourier Transform X(jW)')

```

---

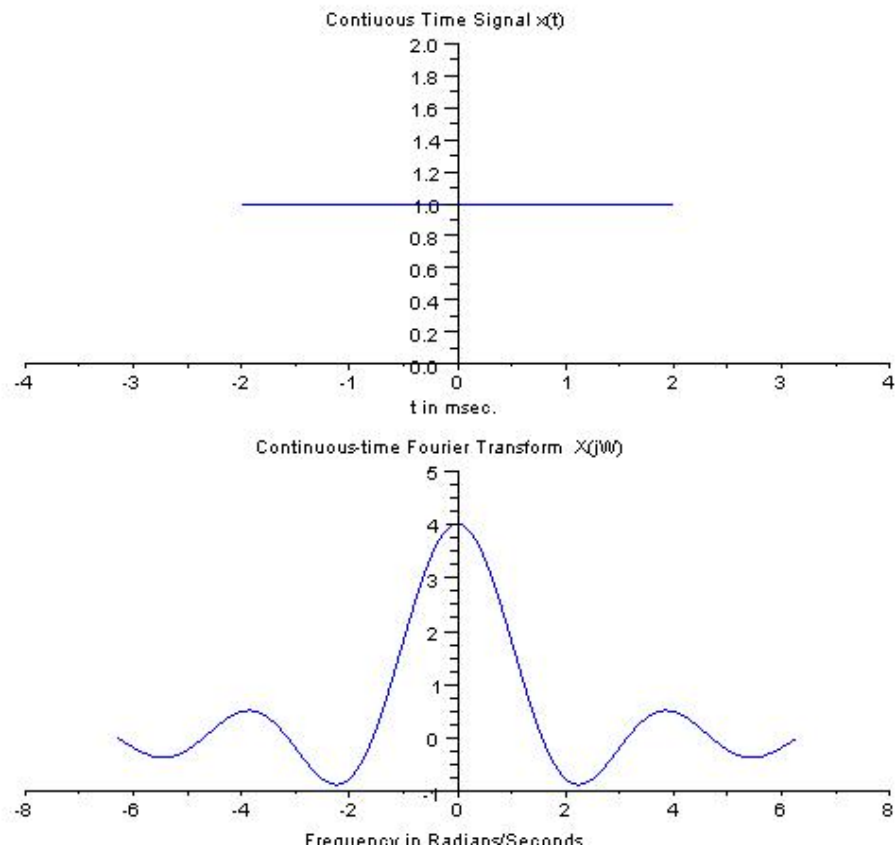


Figure 4.4: Results of Exa 4.4

**Example 4.5** Inverse Continuous Time Fourier Transform  $X(jW) = 1$ , from  $-T_1$  to  $T_1$

```

1 //Example 4.5
2 // Inverse Continuous Time Fourier Transform
3 // X(jW)= 1, from -T1 to T1
4 clear;
5 clc;
6 close;
7 // CTFT
8 A =1; //Amplitude
9 Dw = 0.005;
```

```

10 W1 = 4; //Time in seconds
11 w = -W1/2:Dw:W1/2;
12 for i = 1:length(w)
13     XW(i) = A;
14 end
15 XW = XW';
16 //
17 //Inverse Continuous-time Fourier Transform
18 t = -%pi:%pi/length(w):%pi;
19 xt = (1/(2*%pi))*XW *exp(sqrt(-1)*w'*t)*Dw;
20 xt = real(xt);
21 figure
22 a = gca();
23 a.y_location = "origin";
24 a.x_location = "origin";
25 plot(t,xt);
26 xlabel('                                t time
        in Seconds');
27 title('Inverse Continuous Time Fourier Transform x(t
        )')

```

---

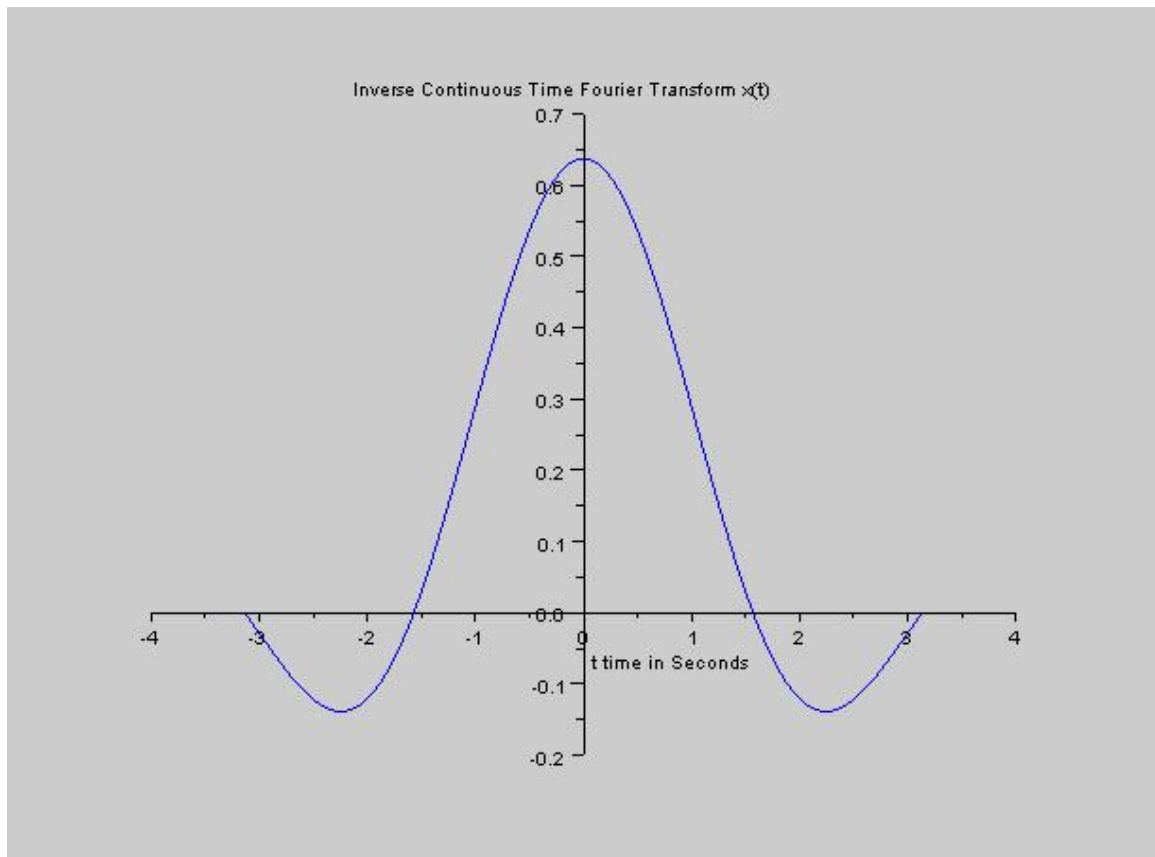


Figure 4.5: Results of Exa 4.5

**Example 4.6** Continuous Time Fourier Transform of Symmetric periodic Square waveform

```

1 //Example 4.6
2 // Continuous Time Fourier Transform of Symmetric
3 // periodic Square waveform
4 clear;
5 clc;
6 close;
7 // CTFT
8 T1 = 2;
9 T = 4*T1;

```

```

10 Wo = 2*%pi/T;
11 W = -%pi:Wo:%pi;
12 delta = ones(1,length(W));
13 XW(1) = (2*%pi*Wo*T1/%pi);
14 mid_value = ceil(length(W)/2);
15 for k = 2:mid_value
16     XW(k) = (2*%pi*sin((k-1)*Wo*T1)/(%pi*(k-1)));
17 end
18 figure
19 a = gca();
20 a.y_location = "origin";
21 a.x_location = "origin";
22 plot2d3('gnn',W(mid_value:$),XW,2);
23 poly1 = a.children(1).children(1);
24 poly1.thickness = 3;
25 plot2d3('gnn',W(1:mid_value-1),XW($:-1:2),2);
26 poly1 = a.children(1).children(1);
27 poly1.thickness = 3;
28 xlabel('W in radians/Seconds');
29 title('Continuous Time Fourier Transform of Periodic
        Square Wave')

```

---

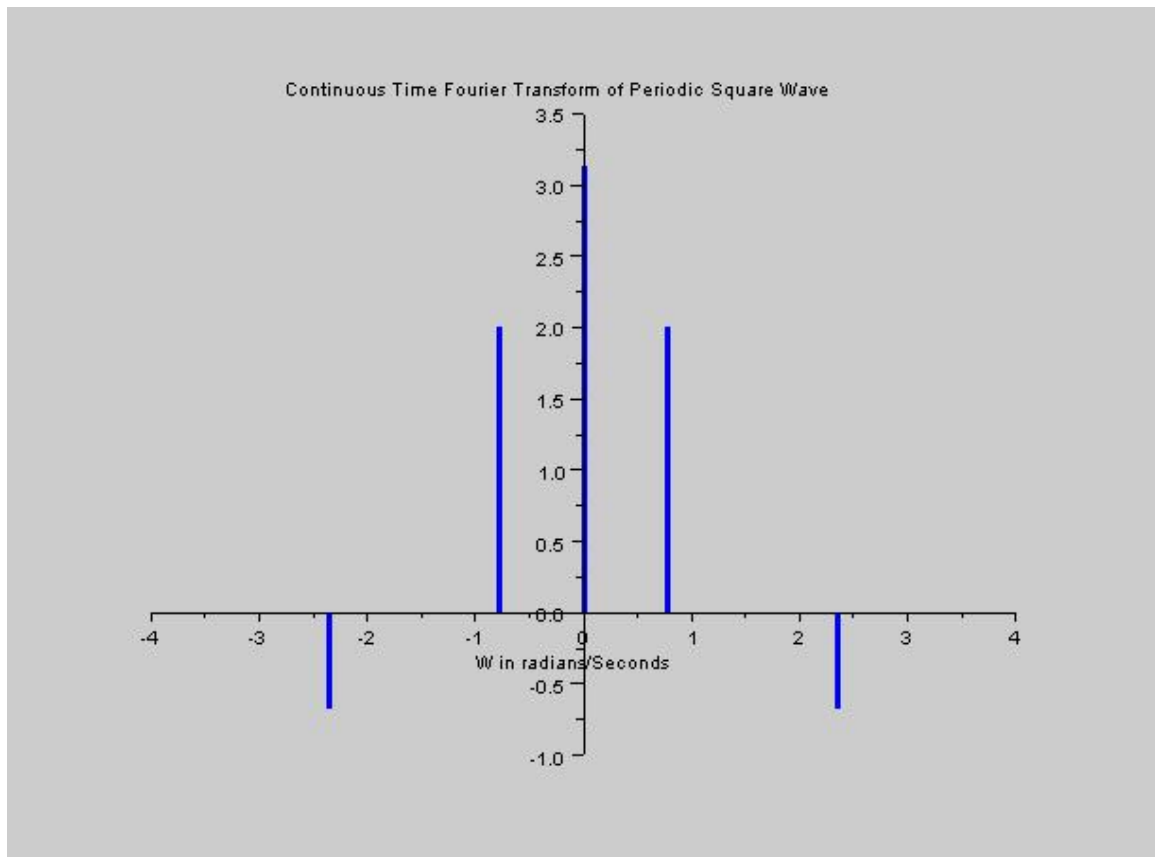


Figure 4.6: Results of Exa 4.6

**Example 4.7** Continuous Time Fourier Transforms of Sinusoidal waveforms  
(a)  $\sin(Wot)$  (b)  $\cos(Wot)$

```

1 //Example 4.7
2 // Continuous Time Fourier Transforms of
3 // Sinusoidal waveforms (a)  $\sin(Wot)$  (b)  $\cos(Wot)$ 
4 clear;
5 clc;
6 close;
7 // CTFT
8 T1 = 2;
9 T = 4*T1;

```

```

10 Wo = 2*%pi/T;
11 W = [-Wo,0,Wo];
12 ak = (2*%pi*Wo*T1/%pi)/sqrt(-1);
13 XW = [-ak,0,ak];
14 ak1 = (2*%pi*Wo*T1/%pi);
15 XW1 =[ak1,0,ak1];
16 //
17 figure
18 a = gca();
19 a.y_location = "origin";
20 a.x_location = "origin";
21 plot2d3('gnn',W,imag(XW),2);
22 poly1 = a.children(1).children(1);
23 poly1.thickness = 3;
24 xlabel('

        W');
25 title('CTFT of sin(Wot)')
26 //
27 figure
28 a = gca();
29 a.y_location = "origin";
30 a.x_location = "origin";
31 plot2d3('gnn',W,XW1,2);
32 poly1 = a.children(1).children(1);
33 poly1.thickness = 3;
34 xlabel('

        W');
35 title('CTFT of cos(Wot)')

```

---

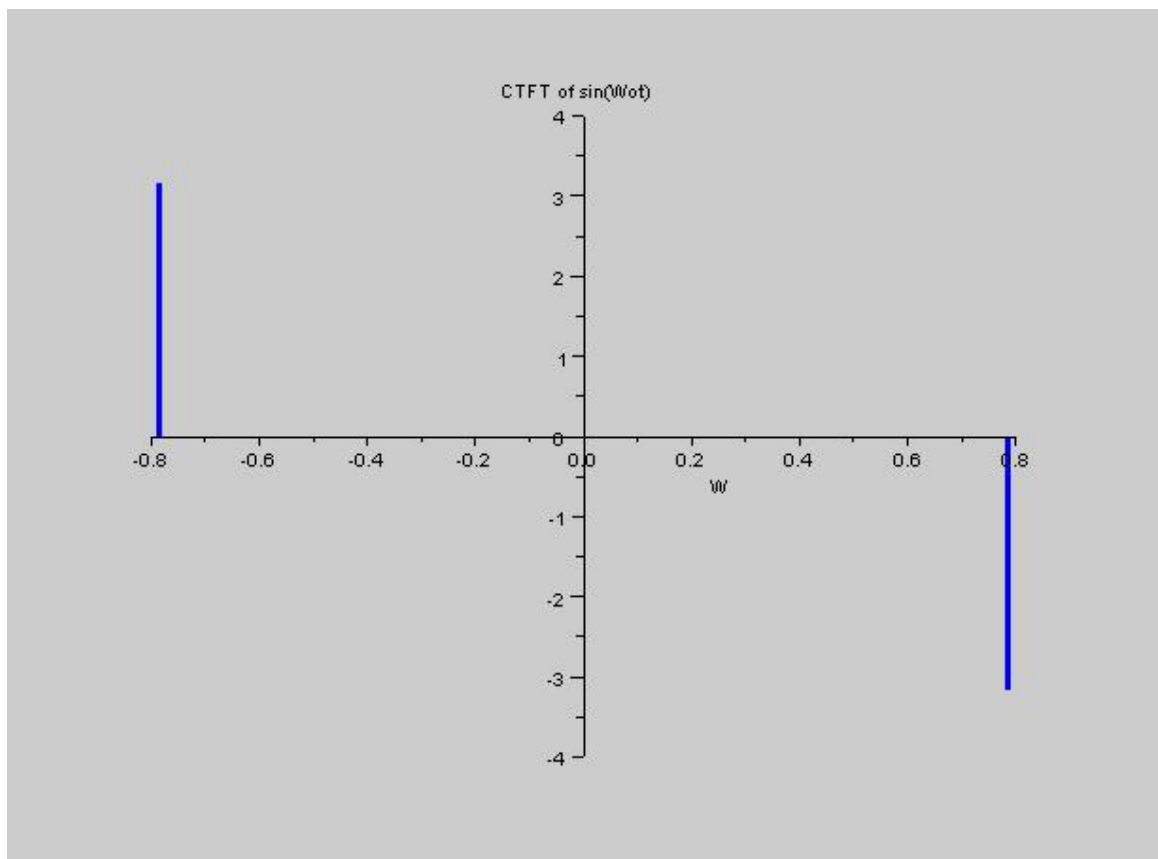


Figure 4.7: Results of Exa [4.7](#)



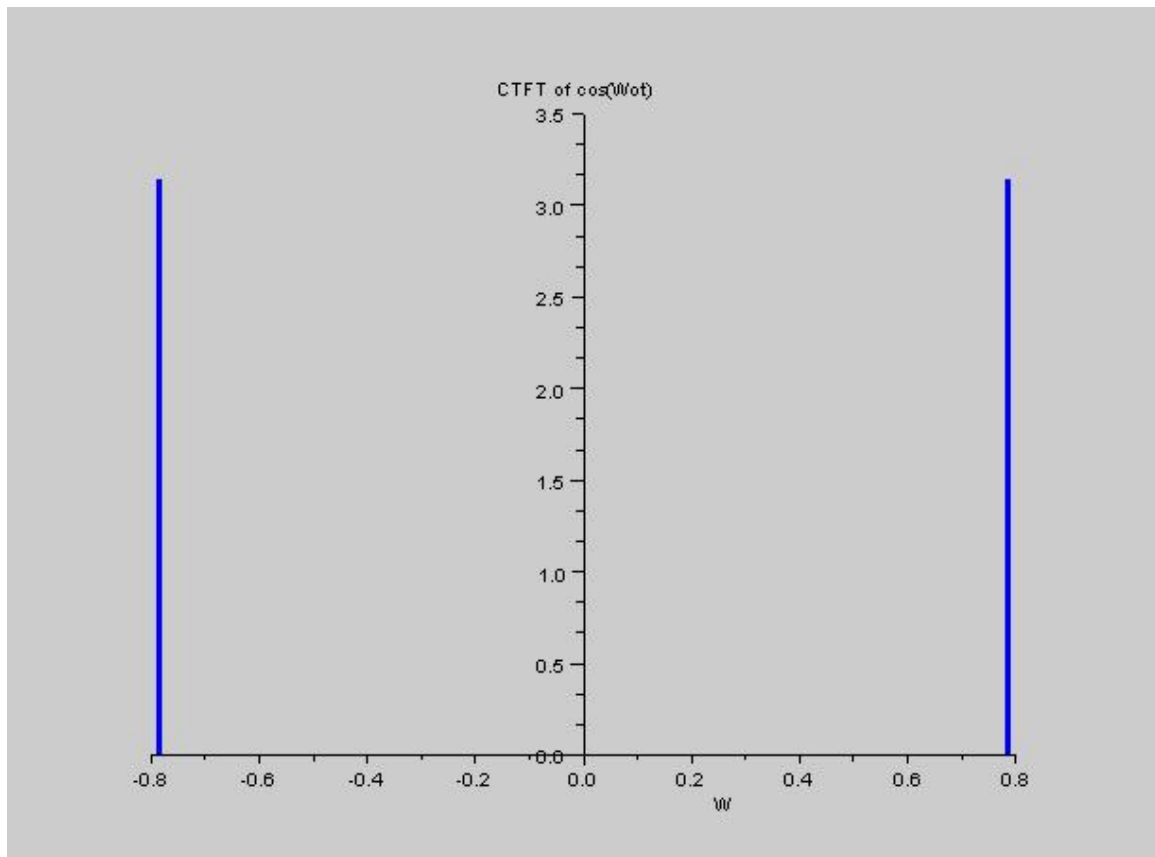


Figure 4.8: Results of Exa 4.7

**Example 4.8** Continuous Time Fourier Transform of Periodic Impulse Train

```

1 //Example 4.8
2 // Continuous Time Fourier Transforms of
3 // Periodic Impulse Train
4 clear;
5 clc;
6 close;
7 // CTFT
8 T = -4:4;;
9 T1 = 1; //Sampling Interval
10 xt = ones(1,length(T));

```

```

11 ak = 1/T1;
12 XW = 2*%pi*ak*ones(1,length(T));
13 Wo = 2*%pi/T1;
14 W = Wo*T;
15 figure
16 subplot(2,1,1)
17 a = gca();
18 a.y_location = "origin";
19 a.x_location = "origin";
20 plot2d3('gnn',T,xt,2);
21 poly1 = a.children(1).children(1);
22 poly1.thickness = 3;
23 xlabel('

        t');
24 title('Periodic Impulse Train')
25 subplot(2,1,2)
26 a = gca();
27 a.y_location = "origin";
28 a.x_location = "origin";
29 plot2d3('gnn',W,XW,2);
30 poly1 = a.children(1).children(1);
31 poly1.thickness = 3;
32 xlabel('

        t');
33 title('CTFT of Periodic Impulse Train')

```

---

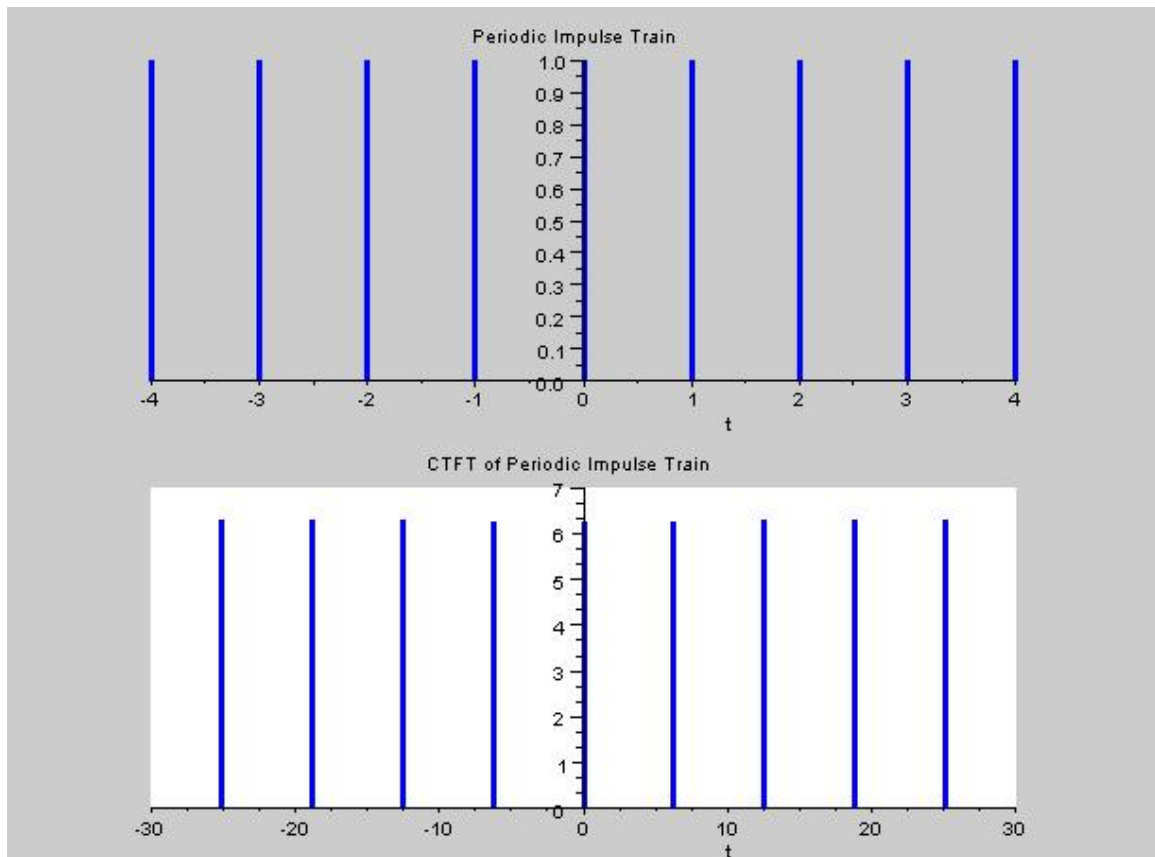


Figure 4.9: Results of Exa 4.8

**Example 4.9** Continuous Time Fourier Transform Properties: Linearity and Time Shift Property

```

1 //Example 4.9:Continuous Time Fourier Transform
  Properties:
2 //Linearity and Time Shift Property
3 clear;
4 clc;
5 close;
6 // CTFT
7 t1 = -1/2:0.1:1/2;
8 t2 = -3/2:0.1:3/2;

```

```

9  x1 = ones(1,length(t1));
10 x2 = ones(1,length(t2));
11 t3 = t1+2.5;
12 t4 = t2+2.5;
13 x1 = (1/2)*x1;
14 x = [x2(1:floor(length(x2)/3)),x1+x2(ceil(length(x2)
    /3):$-floor(length(x2)/3)),x2(($-ceil(length(x2)
    /3))+2:$)];
15 subplot(3,1,1)
16 a = gca();
17 a.x_location = "origin";
18 a.y_location = "origin";
19 plot(t1,x1)
20 xtitle('x1(t)')
21 subplot(3,1,2)
22 a = gca();
23 a.x_location = "origin";
24 a.y_location = "origin";
25 plot(t2,x2)
26 xtitle('x2(t)')
27 subplot(3,1,3)
28 a = gca();
29 a.x_location = "origin";
30 a.y_location = "origin";
31 plot(t4,x)
32 xtitle('x(t)')

```

---

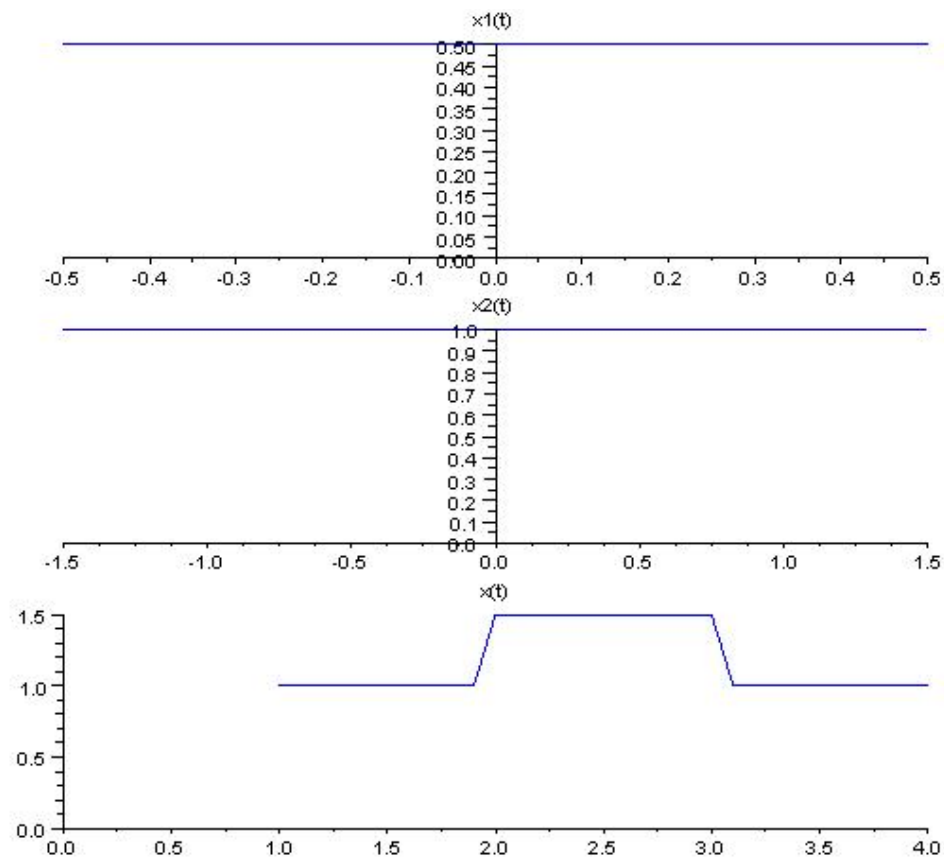


Figure 4.10: Results of Exa 4.9

**Example 4.12** Continuous Time Fourier Transform: Derivative property

```

1 //Example 4.12: Continuous Time Fourier Transform:
2 //Derivative property
3 clear;
4 clc;
5 close;
6 // CTFT
7 t = -1:0.1:1;
8 x1 = ones(1,length(t));
9 x2 = [-1,zeros(1,length(t)-2),-1];
10 x = t;
```

```
11 //differentiation of x can be expressed as
12 //summation of x1 and x2
13 subplot(3,1,1)
14 a = gca();
15 a.x_location = "origin";
16 a.y_location = "origin";
17 plot(t,x1)
18 xtitle('x1(t)')
19 subplot(3,1,2)
20 a = gca();
21 a.x_location = "origin";
22 a.y_location = "origin";
23 plot2d3('gnn',t,x2)
24 xtitle('x2(t)')
25 subplot(3,1,3)
26 a = gca();
27 a.x_location = "origin";
28 a.y_location = "origin";
29 plot(t,x)
30 xtitle('x(t)')
```

---

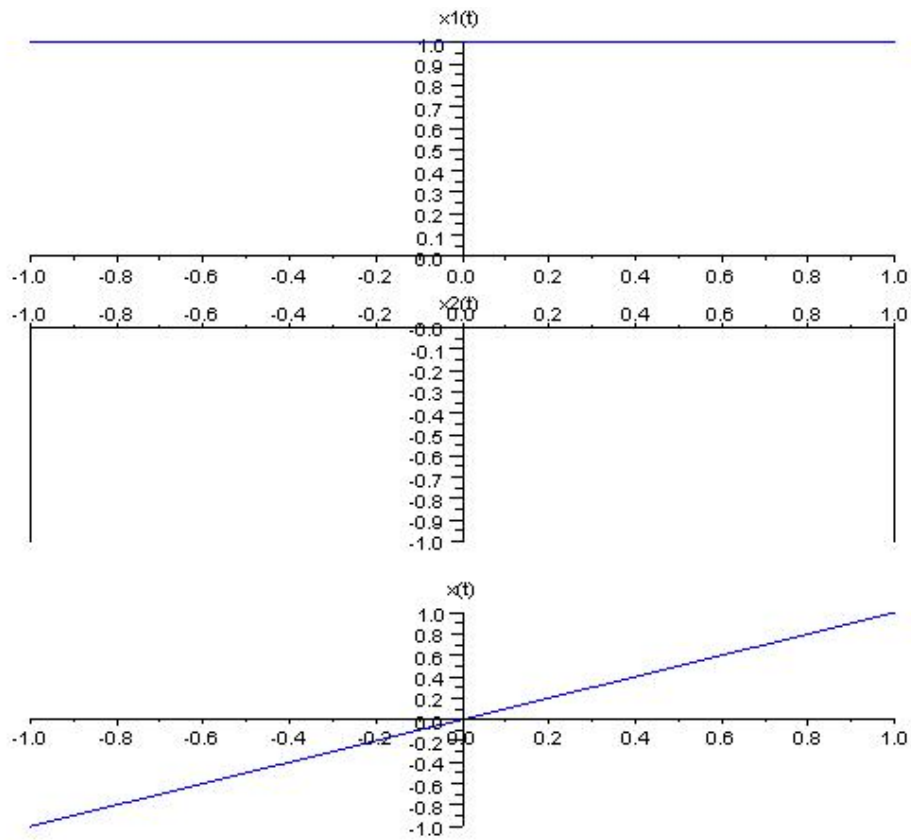


Figure 4.11: Results of Exa 4.12

**Example 4.18** Frequency Response of Ideal Low pass Filter  $X(jW) = 1$ , from  $-T_1$  to  $T_1$

```

1 //Example 4.18:Frequency Response of Ideal Low pass
  Filter
2 // X(jW)= 1, from -T1 to T1
3 clear;
4 clc;
5 close;
6 Wc = 10; //1 rad/sec
7 W = -Wc:0.1:Wc; //Passband of filter
8 HWO = 1; //Magnitude of Filter

```

```

9 HW = HW0*ones(1,length(W));
10 //Inverse Continuous-time Fourier Transform
11 t = -%pi:%pi/length(W):%pi;
12 Dw = 0.1;
13 ht =(1/(2*%pi))*HW *exp(sqrt(-1)*W'*t)*Dw;
14 ht = real(ht);
15 figure
16 subplot(2,1,1)
17 a = gca();
18 a.y_location = "origin";
19 a.x_location = "origin";
20 plot(W,HW);
21 xtitle('Frequency Response of Filter H(jW)')
22 subplot(2,1,2)
23 a = gca();
24 a.y_location = "origin";
25 a.x_location = "origin";
26 plot(t,ht);
27 xtitle('Impulse Response of Filter h(t)')

```

---



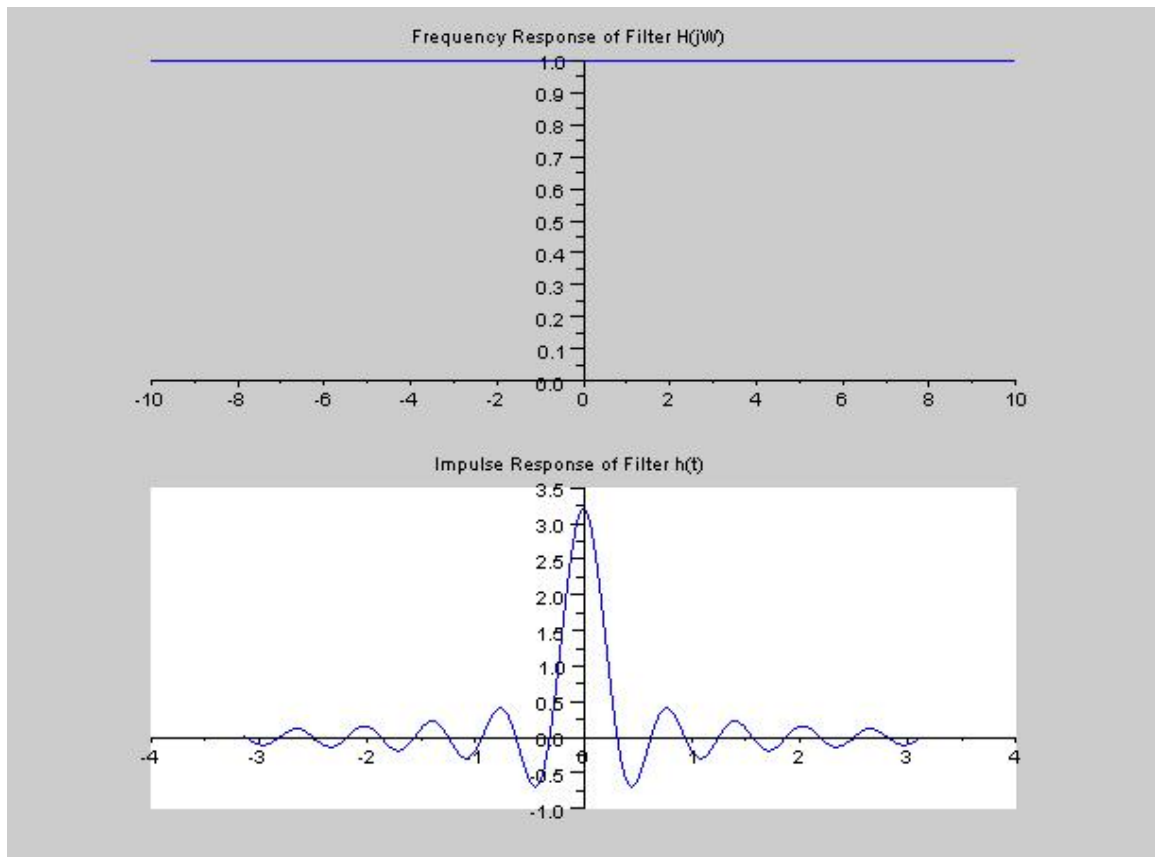


Figure 4.12: Results of Exa 4.18

**Example 4.23** Multiplication Property of CTFT

```

1 //Figure 4.23: Multiplication Property of CTFT
2 clear;
3 clc;
4 close;
5 W1 = -1:0.1:1;
6 W2 = -2:0.1:2;
7 W = -3:0.1:3;
8 //Fourier Transform of sinc function is square wave
9 XW1 = (1/%pi)*ones(1,length(W1)); //CTFT of x1(t)
10 XW2 = (1/(2*%pi))*ones(1,length(W2)); //CTFT of x2(t)

```

```

11 XW = (1/2)*convol(XW1,XW2); //CTFT of  $x(t)=x_1(t)*x_2(t)$ 
12 //X(jw) = linear convolution of X1(jw) and X2(jw)
13 figure
14 a = gca();
15 a.y_location = "origin";
16 a.x_location = "origin";
17 plot(W,XW);
18 xlabel('Frequency in Radians/Seconds --> W');
19 title('Multiplication Property X(jW)')

```

---

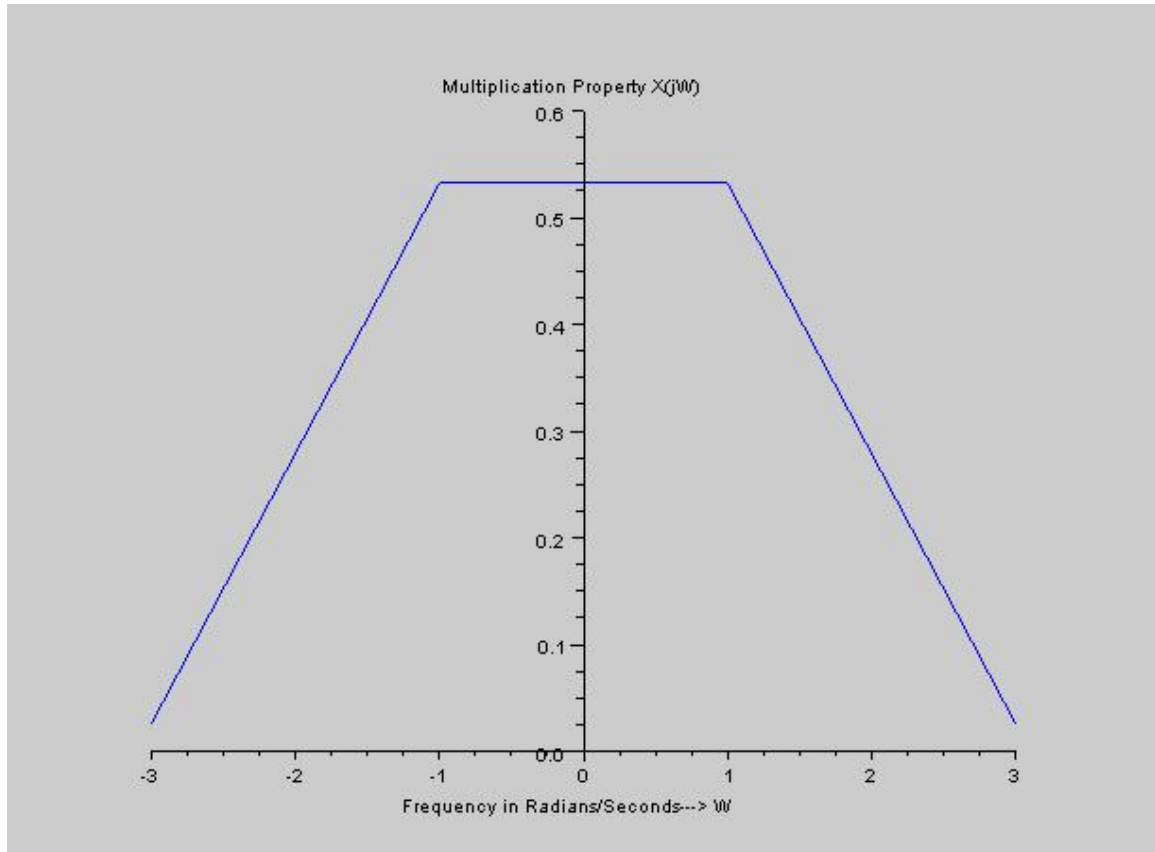


Figure 4.13: Results of Exa 4.23

```

Example 4.22 //Figure 4.22
2 //Plotting Continuous Time Fourier Transform of
3 //Impulse Response  $h(t) = \exp(-A*t)u(t)$ ,  $t > 0$ 
4 clear;
5 clc;
6 close;
7 // Analog Signal
8 A =1; //Amplitude
9 Dt = 0.005;
10 t = 0:Dt:10;
11 ht = exp(-A*t);
12 // Continuous-time Fourier Transform
13 Wmax = 2*pi*1; //Analog Frequency = 1Hz
14 K = 4;
15 k = 0:(K/1000):K;
16 W = k*Wmax/K;
17 HW = ht* exp(-sqrt(-1)*t'*W) * Dt;
18 HW_Mag = abs(HW);
19 W = [-mtlbfliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
20 HW_Mag = [mtlbfliplr(HW_Mag), HW_Mag(2:1001)];
21 //Plotting Continuous Time Signal
22 figure
23 a = gca();
24 a.y_location = "origin";
25 plot(t,ht);
26 xlabel('t in sec. ');
27 title('Impulse Response h(t)')
28 figure
29 //Plotting Magnitude Response of CTS
30 a = gca();
31 a.y_location = "origin";
32 plot(W,HW_Mag);
33 xlabel('Frequency in Radians/Seconds —> W');
34 title('Frequency Response H(jW)')

```

---

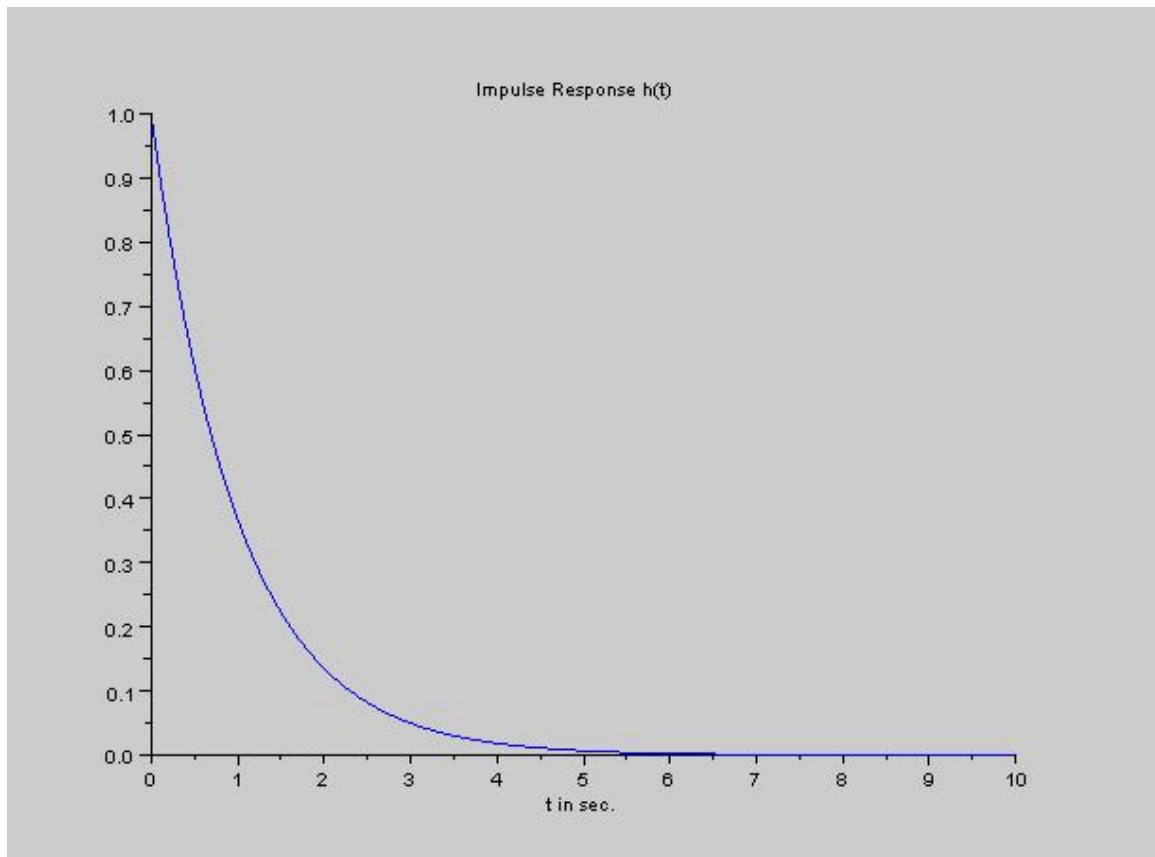


Figure 4.14: Results of Exa [4.22](#)

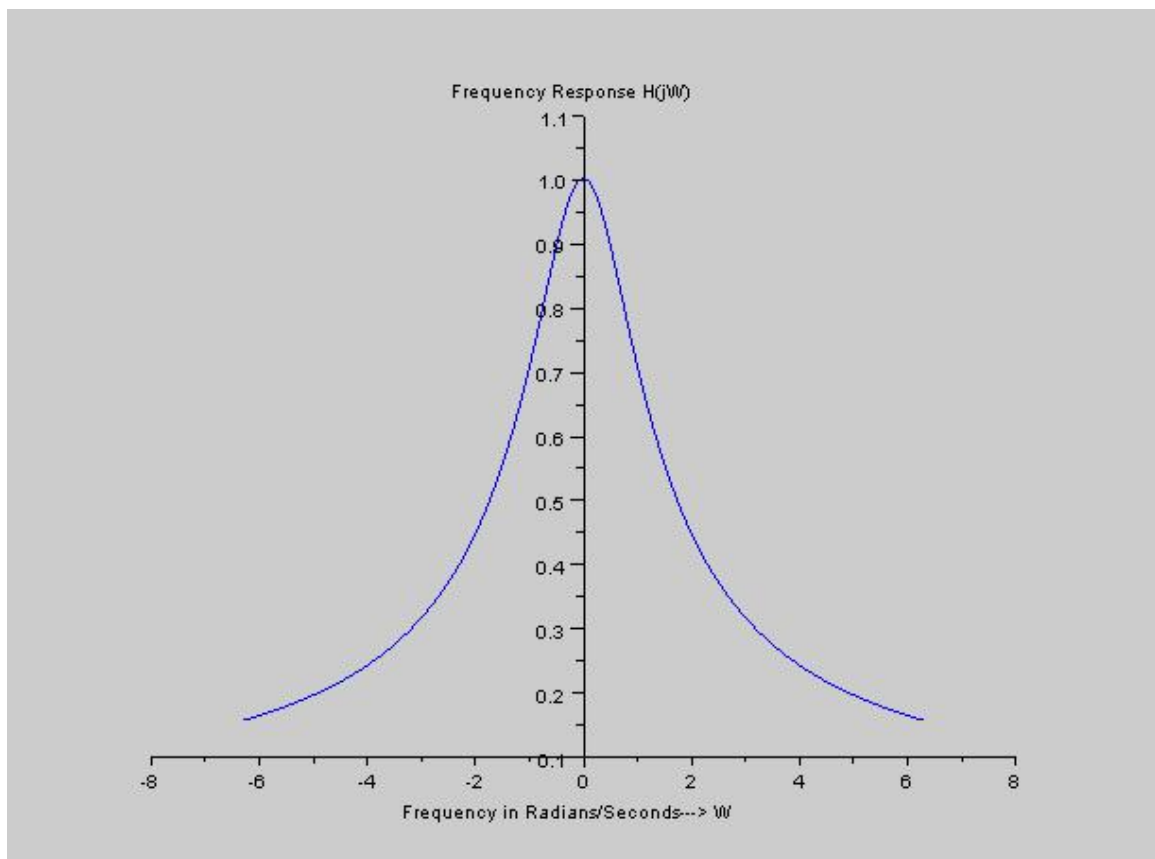


Figure 4.15: Results of Exa [4.22](#)

# Chapter 5

## The Discrete Time Fourier Transform

### 5.1 Scilab Codes

**Example 5.1** Discrete Time Fourier Transform of discrete sequence  $x[n] = (a^n).u[n]$ ,  $a > 0$  and  $a < 0$

```
1 //Example 5.1: Discrete Time Fourier Transform of
   discrete sequence
2 //x[n]= (a^n).u[n], a>0 and a<0
3 clear;
4 clc;
5 close;
6 // DTS Signal
7 a1 = 0.5;
8 a2 = -0.5;
9 max_limit = 10;
10 for n = 0:max_limit-1
11     x1(n+1) = (a1^n);
12     x2(n+1) = (a2^n);
13 end
14 n = 0:max_limit-1;
15 // Discrete-time Fourier Transform
16 Wmax = 2*%pi;
```

```

17 K = 4;
18 k = 0:(K/1000):K;
19 W = k*Wmax/K;
20 x1 = x1';
21 x2 = x2';
22 XW1 = x1* exp(-sqrt(-1)*n'*W);
23 XW2 = x2* exp(-sqrt(-1)*n'*W);
24 XW1_Mag = abs(XW1);
25 XW2_Mag = abs(XW2);
26 W = [-mtlbfliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
27 XW1_Mag = [mtlbfliplr(XW1_Mag), XW1_Mag(2:1001)];
28 XW2_Mag = [mtlbfliplr(XW2_Mag), XW2_Mag(2:1001)];
29 [XW1_Phase,db] = phasemag(XW1);
30 [XW2_Phase,db] = phasemag(XW2);
31 XW1_Phase = [-mtlbfliplr(XW1_Phase),XW1_Phase
    (2:1001)];
32 XW2_Phase = [-mtlbfliplr(XW2_Phase),XW2_Phase
    (2:1001)];
33 //plot for a>0
34 figure
35 subplot(3,1,1);
36 plot2d3('gnn',n,x1);
37 xtitle('Discrete Time Sequence x[n] for a>0')
38 subplot(3,1,2);
39 a = gca();
40 a.y_location = "origin";
41 a.x_location = "origin";
42 plot2d(W,XW1_Mag);
43 title('Magnitude Response abs(X(jW))')
44 subplot(3,1,3);
45 a = gca();
46 a.y_location = "origin";
47 a.x_location = "origin";
48 plot2d(W,XW1_Phase);
49 title('Phase Response <(X(jW))')
50 //plot for a<0
51 figure

```

```

52 subplot(3,1,1);
53 plot2d3('gnn',n,x2);
54 xtitle('Discrete Time Sequence x[n] for a>0')
55 subplot(3,1,2);
56 a = gca();
57 a.y_location = "origin";
58 a.x_location = "origin";
59 plot2d(W,XW2_Mag);
60 title('Magnitude Response abs(X(jW))')
61 subplot(3,1,3);
62 a = gca();
63 a.y_location = "origin";
64 a.x_location = "origin";
65 plot2d(W,XW2_Phase);
66 title('Phase Response <(X(jW))')

```

---



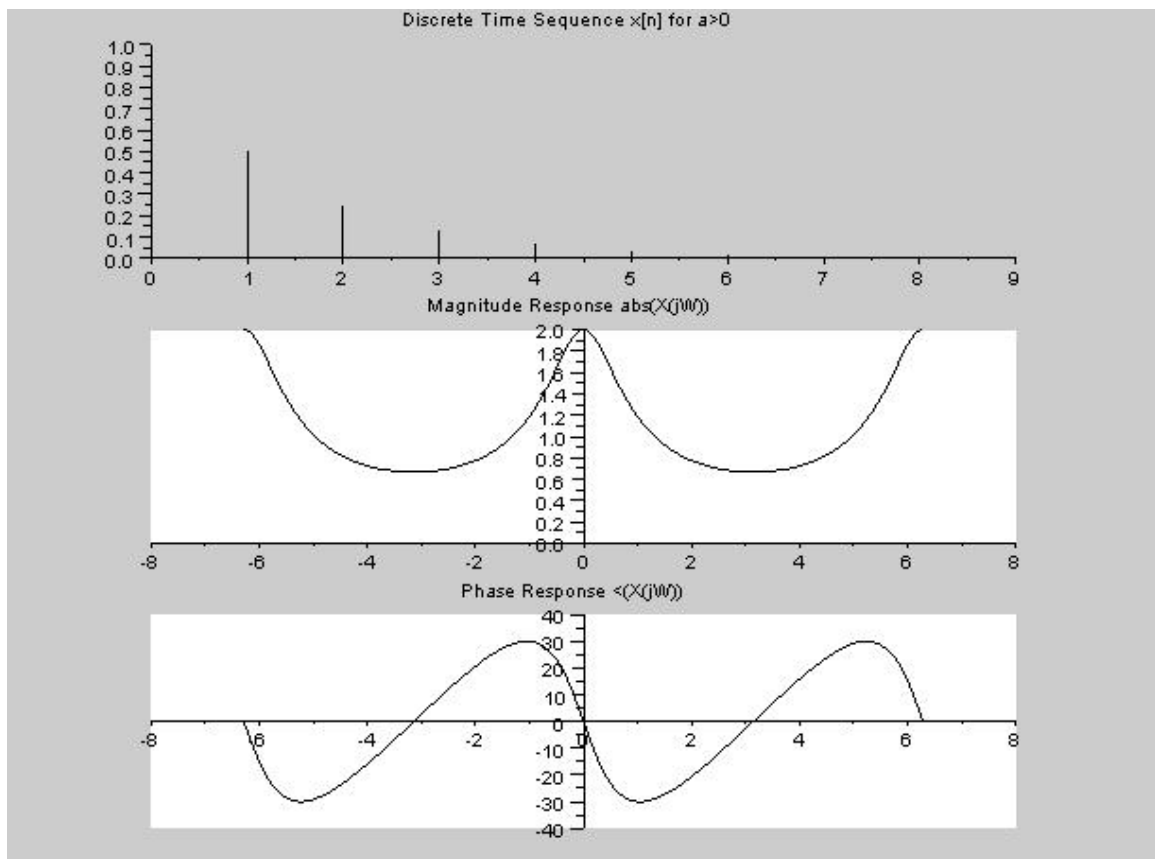


Figure 5.1: Results of Exa 5.1

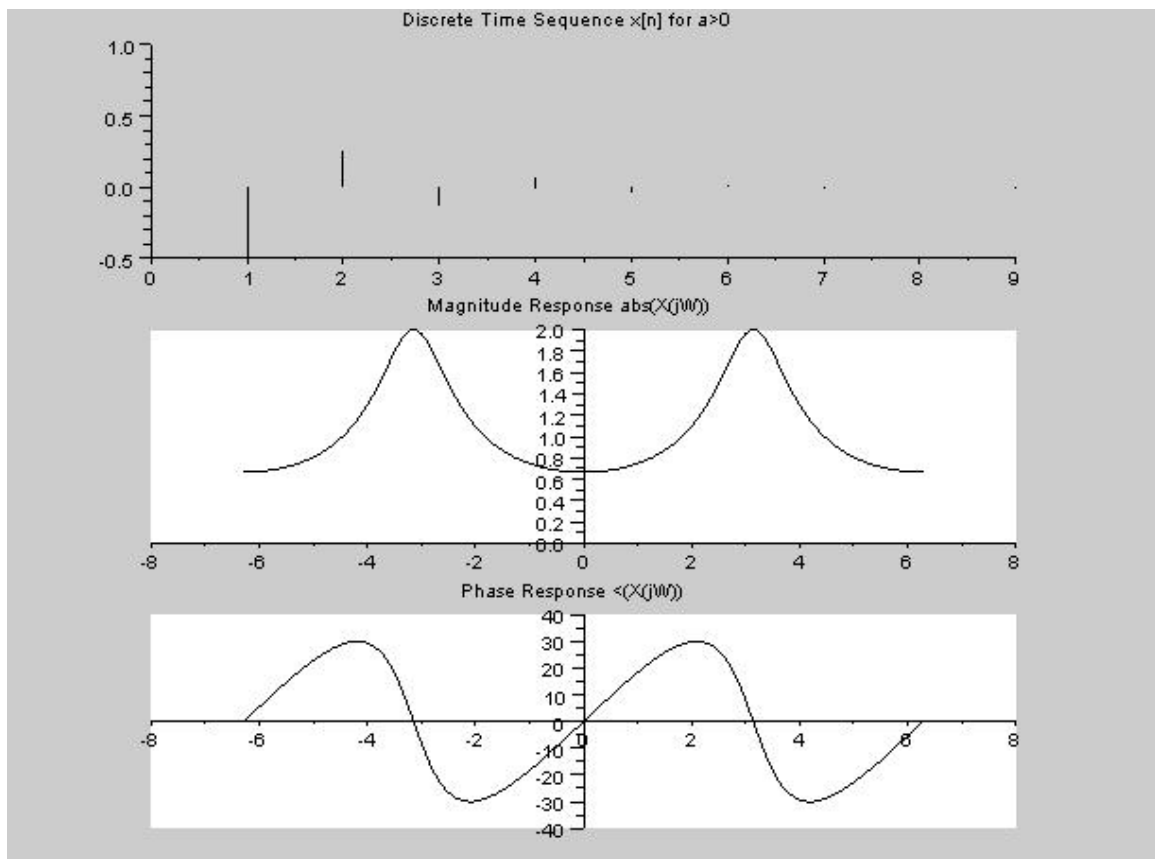


Figure 5.2: Results of Exa 5.1

**Example 5.2** Discrete Time Fourier Transform of  $x[n] = (a^{|n|} \cos(n))$   $a > 0$  and  $a < 0$

```

1 //Example 5.2: Discrete Time Fourier Transform of
2 //x[n]= (a^abs(n)) a>0 and a<0
3 clear;
4 clc;
5 close;
6 // DTS Signal
7 a = 0.5;
8 max_limit = 10;
9 n = -max_limit+1:max_limit-1;

```

```

10 x = a^abs(n);
11 // Discrete-time Fourier Transform
12 Wmax = 2*%pi;
13 K = 4;
14 k = 0:(K/1000):K;
15 W = k*Wmax/K;
16 XW = x* exp(-sqrt(-1)*n'*W);
17 XW_Mag = real(XW);
18 W = [-mtlbfliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
19 XW_Mag = [mtlbfliplr(XW_Mag), XW_Mag(2:1001)];
20 //plot for abs(a)<1
21 figure
22 subplot(2,1,1);
23 a = gca();
24 a.y_location = "origin";
25 a.x_location = "origin";
26 plot2d3('gnn',n,x);
27 xtitle('Discrete Time Sequence x[n] for a>0')
28 subplot(2,1,2);
29 a = gca();
30 a.y_location = "origin";
31 a.x_location = "origin";
32 plot2d(W,XW_Mag);
33 title('Discrete Time Fourier Transform X(exp(jW))')

```

---

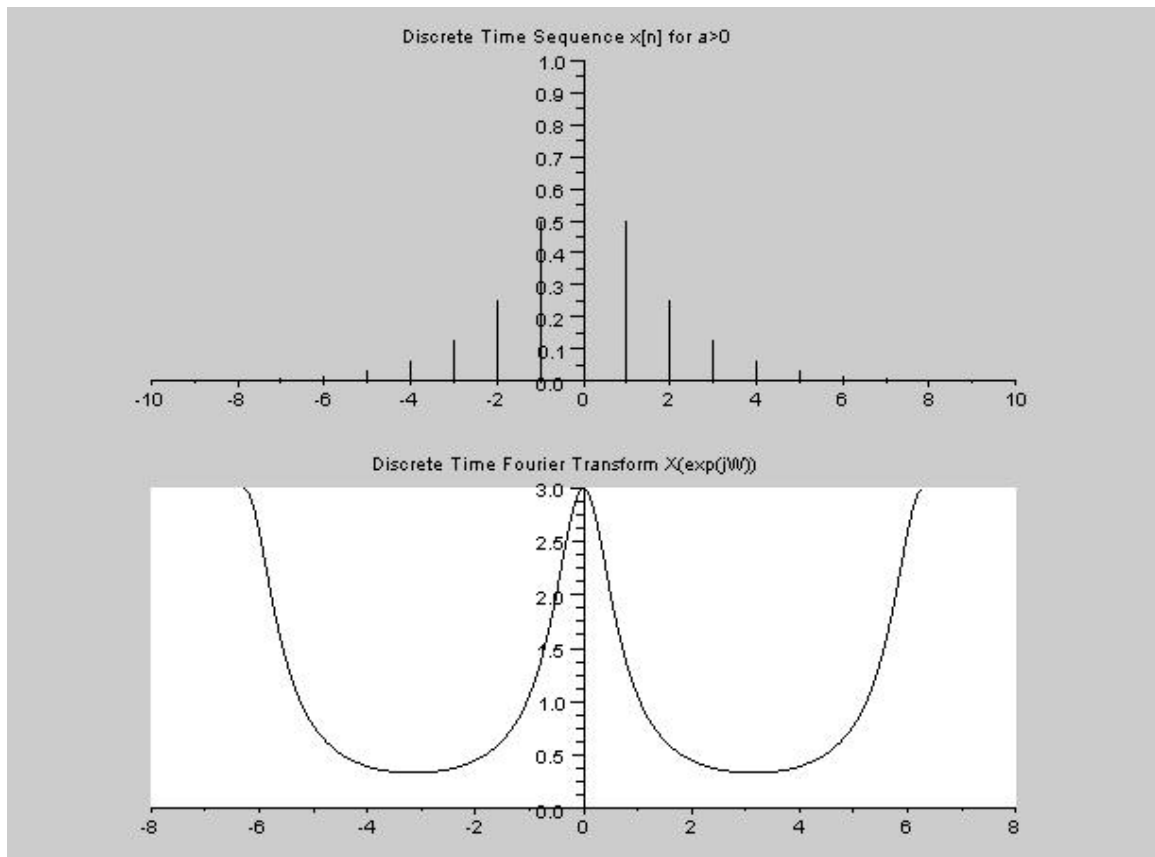


Figure 5.3: Results of Exa 5.2

**Example 5.3** Discrete Time Fourier Transform of  $x[n] = 1, \text{abs}(n) \leq N1$

```

1 //Example 5.3: Discrete Time Fourier Transform of
2 //x[n]= 1 , abs(n)<=N1
3 clear;
4 clc;
5 close;
6 // DTS Signal
7 N1 = 2;
8 n = -N1:N1;
9 x = ones(1,length(n));
10 // Discrete-time Fourier Transform

```

```

11 Wmax = 2*%pi;
12 K = 4;
13 k = 0:(K/1000):K;
14 W = k*Wmax/K;
15 XW = x* exp(-sqrt(-1)*n'*W);
16 XW_Mag = real(XW);
17 W = [-mtlbfliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
18 XW_Mag = [mtlbfliplr(XW_Mag), XW_Mag(2:1001)];
19 //plot for abs(a)<1
20 figure
21 subplot(2,1,1);
22 a = gca();
23 a.y_location = "origin";
24 a.x_location = "origin";
25 plot2d3('gnn',n,x);
26 xtitle('Discrete Time Sequence x[n]')
27 subplot(2,1,2);
28 a = gca();
29 a.y_location = "origin";
30 a.x_location = "origin";
31 plot2d(W,XW_Mag);
32 title('Discrete Time Fourier Transform X(exp(jW))')

```

---

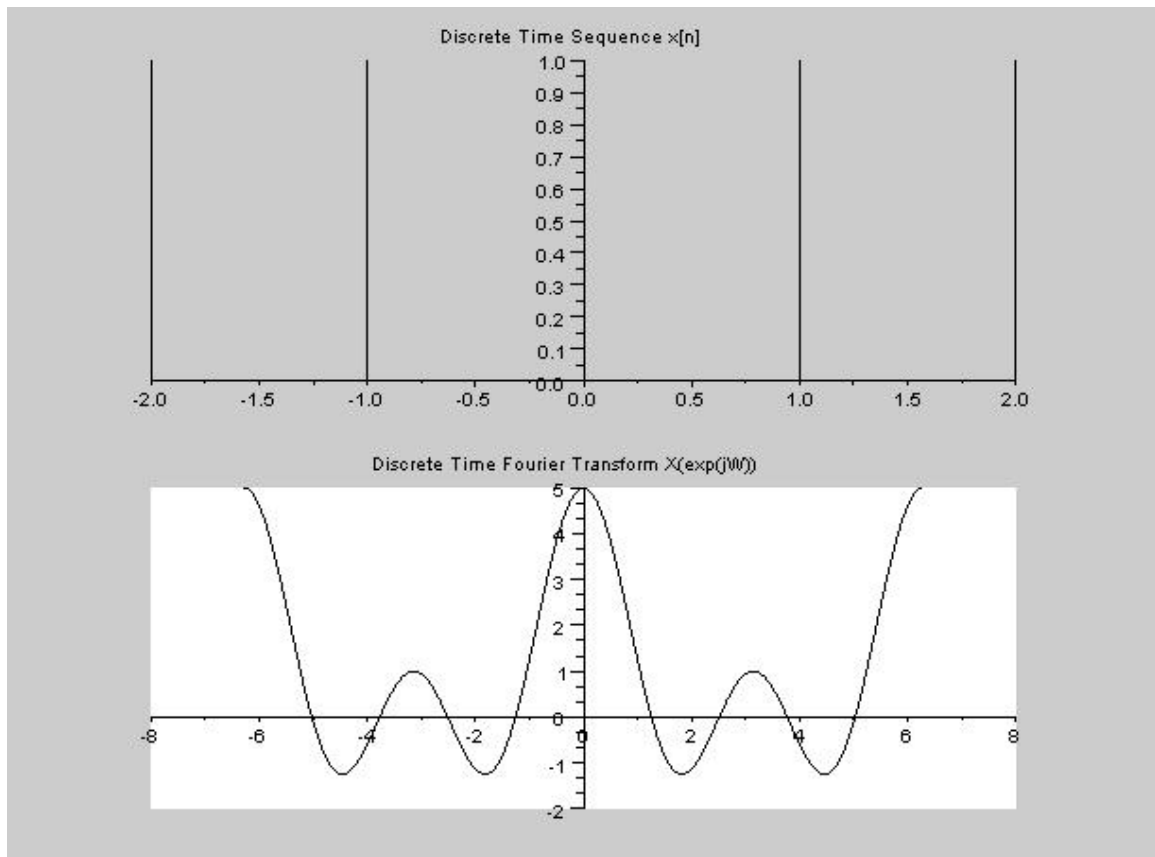


Figure 5.4: Results of Exa 5.3

**Example 5.5** Discrete Time Fourier Transform:  $x[n] = \cos(nW_o)$

```

1 //Example5.5: Discrete Time Fourier Transform: x[n]=
  cos(nWo)
2 clear;
3 clc;
4 close;
5 N = 5;
6 Wo = 2*%pi/N;
7 W = [-Wo,0,Wo];
8 XW =[%pi,0,%pi];
9 //

```

```

10 figure
11 a = gca();
12 a.y_location = "origin";
13 a.x_location = "origin";
14 plot2d3('gnn',W,XW,2);
15 poly1 = a.children(1).children(1);
16 poly1.thickness = 3;
17 xlabel('

        W');
18 title('DTFT of cos(nWo)')
19 disp(Wo)

```

---

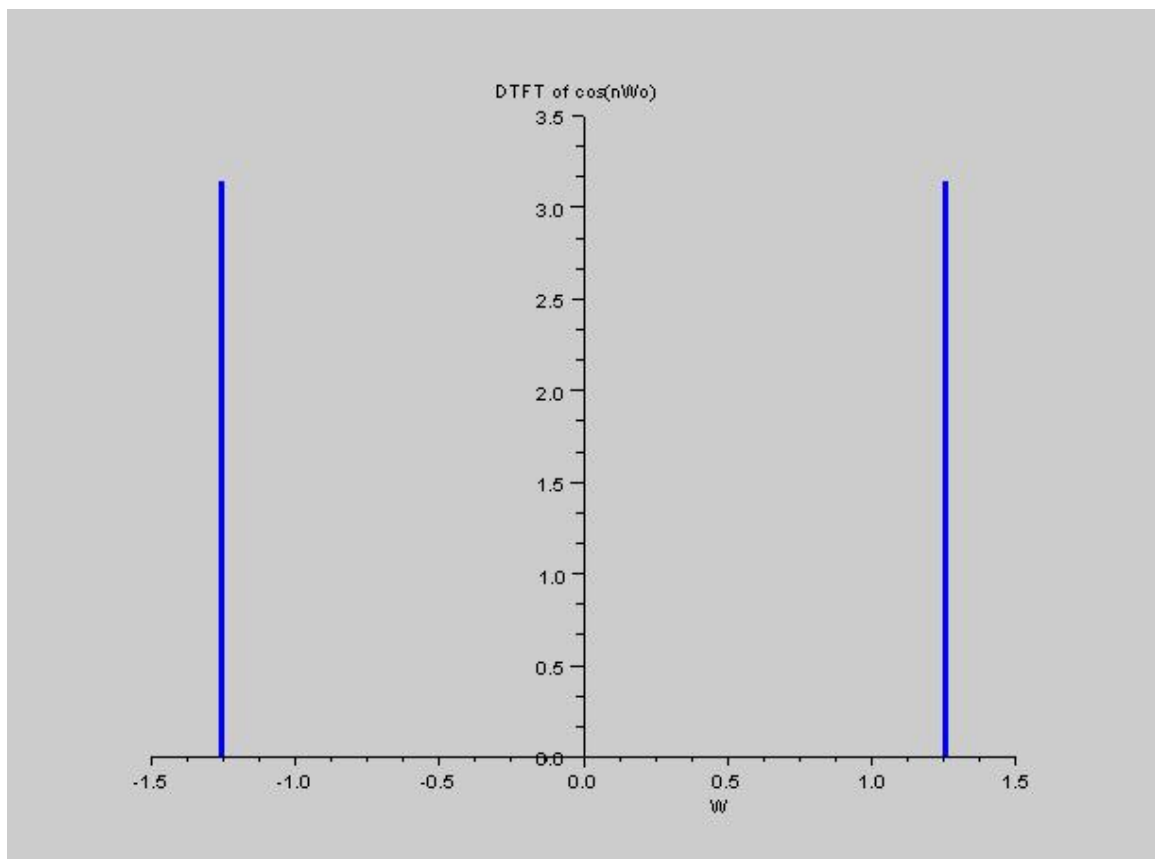


Figure 5.5: Results of Exa 5.5

**Example 5.6** Discrete Time Fourier Transform of Periodic Impulse Train

```

1 //Example5.6:Discrete Time Fourier Transform of
2 // Periodic Impulse Train
3 clear;
4 clc;
5 close;
6 N = 5;
7 N1 = -3*N:3*N;
8 xn = [zeros(1,N-1),1];
9 x = [1 xn xn xn xn xn xn];
10 ak = 1/N;
```



```

11 XW = 2*%pi*ak*ones(1,2*N);
12 Wo = 2*%pi/N;
13 n   = -N:N-1;
14 W = Wo*n;
15 figure
16 subplot(2,1,1)
17 a = gca();
18 a.y_location = "origin";
19 a.x_location = "origin";
20 plot2d3('gnn',N1,x,2);
21 poly1 = a.children(1).children(1);
22 poly1.thickness = 3;
23 xlabel('

        n');
24 title('Periodic Impulse Train')
25 subplot(2,1,2)
26 a = gca();
27 a.y_location = "origin";
28 a.x_location = "origin";
29 plot2d3('gnn',W,XW,2);
30 poly1 = a.children(1).children(1);
31 poly1.thickness = 3;
32 xlabel('

        W');
33 title('DTFT of Periodic Impulse Train')
34 disp(Wo)

```

---

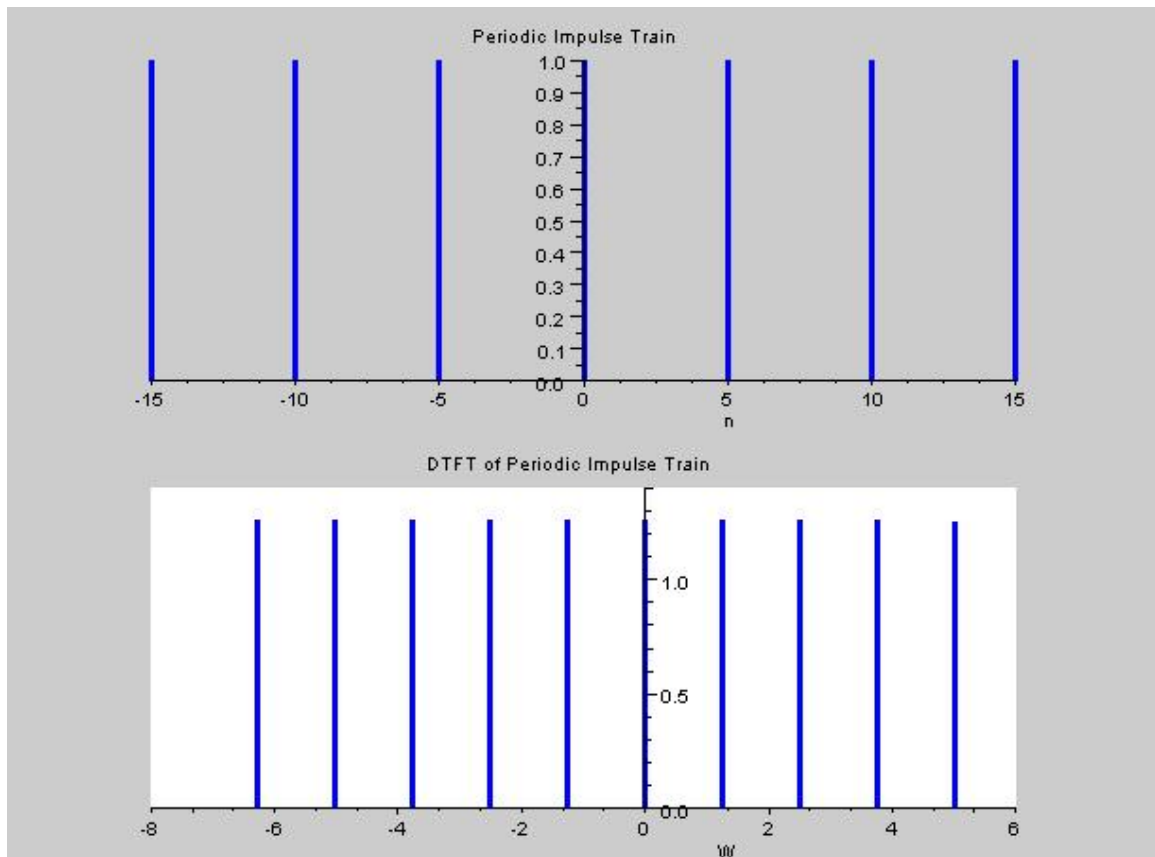


Figure 5.6: Results of Exa 5.6

**Example 5.7** Frequency Shifting Property of DTFT: Frequency Response of Ideal Low pass Filter and HPF

```

1 //Example 5.7: Frequency Shifting Property of DTFT:
  Frequency Response of Ideal Low pass Filter and
  HPF
2 clear;
3 clc;
4 close;
5 Wc = 1; //1 rad/sec
6 W = -Wc:0.1:Wc; //Passband of filter
7 H0 = 1; //Magnitude of Filter

```

```

8 HlpW = H0*ones(1,length(W));
9 Whp1 = W+%pi;
10 Whp2 = -W-%pi;
11 figure
12 subplot(2,1,1)
13 a = gca();
14 a.y_location = "origin";
15 a.x_location = "origin";
16 a.data_bounds=[-%pi,0;%pi,2];
17 plot2d(W,HlpW);
18 xtitle('Frequency Response of LPF  $H(\exp(jW))$  ')
19 subplot(2,1,2)
20 a = gca();
21 a.y_location = "origin";
22 a.x_location = "origin";
23 a.data_bounds=[-2*%pi,0;2*%pi,2];
24 plot2d(Whp1,HlpW);
25 plot2d(Whp2,HlpW);
26 xtitle('Frequency Response of HPF  $H(\exp(jW))$  ')

```

---

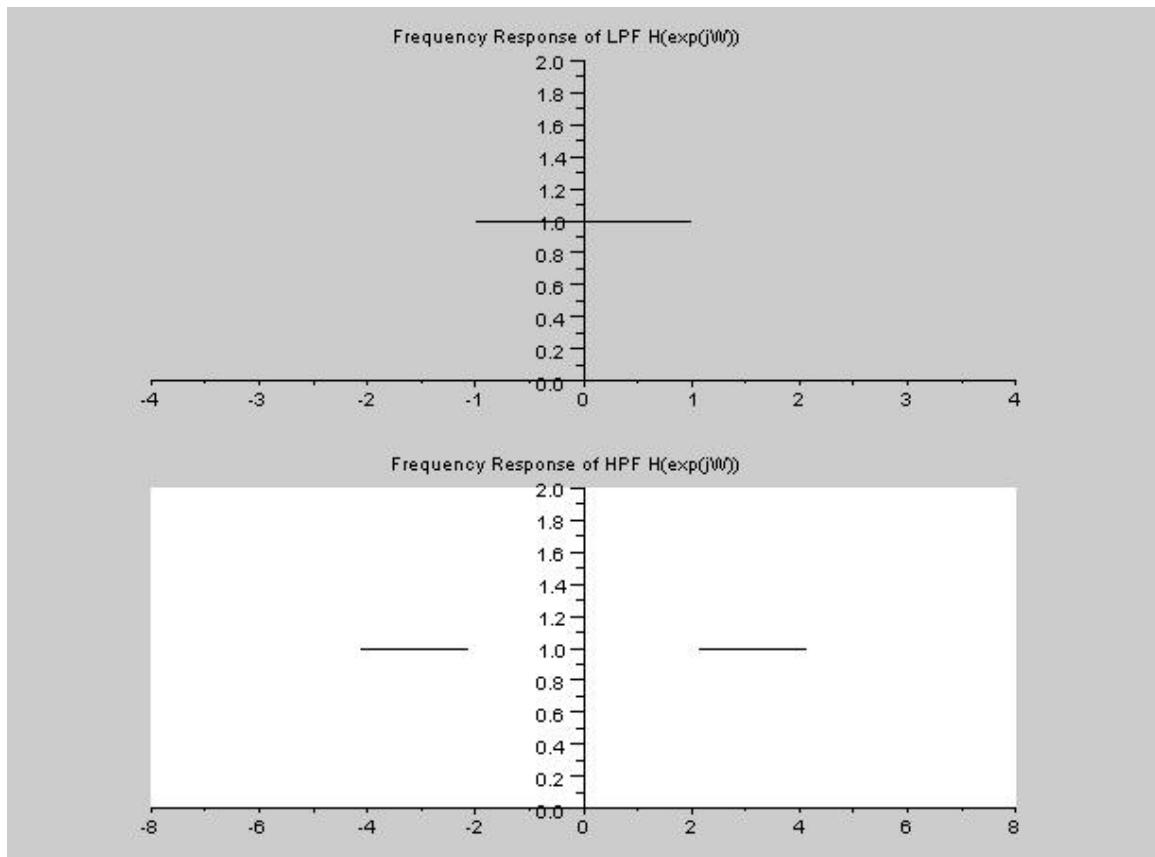


Figure 5.7: Results of Exa 5.7

### Example 5.9 Time Expansion Property of DTFT

```

1 //Example 5.9:Time Expansion Property of DTFT
2 clear;
3 close;
4 clc;
5 n = -1:11;
6 x = [0,1,2,1,2,1,2,1,2,1,2,0,0];
7 y = [1,1,1,1,1];
8 y_2_n = zeros(1,2*length(y)+1);
9 y_2_n(1:2:2*length(y)) = y;
10 y_2_n = [0 y_2_n 0];

```

```

11 y_2_n_1 = [0,y_2_n(1:$-1)];
12 x_r = y_2_n+2*y_2_n_1;
13 y = [0,y,zeros(1,7)];
14 figure
15 subplot(4,1,1)
16 plot2d3('gnn',n,y)
17 title('y[n]')
18 subplot(4,1,2)
19 plot2d3('gnn',n,y_2_n)
20 title('y(2)[n]')
21 subplot(4,1,3)
22 plot2d3('gnn',n,y_2_n_1)
23 title('y(2)[n-1]')
24 subplot(4,1,4)
25 plot2d3('gnn',n,x)
26 title('x[n]=y(2)[n]+2*y(2)[n-1]')

```

---

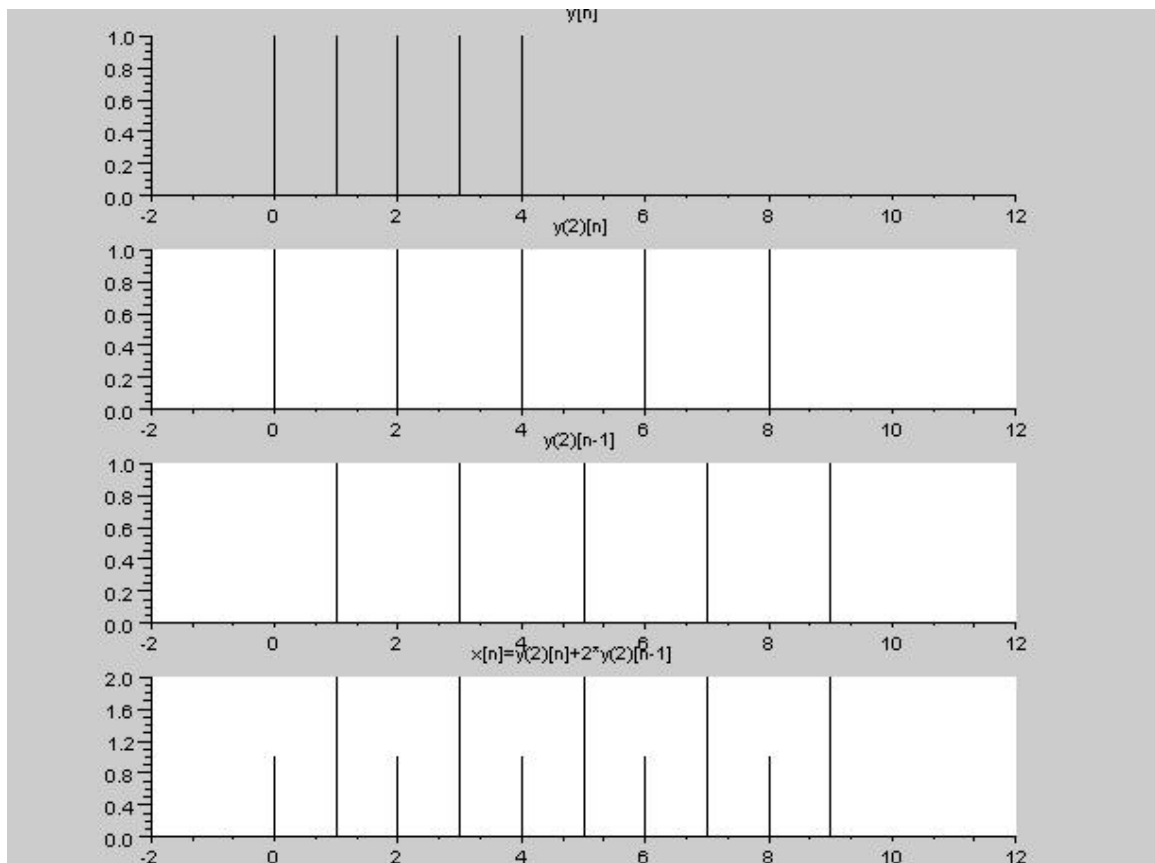


Figure 5.8: Results of Exa 5.9

**Example 5.12** IDTFT: Impulse Response of Ideal Low pass Filter

```

1 //Example 5.12:IDTFT:Impulse Response of Ideal Low
  pass Filter
2 clear;
3 clc;
4 close;
5 Wc = 1; //1 rad/sec
6 W = -Wc:0.1:Wc; //Passband of filter
7 H0 = 1; //Magnitude of Filter
8 HlpW = H0*ones(1,length(W));
9 //Inverse Discrete-time Fourier Transform

```

```

10 t = -2*%pi:2*%pi/length(W):2*%pi;
11 ht =(1/(2*%pi))*HlpW *exp(sqrt(-1)*W'*t);
12 ht = real(ht);
13 figure
14 subplot(2,1,1)
15 a = gca();
16 a.y_location = "origin";
17 a.x_location = "origin";
18 a.data_bounds=[-%pi,0;%pi,2];
19 plot2d(W,HlpW,2);
20 poly1 = a.children(1).children(1);
21 poly1.thickness = 3;
22 xtitle('Frequency Response of LPF H(exp(jW))')
23 subplot(2,1,2)
24 a = gca();
25 a.y_location = "origin";
26 a.x_location = "origin";
27 a.data_bounds=[-2*%pi,-1;2*%pi,2];
28 plot2d3('gnn',t,ht);
29 poly1 = a.children(1).children(1);
30 poly1.thickness = 3;
31 xtitle('Impulse Response of LPF h(t)')

```

---

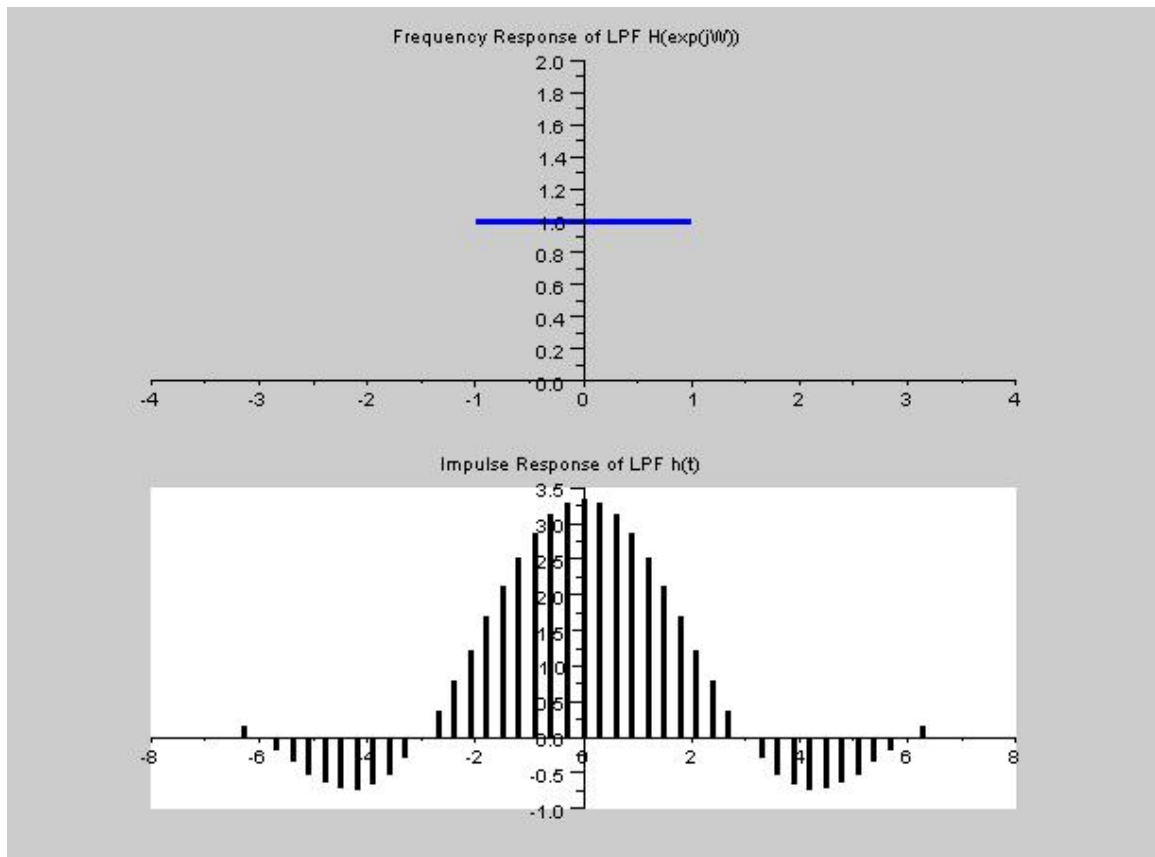


Figure 5.9: Results of Exa 5.12

**Example 5.15** Multiplication Property of DTFT

```

1 //Example5.15: Multiplication Property of DTFT
2 clear;
3 clc;
4 close;
5 n = 1:100;
6 x2 = [3/4, sin(0.75*pi*n)./(%pi*n)];
7 x1 = [1/2, sin(0.5*pi*n)./(%pi*n)];
8 x = x1.*x2;
9 Wmax = %pi;
10 K = 1;

```



```

11 k = 0:(K/1000):K;
12 W = k*Wmax/K;
13 n = 0:100;
14 XW1 = x1* exp(-sqrt(-1)*n'*W);
15 XW2 = x2* exp(-sqrt(-1)*n'*W);
16 XW = x* exp(-sqrt(-1)*n'*W);
17 XW1_Mag = real(XW1);
18 XW2_Mag = real(XW2);
19 XW_Mag = real(XW);
20 W = [-mtlbfliplr(W), W(2:$)]; // Omega from -Wmax
    to Wmax
21 XW1_Mag = [mtlbfliplr(XW1_Mag), XW1_Mag(2:$)];
22 XW2_Mag = [mtlbfliplr(XW2_Mag), XW2_Mag(2:$)];
23 XW_Mag = [mtlbfliplr(XW_Mag), XW_Mag(2:$)];
24 figure
25 subplot(3,1,1)
26 a = gca();
27 a.y_location = "origin";
28 a.x_location = "origin";
29 plot(W,XW1_Mag);
30 title('DTFT X1(exp(jW))');
31 subplot(3,1,2)
32 a = gca();
33 a.y_location = "origin";
34 a.x_location = "origin";
35 plot(W,XW2_Mag);
36 title('DTFT X2(exp(jW))');
37 subplot(3,1,3)
38 a = gca();
39 a.y_location = "origin";
40 a.x_location = "origin";
41 plot(W,XW_Mag);
42 title('Multiplication Property of DTFT');

```

---

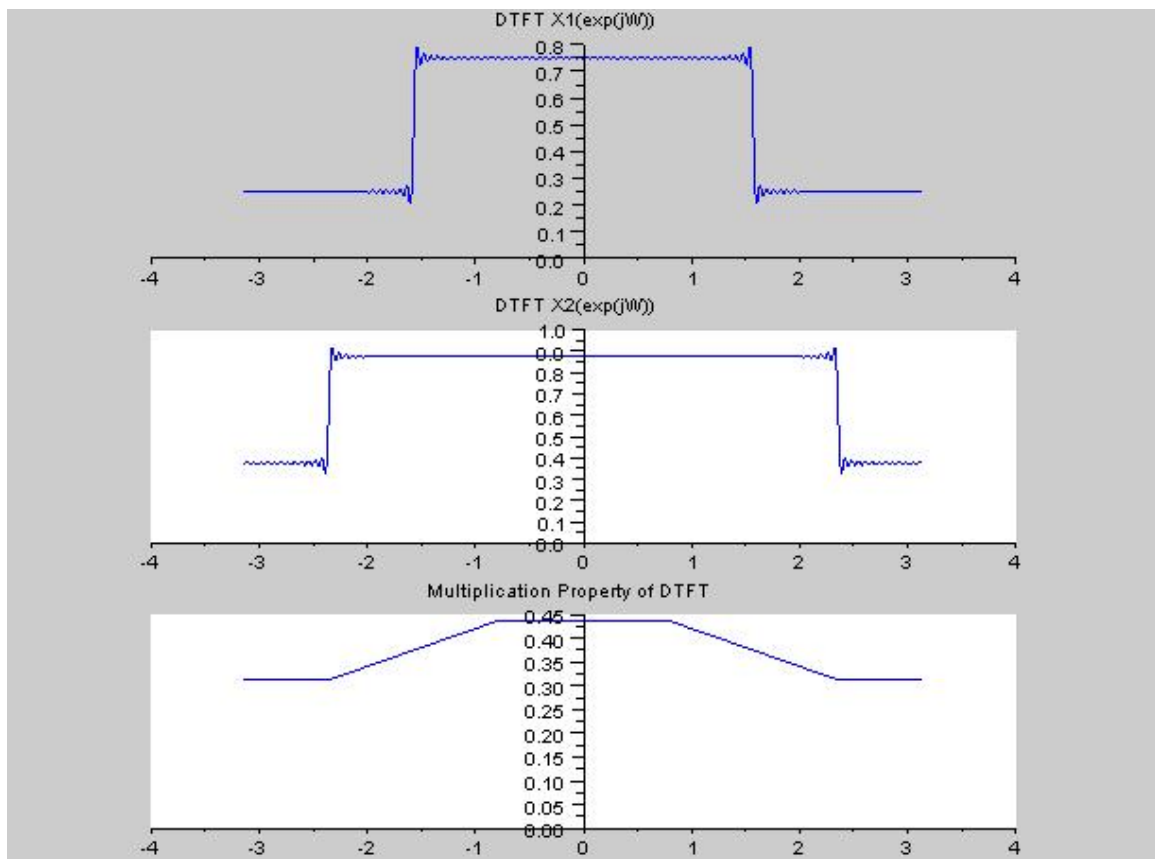


Figure 5.10: Results of Exa 5.15

# Chapter 6

## Time and Frequency Characterization of Signals and Systems

### 6.1 Scilab Codes

**Example 6.1** Phase Response and Group Delay

```
1 //Example6.1:Phase Response and Group Delay
2 clear;
3 clc;
4 close;
5 f1 = 50;
6 f2 = 150;
7 f3 = 300;
8 w1 = 315;
9 tuo1 = 0.066;
10 w2 = 943;
11 tuo2 = 0.033;
12 w3 = 1888;
13 tuo3 = 0.058;
14 f = 0:0.1:400;
15 W = 2*%pi*f;
16 for i =1:length(f)
```

```

17  num1(i) = (1+(sqrt(-1)*f(i)/f1)^2-2*sqrt(-1)*tuo1*(
    f(i)/f1));
18  den1(i) = (1+(sqrt(-1)*f(i)/f1)^2+2*sqrt(-1)*tuo1*(
    f(i)/f1));
19  H1W(i) = num1(i)/den1(i);
20  num2(i) = (1+(sqrt(-1)*f(i)/f2)^2-2*sqrt(-1)*tuo2*(
    f(i)/f2));
21  den2(i) = (1+(sqrt(-1)*f(i)/f2)^2+2*sqrt(-1)*tuo2*(
    f(i)/f2));
22  H2W(i) = num2(i)/den2(i);
23  num3(i) = (1+(sqrt(-1)*f(i)/f3)^2-2*sqrt(-1)*tuo3*(
    f(i)/f3));
24  den3(i) = (1+(sqrt(-1)*f(i)/f3)^2+2*sqrt(-1)*tuo3*(
    f(i)/f3));
25  H3W(i) = num3(i)/den3(i);
26  H_W(i) = H1W(i)*H2W(i);
27  HW(i) = H_W(i)*H3W(i);
28  phase1(i) = -2*atan((2*tuo1*(f(i)/f1))/(1.001-(f(i)
    )/f1)^2));
29  phase2(i) = -2*atan((2*tuo2*(f(i)/f2))/(1.001-(f(i)
    )/f2)^2));
30  phase3(i) = -2*atan((2*tuo3*(f(i)/f3))/(1.001-(f(i)
    )/f3)^2));
31  phase_total(i) = phase1(i)+phase2(i)+phase3(i);
32  if(f(i)<=50)
33      W_phase1(i) = -2*atan((2*tuo1*(f(i)/f1))
        /(1.001-(f(i)/f1)^2));
34      W_phase2(i) = -2*atan((2*tuo2*(f(i)/f2))
        /(1.001-(f(i)/f2)^2));
35      W_phase3(i) = -2*atan((2*tuo3*(f(i)/f3))
        /(1.001-(f(i)/f3)^2));
36      group_delay(i) = -phase_total(i)*0.1/%pi;    //
        delta_f= 0.1
37  elseif(f(i)>=50 & f(i)<=150)
38      W_phase1(i)= -2*%pi-2*atan((2*tuo1*(f(i)/f1))
        /(1.001-(f(i)/f1)^2));
39      W_phase2(i)= -2*atan((2*tuo2*(f(i)/f2))/(1.001-(
        f(i)/f2)^2));

```

```

40     W_phase3(i)= -2*atan((2*tuo3*(f(i)/f3))/(1.001-(
        f(i)/f3)^2));
41     group_delay(i) = -phase_total(i)*0.1/(2*pi);
42 elseif(f(i)>=150 & f(i)<=300)
43     W_phase1(i)= -2*atan((2*tuo1*(f(i)/f1))/(1.001-(
        f(i)/f1)^2));
44     W_phase2(i)= -4*pi-2*atan((2*tuo2*(f(i)/f2))
        /(1.001-(f(i)/f2)^2));
45     W_phase3(i)= -2*atan((2*tuo3*(f(i)/f3))/(1.001-(
        f(i)/f3)^2));
46     group_delay(i) = -phase_total(i)*0.1/(4*pi);
47 elseif(f(i)>300 & f(i)<=400)
48     W_phase1(i)= -2*atan((2*tuo1*(f(i)/f1))/(1.001-(
        f(i)/f1)^2));
49     W_phase2(i)= -2*atan((2*tuo2*(f(i)/f2))/(1.001-(
        f(i)/f2)^2));
50     W_phase3(i)= -6*pi-2*atan((2*tuo3*(f(i)/f3))
        /(1.001-(f(i)/f3)^2));
51     group_delay(i) = -phase_total(i)*0.1/(4*pi);
52 end
53 if(f(i)==300.1)
54     W_phase_total(i) = 2*pi+W_phase1(i)+W_phase2(i)+
        W_phase3(i);
55 else
56     W_phase_total(i) = W_phase1(i)+W_phase2(i)+
        W_phase3(i);
57 end
58 end
59 figure
60 plot2d(f,phase_total,2)
61 xtitle('Principal phase','Frequency(Hz)','Phase(rad)
    ');
62 figure
63 plot2d(f,W_phase_total,2)
64 xtitle('unwrapped phase','Frequency(Hz)','Phase(rad)
    ');
65 figure
66 plot2d(f,abs(group_delay),2)

```

```
67 xtitle('group delay','Frequency(Hz)','Group Delay(  
    sec)');
```

---

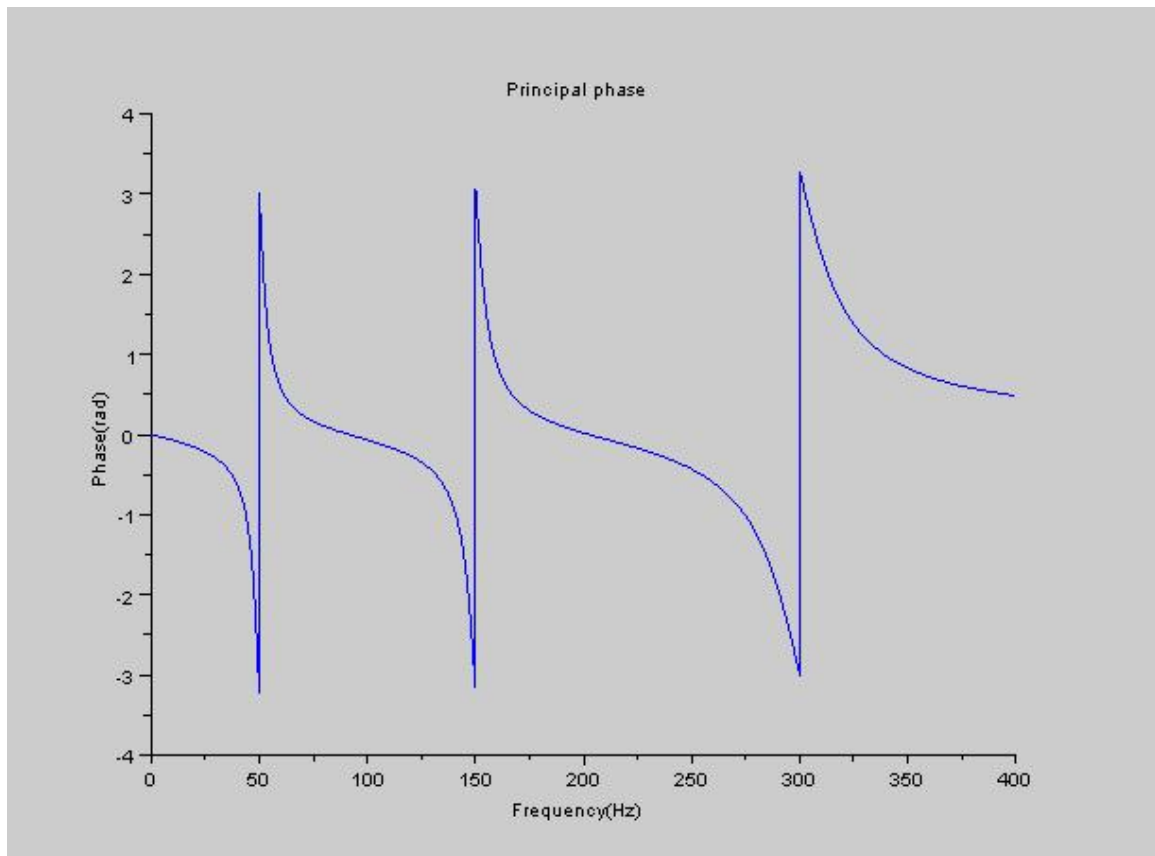


Figure 6.1: Results of Exa [6.1](#)

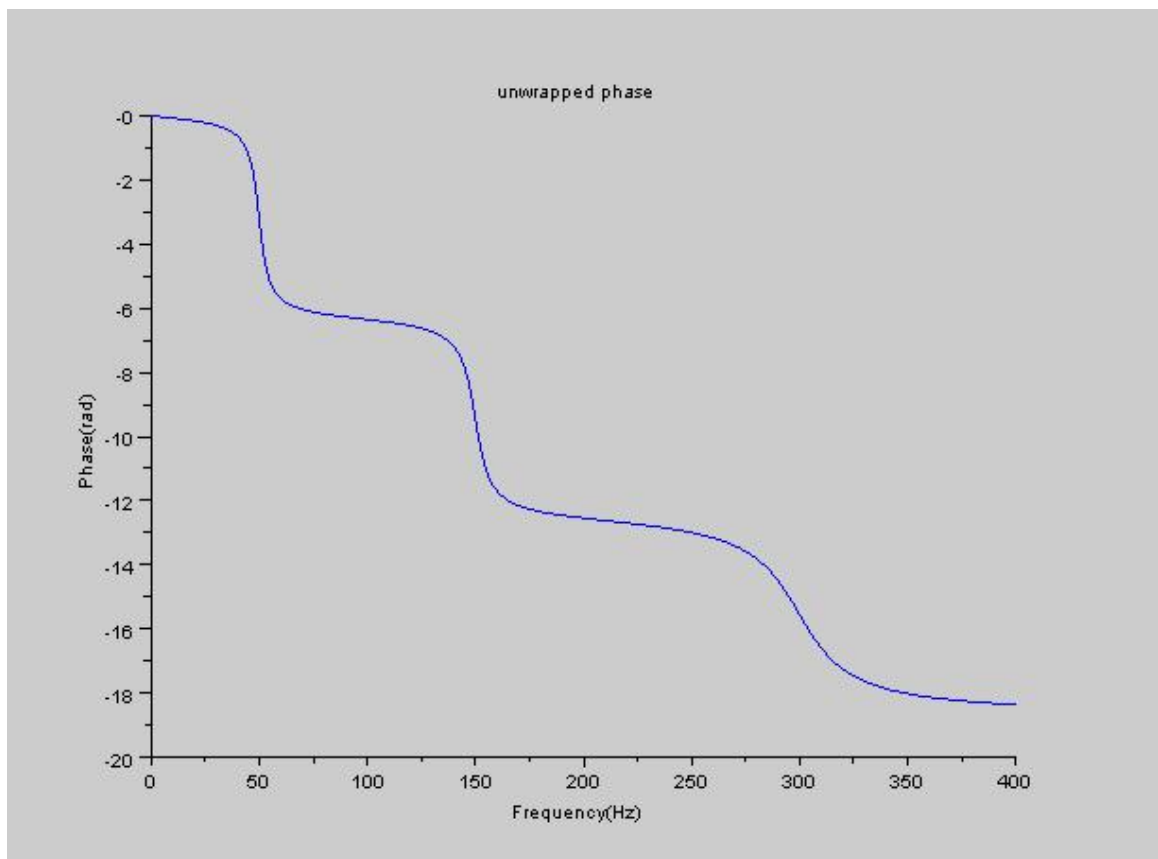


Figure 6.2: Results of Exa [6.1](#)

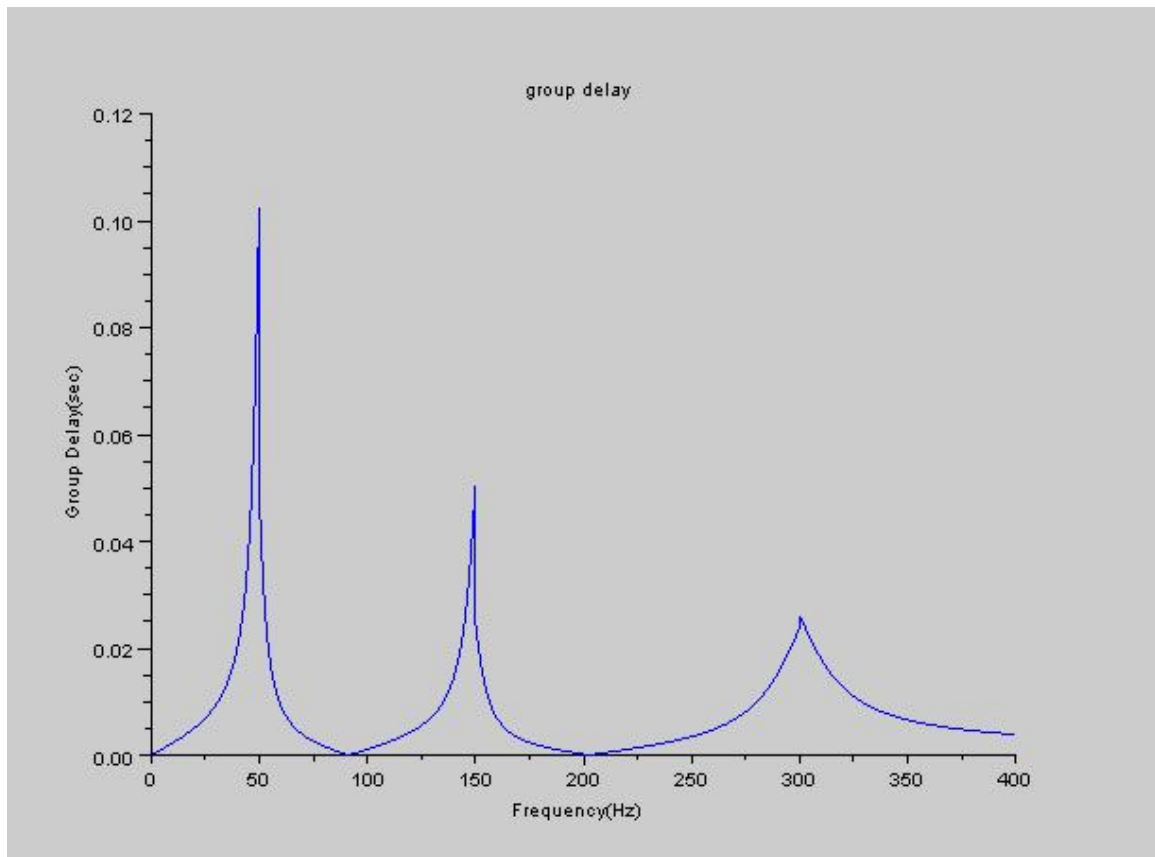


Figure 6.3: Results of Exa 6.1

**Example 6.3** Analog Lowpass IIR filter design Cutoff frequency  $F_c = 500\text{Hz}$   
 Passband ripple 1-0.05 and stopband ripple = 0.05

```

1 //Example6.3:Analog Lowpass IIR filter design
2 //Cutoff frequency  $F_c = 500\text{Hz}$ 
3 //Passband ripple 1-0.05 and stopband ripple = 0.05
4 clear;
5 close;
6 clc;
7 hs_butt = analpf(5,'butt',[0.05,0.05],500);
8 hs_ellip = analpf(5,'ellip',[0.05,0.05],500);
9 fr=0:.1:2000;
```



```

10 hf_butt=freq(hs_butt(2),hs_butt(3),%i*fr);
11 hm_butt = abs(hf_butt);
12 hf_ellip=freq(hs_ellip(2),hs_ellip(3),%i*fr);
13 hm_ellip = abs(hf_ellip);
14 //Plotting Magnitude Response of Analog IIR Filters
15 a = gca();
16 plot2d(fr,hm_butt)
17 poly1 = a.children(1).children(1);
18 poly1.foreground = 2;
19 poly1.thickness = 2;
20 poly1.line_style = 3;
21 plot2d(fr,hm_ellip)
22 poly1 = a.children(1).children(1);
23 poly1.foreground = 5;
24 poly1.thickness = 2;
25 xlabel('Frequency (Hz)')
26 ylabel('Magnitude of frequency response')
27 legend(['Butterworth Filter';'Elliptic Filter'])

```

---

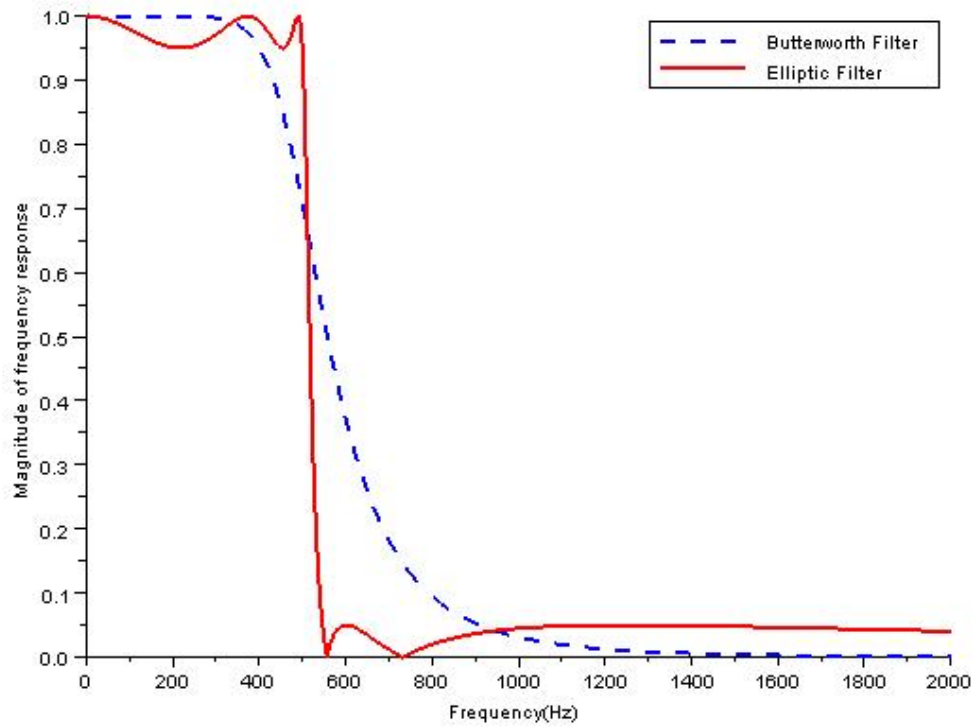


Figure 6.4: Results of Exa 6.3

#### Example 6.4 Bode Plot

```

1 //Example 6.4:Bode Plot
2 s = %s;
3 //Open Loop Transfer Function
4 H = syslin('c',[20000/(s^2+100*s+10000)]); //jw
   replaced by s
5 clf;
6 bode(H,0.01,10000)

```

---

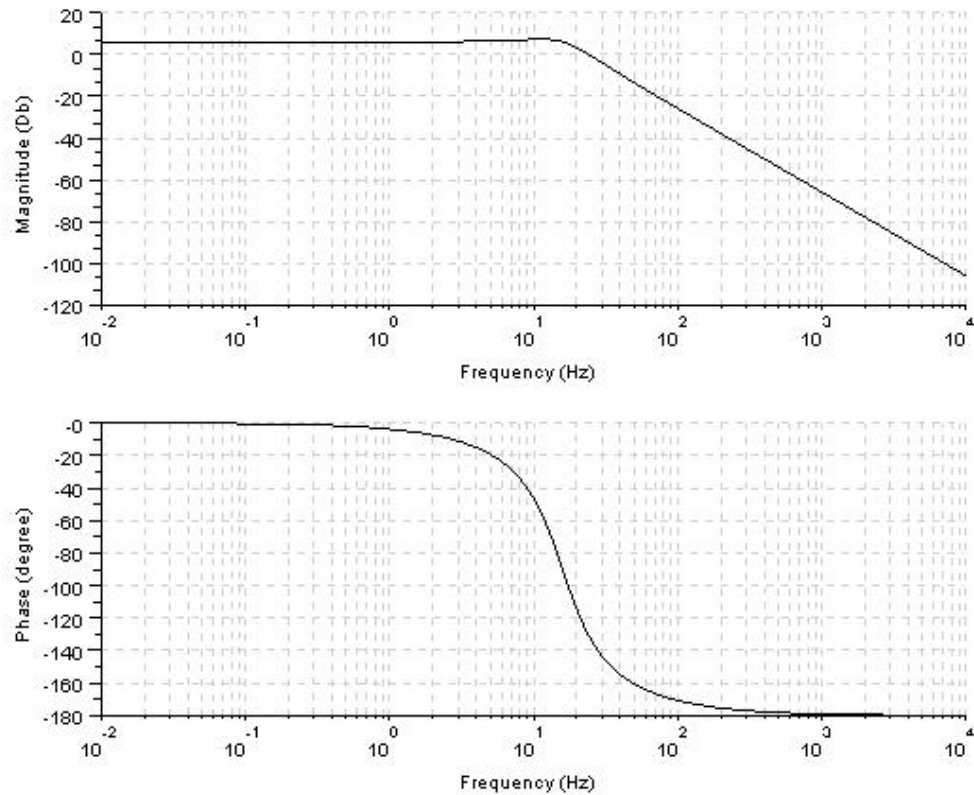


Figure 6.5: Results of Exa 6.4

#### Example 6.5 Bode Plot

```

1 //Example 6.5:Bode Plot
2 s = %s;
3 //Open Loop Transfer Function
4 H = syslin('c',[(100*(1+s))/((10+s)*(100+s))]); //jw
    replaced by s
5 clf;
6 bode(H,0.01,10000)

```

---

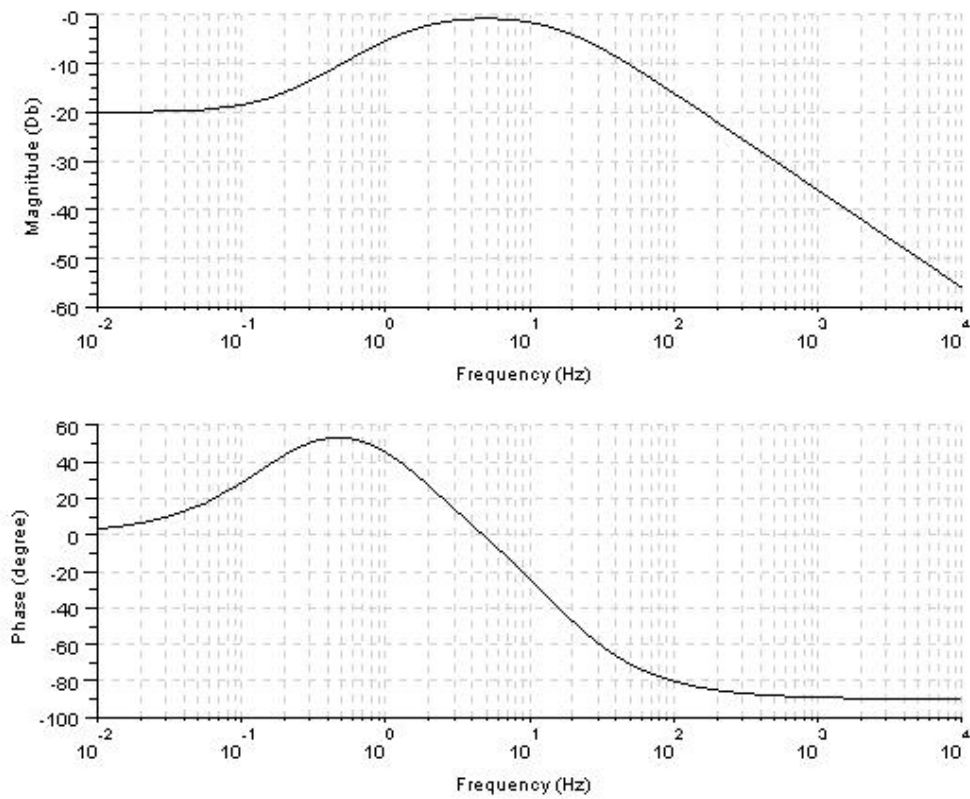


Figure 6.6: Results of Exa 6.5

# Chapter 7

## Sampling

### 7.1 Scilab Codes

Example 7.1 Sinusoidal signal

```
1 //Example7.1: Sinusoidal signal
2 clear;
3 close;
4 clc;
5 Wm = 2*%pi;
6 Ws = 2*Wm;
7 t = -2:0.01:2;
8 phi = -%pi/2;
9 x = cos((Ws/2)*t+phi);
10 y = sin((Ws/2)*t);
11 subplot(2,1,1)
12 a = gca();
13 a.x_location = "origin";
14 a.y_location = "origin";
15 plot(t,x)
16 title('cos(Ws/2*t+phi)')
17 subplot(2,1,2)
18 a = gca();
19 a.x_location = "origin";
20 a.y_location = "origin";
```

```

21 plot(t,y)
22 title('sin(Ws/2*t)')

```

---

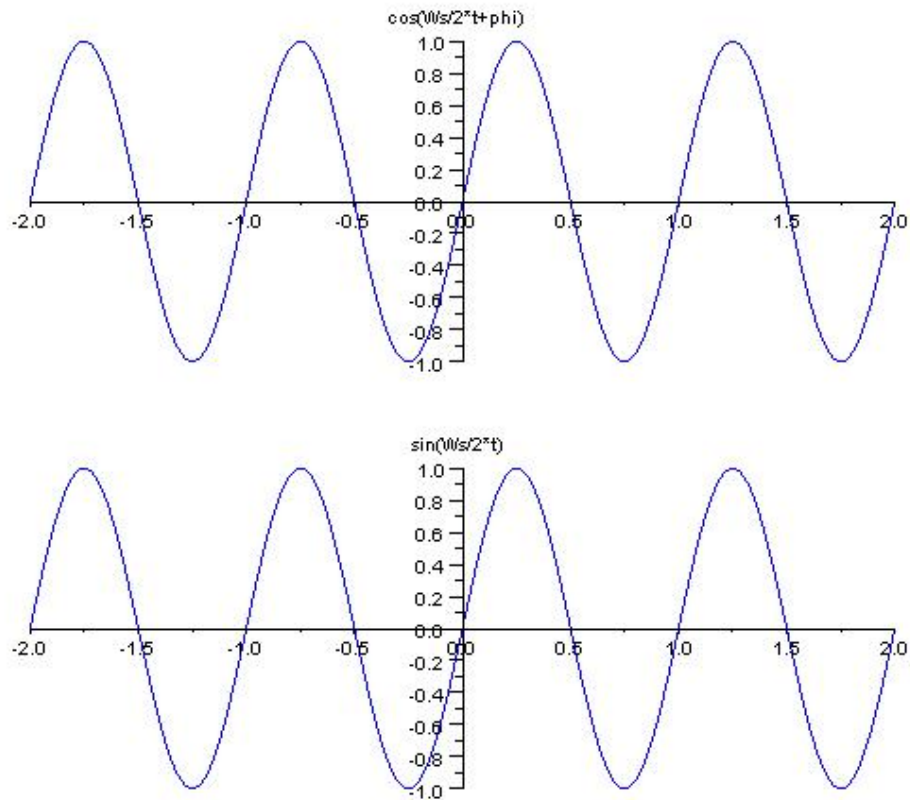


Figure 7.1: Results of Exa 7.1

### Example 7.2 Digital Differentiator

```

1 //Example7.2: Digital Differentiator
2 syms t n;
3 T = 0.1; //Sampling time in seconds
4 xct = sin(%pi*t/T)/(%pi*t);
5 yct = diff(xct,t);
6 disp(yct,'yc(t)=');

```

```

7  t = n*T;
8  xdn = sin(%pi*t/T)/(%pi*t);
9  ydn = diff(xdn,n);
10 disp(ydn,'yd[n]=');
11 hdn = T*ydn;
12 disp(hdn,'hd[n]=');
13 //Result
14 //yc(t) = (10*cos(31.415927*t)/t) - (0.3183099*sin
      (31.415927*t)/(t^2))
15 //yd[n]=(10*cos(3.1415927*n)/n) - 3.183*sin(3.1415927*
      n)/(n^2)
16 //hd[n]=(cos(3.1415927*n)/n) - 0.3183*sin(3.1415927*n)
      /(n^2)

```

---

### Example 7.3 Half Sample Delay system

```

1  //Example7.3:Half Sample Delay system
2  syms t n T;
3  //T = 0.1; //Sampling time in seconds
4  xct = sin(%pi*t/T)/(%pi*t);
5  t = t-T/2;
6  yct_del = sin(%pi*t/T)/(%pi*t);
7  disp(yct_del,'Output of Half Sample delay system
      continuous =');
8  t = n*T-T/2;
9  xdn = sin(%pi*t/T)/(%pi*t);
10 ydn_del = xdn;
11 disp(ydn_del,'Output of Half Sample delay system
      discrete =');
12 hdn = T*ydn_del;
13 disp(hdn,'Impulse Response of discrete time half
      sample delay system=');
14 //Result
15 //Output of Half Sample delay system continuous =
16 //sin(3.14*(t-T/2)/T)/(3.14*(t-T/2))
17 //Output of Half Sample delay system discrete =
18 // sin(3.14*(n*T-T/2)/T)/(3.14*(n*T-T/2))

```

```

19 // Impulse Response of discrete time half sample
    delay system=
20 //  $T \sin(3.14 \cdot (nT - T/2) / T) / (3.14 \cdot (nT - T/2))$ 

```

---

**Example 7.4** Finding the period of the sampled signal and Sampling frequency

```

1 //Example7.4:Finding the period of the sampled
    signal
2 //and Sampling frequency
3 clear;
4 close;
5 clc;
6 Wm = 2*%pi/9;
7 N = floor(2*%pi/(2*Wm))
8 disp(N,'Period of the discrete signal')
9 Ws = 2*%pi/N;
10 disp(Ws,'The Sampling frequency corresponding to the
    period N')

```

---

**Example 7.5** Multirate Signal Processing:Sampling Rate Conversion (1)Downsampling by 4 (2)Upsampling by 2 (3)Upsampling by 2 and followed by downsampling by 9

```

1 //Example7.5:Multirate Signal Processing:Sampling
    Rate Conversion
2 //(1)Downsampling by 4
3 //(2)Upsampling by 2
4 //(3)Upsampling by 2 and followed by downsampling by
    9
5 clear;
6 close;
7 clc;
8 Wm = 2*%pi/9; //Maximum frequency of signal
9 Ws = 2*Wm; //Sampling frequency
10 N = floor(2*%pi/Ws); //period of discrete signal
11 //Original discrete time signal generation and
    Magnitude response

```



```

12 n = 0:0.01:N;
13 x = sin(Wm*n);
14 Wmax = 2*pi/9;
15 K = 4;
16 k = 0:(K/1000):K;
17 W = k*Wmax/K;
18 XW = x* exp(-sqrt(-1)*n'*W);
19 XW_Mag = real(XW);
20 XW_Mag = XW_Mag/max(XW_Mag);
21 W = [-mtlbfliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
22 XW_Mag = [mtlbfliplr(XW_Mag), XW_Mag(2:1001)];
23 //(1)downsampling by 4 and corresponding magnitude
    response
24 n1 = 0:0.01:N/4;
25 y = x(1:4:length(x));
26 k1 = 0:(K/2000):K;
27 W1 = k1*4*Wmax/K;
28 XW4 = y* exp(-sqrt(-1)*n1'*W1);
29 XW4_Mag = real(XW4);
30 XW4_Mag = XW4_Mag/max(XW4_Mag);
31 W1 = [-mtlbfliplr(W1), W1(2:$)]; // Omega from -
    Wmax to Wmax
32 XW4_Mag = [mtlbfliplr(XW4_Mag), XW4_Mag(2:$)];
33 //(2)Upsampling by 2 and corresponding magnitude
    response
34 n2 = 0:0.01:2*N;
35 z = zeros(1,length(n2));
36 z([1:2:length(z)]) = x;
37 k2 = 0:(K/500):K;
38 W2 = k2*Wmax/(2*K);
39 XW2 = z* exp(-sqrt(-1)*n2'*W2);
40 XW2_Mag = real(XW2);
41 XW2_Mag = XW2_Mag/max(XW2_Mag);
42 W2 = [-mtlbfliplr(W2), W2(2:$)]; // Omega from -
    Wmax to Wmax
43 XW2_Mag = [mtlbfliplr(XW2_Mag), XW2_Mag(2:$)];
44 //(3)Upsampling by 2 and Downsampling by 9

```

```

    corresponding magnitude response
45 n3 = 0:0.01:2*N/9;
46 g = z([1:9:length(z)]);
47 k3 = 0:K/(9*500):K;
48 W3 = k3*9*Wmax/(2*K);
49 XW3 = g* exp(-sqrt(-1)*n3'*W3);
50 XW3_Mag = real(XW3);
51 XW3_Mag = XW3_Mag/max(XW3_Mag);
52 W3 = [-mtlb_fliplr(W3), W3(2:$)]; // Omega from -
    Wmax to Wmax
53 XW3_Mag = [mtlb_fliplr(XW3_Mag), XW3_Mag(2:$)];
54 //
55 figure
56 subplot(2,2,1)
57 a = gca();
58 a.y_location = "origin";
59 a.x_location = "origin";
60 a.data_bounds = [-%pi,0;%pi,1.5];
61 plot2d(W,XW_Mag,5);
62 title('Spectrum of Discrete Signal X(exp(jW))')
63 subplot(2,2,2)
64 a = gca();
65 a.y_location = "origin";
66 a.x_location = "origin";
67 a.data_bounds = [-%pi,0;%pi,1.5];
68 plot2d(W1,XW4_Mag,5);
69 title('Spectrum of downsampled signal by 4 X(exp(jW
    /4))')
70 subplot(2,2,3)
71 a = gca();
72 a.y_location = "origin";
73 a.x_location = "origin";
74 a.data_bounds = [-%pi,0;%pi,1.5];
75 plot2d(W2,XW2_Mag,5);
76 title('Spectrum of Upsampled signal by 2 X(exp(2jW)
    )')
77 subplot(2,2,4)
78 a = gca();

```

```

79 a.y_location =" origin";
80 a.x_location =" origin";
81 a.data_bounds =[-%pi,0;%pi,1.5];
82 plot2d(W3,XW3_Mag,5);
83 title('Spectrum of Upsampled by 2 and Downsampled by
      9 X(exp(2jW/9))')

```

---

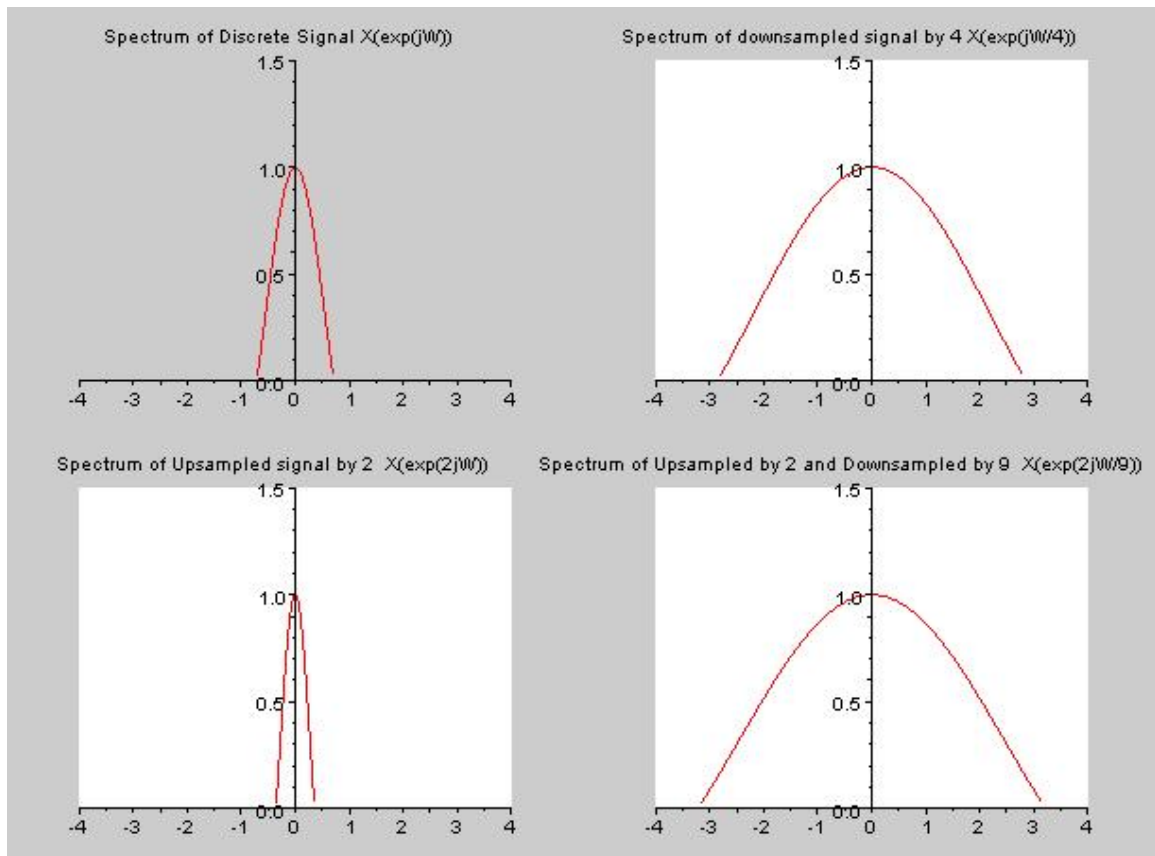


Figure 7.2: Results of Exa 7.5

# Chapter 9

## The Laplace Transform

### 9.1 Scilab Codes

**Example 9.1** Lapalce Transform  $x(t) = \exp(-at).u(t)$

```
1 //Example9.1:Lapalce Transform x(t) = exp(-at).u(t)
2 syms t s;
3 a = 3;
4 y =laplace( '%e^(-a*t)',t,s);
5 disp(y)
6 //Result
7 //1/(s+a)
```

---

**Example 9.2** Lapalce Transform  $x(t) = \exp(-at).u(-t)$

```
1 //Example9.2:Lapalce Transform x(t) = -exp(-at).u(-t)
2 syms t s;
3 a =3;
4 y = laplace( '%e^(a*-t)',t,s);
5 disp(y)
6 //Result
7 //1/(s+a)
```

---

**Example 9.3** Lapalce Transform  $x(t) = 3\exp(-2t)u(t) - 2\exp(-t)u(t)$

```

1 //Example9.3:Lapalce Transform  $x(t) = 3\exp(-2t)u(t) - 2\exp(-t)u(t)$ 
2 syms t s;
3 y = laplace('3*%e^(-2*t)-2*%e^(-t)',t,s);
4 disp(y)
5 //Result
6 //(3/(s+2))-(2/(s+1))

```

---

**Example 9.4** Lapalce Transform  $x(t) = \exp(-2t)u(t) + \exp(-t)(\cos 3t)u(t)$

```

1 //Example9.4:Lapalce Transform  $x(t) = \exp(-2t)u(t) + \exp(-t)(\cos 3t)u(t)$ 
2 syms t s;
3 y = laplace('%e^(-2*t)+%e^(-t)*cos(3*t)',t,s);
4 disp(y)
5 //Result
6 //[(s+1)/(s^2+2*s+10)]+[1/(s+2)] refer equation 9.29
7 //Equivalent to (2*s^2+5*s+12)/((s^2+2*s+10)*(s+2)) refer equation 9.30

```

---

**Example 9.5** Lapalce Transform  $x(t) = s(t) - (4/3)\exp(-t)u(t) + (1/3)\exp(2t)u(t)$

```

1 //Example9.5:Lapalce Transform of  $x(t) = s(t) - (4/3)\exp(-t)u(t) + (1/3)\exp(2t)u(t)$ 
2 syms t s;
3 y = laplace('-(4/3)*%e^(-t)+(1/3)*%e^(2*t)',t,s);
4 y = 1+y;
5 disp(y)
6 //Result
7 //[-4/(3*(s+1))]+[1/(3*(s-2))]+1

```

---

**Example 9.6** Lapalce Transform  $x(t) = \exp(-at)u(t), 0 < t < T$

```

1 //Example9.6
2 //Lapalce Transform  $x(t) = \exp(-at)u(t), 0 < t < T$ 
3 syms t s;

```

```

4 a = 3;
5 T = 10;
6 //t = T;
7 y = laplace(' %e^(-a*t)-%e^(-a*t)*%e^(-(s+a)*T) ',t,s)
      ;
8 disp(y)
9 //Result
10 // [1/(s+a)] - [(exp((-s-a)*T))/(s+a)]

```

---

**Example 9.7** Lapalce Transform  $x(t) = \exp(-b.\text{abs}(t)).u(t), 0 < t < T$   
 $x(t) = \exp(-bt).u(t) + \exp(bt).u(-t)$

```

1 //Example9.7
2 //Lapalce Transform x(t) = exp(-b.abs(t)).u(t), 0<t<
  T
3 //x(t) = exp(-bt).u(t)+exp(bt).u(-t)
4 syms t s;
5 b = 3;
6 y = laplace(' %e^(-b*t)-%e^(b*t) ',t,s);
7 disp(y)
8 //Result
9 // [1/(s+b)] - [1/(s-b)]

```

---

**Example 9.8** Inverse Lapalce Transform  $X(S) = 1/((s+1)(s+2))$

```

1 //Example9.8:Inverse Lapalce Transform
2 //X(S) = 1/((s+1)(s+2))
3 s =%s ;
4 G =syslin('c',(1/((s+1)*(s+2)))) ;
5 disp(G,"G( s )=")
6 plzr(G)
7 x=denom(G) ;
8 disp(x,"Ch a r a c t e r i s t i c s Polynomial=")
9 y = roots(x) ;
10 disp(y,"Poles of a system=")
11 //Result
12 // -1 and -2

```

---

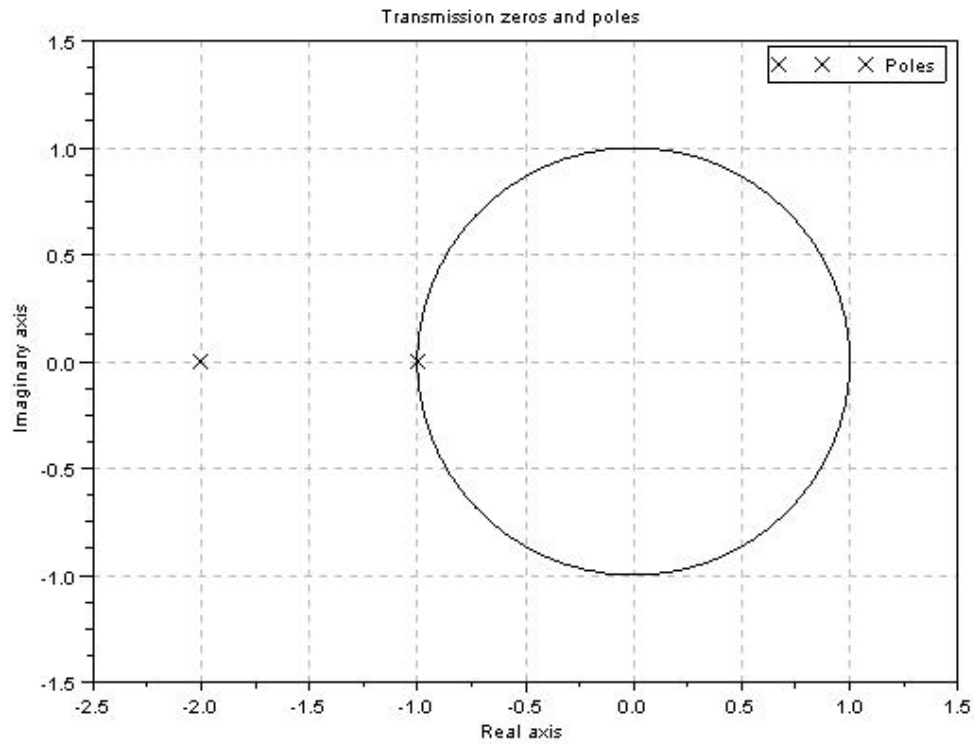


Figure 9.1: Results of Exa 9.8

**Example 9.9** Inverse Laplace Transform  $X(S) = 1/((s + 1)(s + 2))$

```

1 //Example9.9:Inverse Laplace Transform
2 //X(S) = 1/((s+1)(s+2))
3 s =%s ;
4 syms t ;
5 [A]=pfs(1/((s+1)*(s+2))) //partial fraction of F(s)
6 F1 = ilaplace(A(1),s,t)
7 F2 = ilaplace(A(2),s,t)
8 F=F1+F2;
9 disp(F," f (t)=")
10 //Result

```

```
11 // (%e^-t) - (%e^-(2*t))
```

---

**Example 9.10** Inverse Lapalce Transform  $X(S) = 1/((s+1)(s+2))$   $Re(s) < -1, Re(s) < -2$

```
1 //Example9.10: Inverse Lapalce Transform
2 //X(S) = 1/((s+1)(s+2)) Re(s)< -1,Re(s)< -2
3 s =%s ;
4 syms t ;
5 [A]=pfs(1/((s+1)*(s+2))) //partial fraction of F(s)
6 F1 = ilaplace(A(1),s,t)
7 F2 = ilaplace(A(2),s,t)
8 F = -F1-F2;
9 disp(F,"f(t)=")
10 //Result
11 // %e^-(2*t)-%e^-t
```

---

**Example 9.11** Inverse Lapalce Transform  $X(S) = 1/((s+1)(s+2)) - 2 < Re(s) < -1$

```
1 //Example9.11: Inverse Lapalce Transform
2 //X(S) = 1/((s+1)(s+2)) -2< Re(s)< -1
3 s =%s ;
4 syms t ;
5 [A]=pfs(1/((s+1)*(s+2))) //partial fraction of F(s)
6 F1 = ilaplace(A(1),s,t)
7 F2 = ilaplace(A(2),s,t)
8 F = -F1+F2;
9 disp(F,"f(t)=")
10 //Result
11 // -(%e^-t) - (%e^-(2*t))
```

---

**Example 9.12** Inverse Lapalce Transform  $X(S) = 1/(s + (1/2))$   $Re(s) > -1/2$

```
1 //Example9.12: Inverse Lapalce Transform
2 //X(S) = 1/(s+(1/2)) Re(s)> -1/2
```



```

3 s =%s ;
4 G =syslin('c',(1/(s+0.5)))
5 disp(G,"G( s )=")
6 plzr(G)

```

---

**Example 9.13** Inverse Lapalce Transform  $X1(S) = 1/(s+1)Re(s) > -1$ ,  $X2(S) = 1/((s+1)(s+2))Re(s) > -1$

```

1 //Example9.13
2 //Inverse Lapalce Transform
3 //X1(S) = 1/(s+1) Re(s)> -1
4 //X2(S) = 1/((s+1)(s+2)) Re(s)>-1
5 s =%s ;
6 syms t ;
7 G1 =syslin('c',(1/(s+1)));
8 disp(G1,"G( s )=")
9 figure
10 plzr(G1)
11 G2 =syslin('c',(1/((s+1)*(s+2))));
12 disp(G2,"G( s )=")
13 figure
14 plzr(G2)
15 G3 = syslin('c',(1/(s+1))-(1/((s+1)*(s+2))));
16 disp(G3,"G( s )=")
17 figure
18 plzr(G3)

```

---

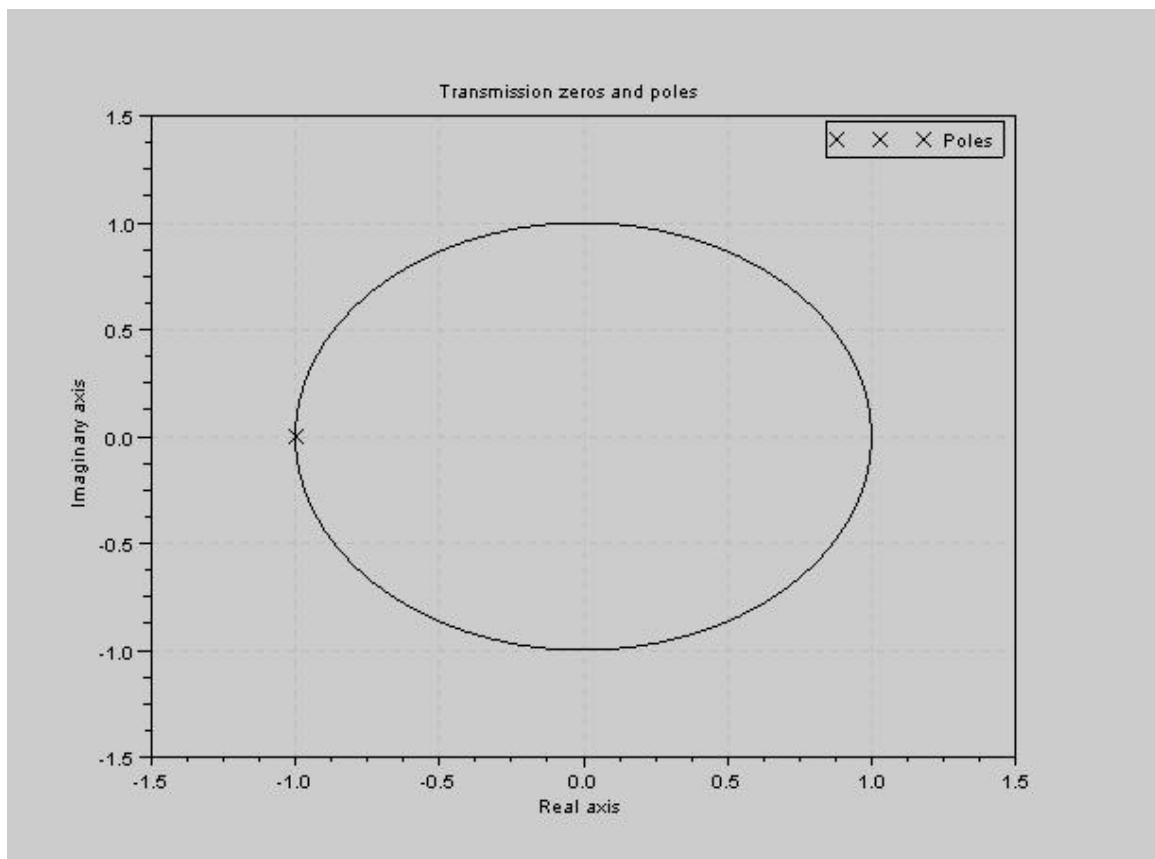


Figure 9.2: Results of Exa [9.13](#)

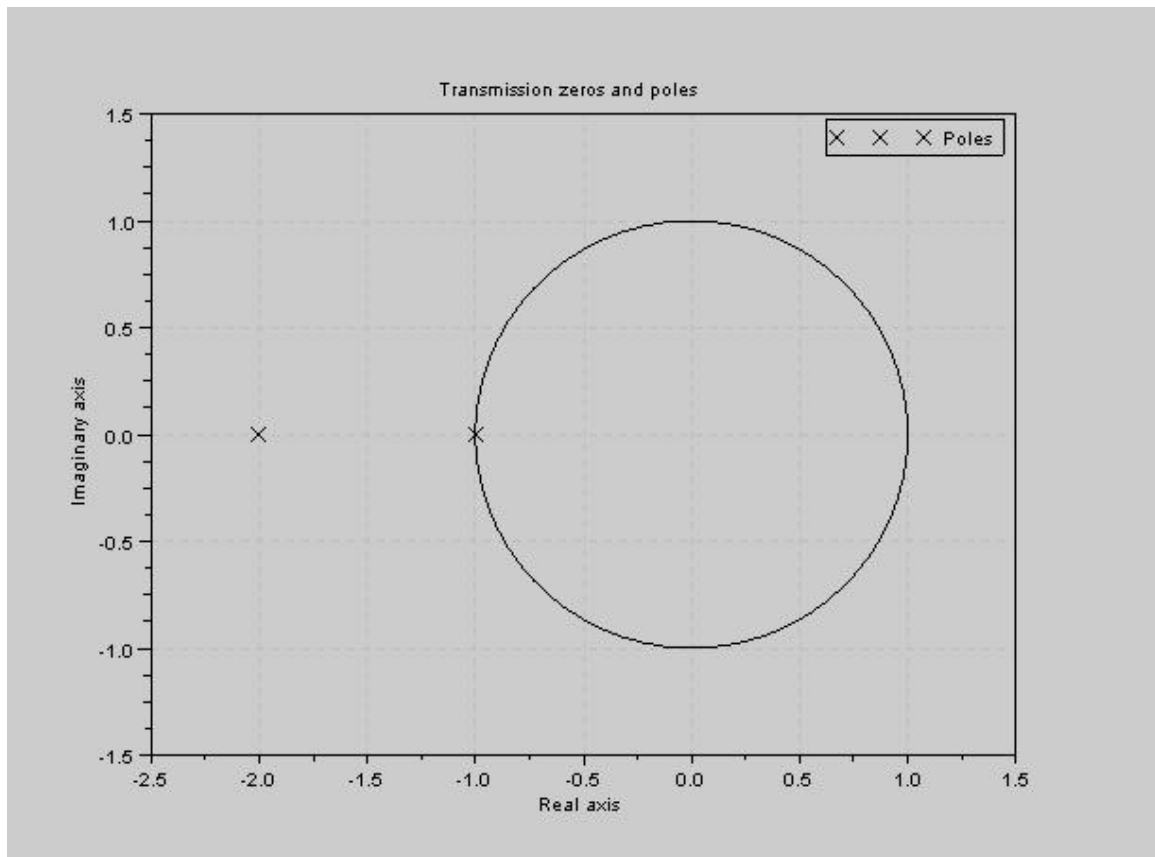


Figure 9.3: Results of Exa [9.13](#)

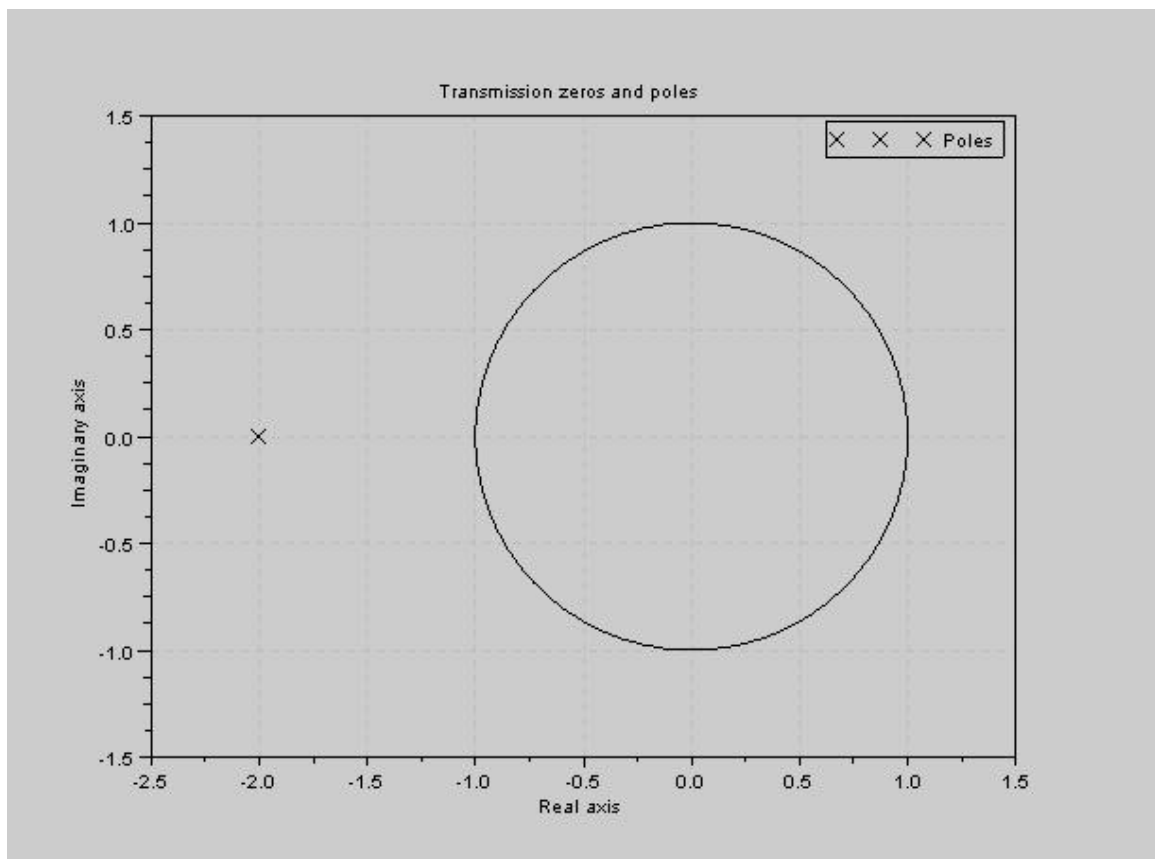


Figure 9.4: Results of Exa 9.13

**Example 9.14** Lapalce Transform  $x(t) = t.exp(-at), t > 0, x(t) = (t^2)/2.exp(-at), t > 0$

```

1 //Example9.14:Lapalce Transform
2 //x(t) = t.exp(-at), t>0
3 //x(t) = (t^2)/2.exp(-at), t>0
4 s =%s ;
5 syms t ;
6 a =10;
7 x1 = laplace('t*%e^(-10*t)',t,s);
8 disp(x1)
9 x2 = laplace('((t^2)/2)*%e^(-10*t)',t,s);

```

```

10 disp(x2)
11 // Result
12 // 1/((s+10)^2)
13 // 1/((s+10)^3)

```

---

**Example 9.15** Inverse Laplace Transform  $X(S) = (2s^2 + 5s + 5)/((s + 1)^2)(s + 2) \operatorname{Re}(s) > -1$

```

1 //Example9.15:Inverse Laplace Transform
2 //X(S) = (2s^2+5s+5)/((s+1)^2)(s+2) Re(s)>-1
3 s =%s ;
4 syms t ;
5 [A]=pfs((2*(s^2)+5*s+5)/(((s+1)^2)*(s+2))); //
    partial fraction of F(s)
6 F1 = ilaplace(A(1),s,t)
7 F2 = ilaplace(A(2),s,t)
8 //F3 = ilaplace(A(3),s,t)
9 F = F1+F2;
10 disp(F,"f(t)=")
11 // Result
12 //(2*t*(%e^-t))-(%e^-t)+(3*%e^-(2*t))

```

---

**Example 9.16** Initial Value Theorem of Laplace Transform

```

1 //Example9.16:Initial Value Theorem of Laplace
    Transform
2 syms s;
3 num =poly([12 5 2], 's', 'coeff')
4 den =poly([20 14 4 1], 's', 'coeff')
5 X = num/den
6 disp(X,"X(s)=")
7 SX = s*X;
8 Initial_Value =limit(SX,s,%inf);
9 disp(Initial_Value,"x(0)=")
10 // Result
11 //(2*%inf^3+5*%inf^2+12*%inf)/(%inf^3+4*%inf^2+14*
    %inf+20) =2

```

---

**Example 9.17** Analysis and Characterization of LTI System Lapalce Transform  $h(t) = \exp(-t).u(t)$

```

1 //Example9.17: Analysis and Characterization of LTI
  System
2 //Lapalce Transform h(t) = exp(-t).u(t)
3 syms t s;
4 h = laplace( '%e^(-t)', t, s );
5 disp(h)
6 //Result
7 // 1/(s+1)

```

---

**Example 9.18** Analysis and Characterization of LTI System Lapalce Transform  $x(t) = \exp(-\text{abs}(t))x(t) = \exp(-t).u(t) + \exp(t).u(-t)$

```

1 //Example9.18: Analysis and Characterization of LTI
  System
2 //Lapalce Transform x(t) = exp(-abs(t))
3 //x(t) = exp(-t).u(t)+exp(t).u(-t)
4 syms t s;
5 y = laplace( '%e^(-t)-%e^(t)', t, s );
6 disp(y)
7 //Result
8 // (1/(s+1)) - (1/(s-1))

```

---

**Example 9.19** Analysis and Characterization of LTI System Inverse Lapalce Transform  $X(S) = (e^s)/(s+1)$

```

1 //Example9.19: Analysis and Characterization of LTI
  System
2 //Inverse Lapalce Transform
3 //X(S) = (e^s)/(s+1)
4 syms s t ;
5 h1 = exp(-1); //Inverse Laplace Transform of exp(s)
6 H2 = 1/(s+1);
7 h2 = ilaplace(H2, s, t)
8 h = h1*h2;
9 disp(h, "h(t)=")

```

```

10 // Result
11 // (18089*(%e^-t))/49171 = 0.3678794(%e^-t)

```

---

**Example 9.20** Inverse Lapalce Transform  $X(S) = ((s-1)/((s+1)*(s-2)))$

```

1 //Example9.20:Inverse Lapalce Transform
2 //X(S) = ((s-1)/((s+1)*(s-2)))
3 s =%s ;
4 syms t ;
5 [A] = pfss(s/((s+1)*(s-2)));
6 [B] = pfss(1/((s+1)*(s-2)));
7 F1 = ilaplace(A(1),s,t)
8 F2 = ilaplace(A(2),s,t)
9 F3 = ilaplace(B(1),s,t)
10 F4 = ilaplace(B(2),s,t)
11 F = F1+F2-F3-F4;
12 disp(F,"f(t)=")
13 //Result
14 //f(t)= 33333329999999*exp(2*t)
        /99999970000000+6666664*%e^-t/9999997
15 //i.e. f(t) =0.3333334*exp(2*t)+0.6666666*%e^(-t)
16 //Refer equation 9.120. (1/3)=0.3333 and (2/3) =
        0.66666

```

---

**Example 9.21** Analysis and Characterization of LTI System Lapalce Transform  $h(t) = \exp(2t)u(t)$ ,  $Re(s) > 2$

```

1 //Example9.21:Analysis and Characterization of LTI
    System
2 //Lapalce Transform h(t) = exp(2t)u(t), Re(s)>2
3 syms t s;
4 X = laplace('%e^(2*t)',t,s);
5 disp(X)
6 //Result
7 //1/(s-2)

```

---

**Example 9.25** LTI Systems Characterized by Linear Constant Coefficient differential Equation Finding Transfer function  $H(S)$  of LTI system  $x(t) = \exp(-3t).u(t)y(t) = [\exp(-t) - \exp(-2t)].u(t)$

```

1 //Example9.25:LTI Systems Characterized by Linear
  Constant
2 //Coefficient differential Equation
3 //Finding Transfer function H(S) of LTI system
4 //x(t) = exp(-3t).u(t)
5 //y(t) = [exp(-t)-exp(-2t)].u(t)
6 syms t s;
7 X = laplace('%e^(-3*t)',t,s);
8 Y = laplace('%e^(-t)-%e^(-2*t)',t,s);
9 H = Y/X;
10 disp(H)
11 //Result
12 //(s+3)*(1/(s+1)-1/(s+2))

```

---

**Example 9.31** Causal LTI Systems described by differential equations and Rational System functions Partial Fraction  $H(S) = ((s-1)/((s+1)*(s-2)))$

```

1 //Example9.31:Causal LTI Systems described by
  differential equations
2 //and Rational System functions
3 //Partial Fraction
4 //H(S) = ((s-1)/((s+1)*(s-2)))
5 s =%s ;
6 syms t ;
7 [A] = pfs((2*s^2+4*s-6)/(s^2+3*s+2));
8 disp(A,"H(S)=")
9 //Result H(S)=
10 //// - 8
11 //      -----
12 //      1 + s
13 //      6
14 //      -----
15 //      2 + s
16 //

```



17 // 2

---

**Example 9.33** Unilateral Laplace Transform: Time Shifting Property  $x(t) = \exp(-a(t+1)).u(t+1)$

```
1 //Example9.33: Unilateral Laplace Transform: Time
    Shifting Property
2 //x(t) = exp(-a(t+1)).u(t+1)
3 syms t s;
4 a = 2;
5 X = laplace(' %e^(-a*(t+1)) ', t, s);
6 disp(X)
7 //Result
8 // %e^-a/(s+a)
```

---

**Example 9.34** Unilateral Laplace Transform  $x(t) = s(t) + 2u(t) + e^t.u(t)$

```
1 //Example9.34: Unilateral Laplace Transform
2 //x(t) = s(t)+2u(t)+e^t.u(t)
3 syms t s;
4 a = 2;
5 X = laplace(' 2+%e^(t) ', t, s);
6 Y = 1+X;
7 disp(X)
8 disp(Y)
9 //Result
10 // (2/s)+(1/(s-1))+1
```

---

**Example 9.35** Unilateral Inverse Laplace Transform  $X(S) = 1/((s+1)(s+2))$

```
1 //Example9.35: Unilateral Inverse Laplace Transform
2 //X(S) = 1/((s+1)(s+2))
3 s = %s;
4 syms t;
5 X = 1/((s+1)*(s+2));
6 x = ilaplace(X, s, t);
```

```

7  disp(X)
8  disp(x)
9  // Result
10 // (%e^-t) - (%e^-(2*t))

```

---

**Example 9.36** Unilateral Laplace Transform  $X(S) = ((s^2) - 3)/(s + 2)$

```

1  //Example9.36: Unilateral Laplace Transform
2  //X(S) = ((s^2)-3)/(s+2)
3  s = %s;
4  syms t;
5  [X] = pfss(((s^2)-3)/(s+2));
6  disp(X)

```

---

**Example 9.37** Unilateral Laplace Transform:Solving Differential Equation  $Y(S) = \alpha/(s(s+1)(s+2))$

```

1  //Example9.37: Unilateral Laplace Transform:Solving
    Differential Equation
2  //Y(S) = alpha/(s(s+1)(s+2))
3  s = %s;
4  syms t;
5  alpha = 1;    //Alpha value assigned as some constant
    one
6  [A] = pfss(alpha/(s*(s+1)*(s+2)));
7  F1 = ilaplace(A(1),s,t)
8  F2 = ilaplace(A(2),s,t)
9  F3 = ilaplace(A(3),s,t)
10 F = F1+F2+F3
11 disp(F)
12 //result
13 // (-%e^-t) + ((%e^-(2*t))/2) + (1/2 )

```

---

**Example 9.38** Unilateral Laplace Transform:Solving Differential Equation  $Y(S) = [\beta(s+3)/((s+1)(s+2))] + [\gamma/((s+2)(s+2))] + [\alpha/(s(s+1)(s+2))]$

```

1 //Example9.38: Unilateral Laplace Transform: Solving
   Differential Equation
2 //Y(S)=[beta(s+3)/((s+1)(s+2))]+[gamma/((s+2)(s+2))
   ]+[alpha/(s(s+1)(s+2))]
3 s = %s;
4 syms t;
5 alpha = 2; //input constant
6 beta_B = 3; //intial condition
7 gamma_v = -5; //initial condition
8 Y1 = 1/s;
9 Y2 = 1/(s+1);
10 Y3 = 3/(s+2);
11 Y = Y1-Y2+Y3;
12 disp(Y)
13 y = ilaplace(Y,s,t)
14 disp(y)
15 //result
16 // (-%e^(-t))+3*(%e^(-(2*t)))+1

```

---

# Chapter 10

## The Z-Transform

### 10.1 Scilab Codes

**Example 10.1** Ztransform of  $x[n] = (a)^n \cdot u[n]$

```
1 // Example10.1: Ztransform of x[n] = (a)^n.u[n]
2 syms n z;
3 a = 0.5;
4 x =(a)^n
5 X = symsum(x*(z^(-n)),n,0,%inf)
6 disp(X,"ans=")
7 // Result
8 // 1.0*(2^(-%inf-1)*z^(-%inf-1)-1)/(1/(2*z)-1)
9 // Equivalent to -1/(0.5*(z^-1)-1)
```

---

**Example 10.2** Ztransform of  $x[n] = -a^n \cdot u[-n-1]$

```
1 //Example 10.2: Z transform of x[n] = -a^n. u[-n-1]
2 //a = 0.5
3 clear all;
4 close;
5 clc;
6 syms n z;
7 a = 0.5;
8 x=-(0.5)^(-n)
9 X=symsum(x*(z^(n)),n,1,%inf)
```

```

10 disp(X,"ans=")
11 // Result
12 // -1.0*(2^( %inf+1)*z^( %inf+1)-2*z)/(2*z-1)
13 // Equivalent to -1*-2*z/(2*z-1) = 1/(1-0.5*z^-1)

```

---

**Example 10.3** Ztransform of  $x[n] = 7.(1/3)^n.u[n] - 6.(1/2)^n.u[n]$

```

1 //Example 10.3:Z transform of x[n] = 7.(1/3)^n.u[n
  ] - 6.(1/2)^n.u[n]
2 syms n z;
3 x1=(0.33)^(n)
4 X1=symsum(7*x1*(z^(-n)),n,0,%inf)
5 x2=(0.5)^(n)
6 X2=symsum(6*x2*(z^(-n)),n,0,%inf)
7 X = X1-X2
8 disp(X,"ans=")
9 // Result
10 // -6.0*(2^(- %inf-1)*z^(- %inf-1)-1)/(1/(2*z)-1)
11 // Equivalent to -6*-1/(0.5*z^-1 -1)
12 //The Region of Convergence is |z|>1/2

```

---

**Example 10.4** Z-transform of sine signal

```

1 //Example10.4:Z-transform of sine signal
2 syms n z;
3 Wo =%pi/4;
4 a = (0.33)^n;
5 x1=%e^(sqrt(-1)*Wo*n);
6 X1=symsum(a*x1*(z^(-n)),n,0,%inf)
7 x2=%e^(-sqrt(-1)*Wo*n)
8 X2=symsum(a*x2*(z^(-n)),n,0,%inf)
9 X = (1/(2*sqrt(-1)))*(X1-X2)
10 disp(X,"ans=")

```

---

**Example 10.5** Z-transform of Impulse Sequence

```

1 //Example10.5:Z-transform of Impulse Sequence

```

```

2 syms n z;
3 X=symsum(1*(z^(-n)),n,0,0);
4 disp(X,"ans=")
5 //Result
6 // 1

```

---

**Example 10.6** Ztransform of  $x[n] = a^n, 0 < n < N - 1$

```

1 //Example 10.6:Z transform of x[n] = a^n, 0 < n < N
  -1
2 syms n z;
3 a = 0.5;
4 N =6;
5 x=(a)^(n)
6 X=symsum(x*(z^(-n)),n,0,N)
7 disp(X,"ans=")
8 //Result
9 // 0.5/z+0.25/z^2+0.125/z^3+0.0625/z^4+0.03125/z
  ^5+0.015625/z^6+1.0

```

---

**Example 10.7** Ztransform of  $x[n] = b^n.u[n] + b^{-n}.u[-n - 1]$

```

1 //Example 10.7:Z transform of x[n] = b^n.u[n]+b^-n.u
  [-n-1]
2 syms n z;
3 b = 0.5;
4 x1=(b)^(n)
5 x2=(b)^(-n)
6 X1=symsum(x1*(z^(-n)),n,0,%inf)
7 X2=symsum(x2*(z^(n)),n,1,%inf)
8 X = X1+X2;
9 disp(X,"ans=")
10 //Result
11 // +1.0*(2^(-%inf-1)*z^(-%inf-1)-1)/(1/(2*z)-1)
12 //Equivalent to -1/(0.5*z^-1 - 1)
13 //Region of Convergence |z|>0.5

```

---

**Example 10.9** Inverse Z Transform :ROC  $|z| > 1/3$

```

1 //Example10.9: Inverse Z Transform:ROC  $|z|>1/3$ 
2 z = %z;
3 syms n z1; //To find out Inverse z transform z must
    be linear z = z1
4 X = z*(3*z-(5/6))/((z-(1/4))*(z-(1/3)))
5 X1 = denom(X);
6 zp = roots(X1);
7 X1 = z1*(3*z1-(5/6))/((z1-(1/4))*(z1-(1/3)))
8 F1 = X1*(z1^(n-1))*(z1-zp(1));
9 F2 = X1*(z1^(n-1))*(z1-zp(2));
10 h1 = limit(F1,z1,zp(1));
11 disp(h1,'h1[n]=')
12 h2 = limit(F2,z1,zp(2));
13 disp(h2,'h2[n]=')
14 h = h1+h2;
15 disp(h,'h[n]=')
16 ////Result
17 //h[n]= (1/4)^n+(2/3)^n

```

---

**Example 10.10** Inverse Z Transform :ROC  $1/4 < |z| < 1/3$

```

1 //Example10.10: Inverse Z Transform:ROC  $1/4<|z|<1/3$ 
2 z = %z;
3 syms n z1; //To find out Inverse z transform z must
    be linear z = z1
4 X = z*(3*z-(5/6))/((z-(1/4))*(z-(1/3)))
5 X1 = denom(X);
6 zp = roots(X1);
7 X1 = z1*(3*z1-(5/6))/((z1-(1/4))*(z1-(1/3)))
8 F1 = X1*(z1^(n-1))*(z1-zp(1));
9 F2 = X1*(z1^(n-1))*(z1-zp(2));
10 h1 = limit(F1,z1,zp(1));
11 disp(h1*'u(n)', 'h1[n]=')
12 h2 = limit(F2,z1,zp(2));
13 disp((h2)*'u(-n-1)', 'h2[n]=')
14 disp((h1)*'u(n)'-(h2)*'u(n-1)', 'h[n]=')
15 ////Result
16 // h[n]= u(n)/4^n-2*u(n-1)/3^n

```

---

```
17 //Equivalent to  $h[n] = (1/4)^n \cdot u[n] - 2 \cdot (1/3)^n \cdot u[-n-1]$ 
```

---

**Example 10.11** Inverse Z Transform :ROC  $|z| < 1/4$

```
1 //Example10.11:Inverse Z Transform:ROC |z|<1/4
2 z = %z;
3 syms n z1; //To find out Inverse z transform z must
    be linear z = z1
4 X = z*(3*z-(5/6))/((z-(1/4))*(z-(1/3)))
5 X1 = denom(X);
6 zp = roots(X1);
7 X1 = z1*(3*z1-(5/6))/((z1-(1/4))*(z1-(1/3)))
8 F1 = X1*(z1^(n-1))*(z1-zp(1));
9 F2 = X1*(z1^(n-1))*(z1-zp(2));
10 h1 = limit(F1,z1,zp(1));
11 disp(h1*'u(-n-1)', 'h1[n]=')
12 h2 = limit(F2,z1,zp(2));
13 disp((h2)*'u(-n-1)', 'h2[n]=')
14 disp(-(h1)*'u(-n-1)'-(h2)*'u(-n-1)', 'h[n]=')
15 ///Result
16 // h[n]= -u(-n-1)/4^n-2*u(-n-1)/3^n
17 //Equivalent to  $h[n] = -(1/4)^n \cdot u[-n-1] - 2 \cdot (1/3)^n \cdot u[-n-1]$ 
```

---

**Example 10.12** Inverse z tranform:For Finite duration discrete sequence

```
1 //Example10.12:Inverse z tranform:For Finite
    duration discrete sequence
2 syms z;
3 X = [4*z^2 0 2 3*z^-1];
4 n = -2:1;
5 for i = 1:length(X)
6     x(i) = X(i)*(z^n(i));
7 end
8 disp(x, 'x[n]=')
```

---

**Example 10.13** Inverse z tranform ofInFinite duration discrete sequence  
Power Series Method (OR) Long Division Method



```

1 //Example10.13:Inverse z tranform ofInFinite
   duration discrete sequence
2 //Power Series Method (OR)//Long Division Method
3 z = %z;
4 a = 2;
5 X = ldiv(z,z-a,5)

```

---

**Example 10.18** Ztransform-Differentiation Property  $x[n] = (a)^n.u[n]$

```

1 // Example10.18:Ztransform-Differentiation Property
2 // x[n] = (a)^n.u[n]
3 syms n z;
4 a = 0.5;
5 x =(a)^n
6 X = symsum(x*(z^(-n)),n,0,%inf)
7 X1 = -1/((1/(2*z))-1) //z transform of 0.5^n.u[
   n]
8 Y = -z*diff(X,z) //Differentiation property of z-
   transform
9 disp(X,"ans=")
10 disp(Y,"ans=")
11 //Result
12 //X(z) = 1.0*(2^(-%inf-1)*z^(-%inf-1)-1)/(1/(2*z)-1)
13 //Y(z) = -1.0*(-%inf-1)*2^(-%inf-1)*z^(-%inf-1)
   /(1/(2*z)-1)
14 //Y1(z) = 1/(2*(1/(2*z)-1)^2*z)
15 //Equivalent to Y1(z) = 0.5*z^-1/((1-0.5*z^-1)^2)

```

---

**Example 10.19** Z Transform : Initial Value Theorem

```

1 //Example10.19:Z Transform : Initial Value Theorem
2 z = %z;
3 syms n z1;//To find out Inverse z transform z must
   be linear z = z1
4 X =z*(z-(3/2))/((z-(1/3))*(z-(1/2)))
5 X1 = denom(X);
6 zp = roots(X1);
7 X1 = z1*(z1-(3/2))/((z1-(1/3))*(z1-(1/2)))

```

```

8 F1 = X1*(z1^(n-1))*(z1-zp(1));
9 F2 = X1*(z1^(n-1))*(z1-zp(2));
10 x1 = limit(F1,z1,zp(1));
11 x2 = limit(F2,z1,zp(2));
12 x = x1+x2;
13 disp(x, 'x[n]=')
14 x_initial = limit(x,n,0);
15 disp(x_initial, 'x[0]=')
16 ///Result
17 //x[n]= 7/3^n-3*2^(1-n)
18 //x[0]= 1; Initial Value

```

---

**Example 10.23** Inverse Z Transform  $H(z) = z/z-a$

```

1 //Example10.23:Inverse Z Transform H(z) =z/z-a
2 //z = %z;
3 syms n z;
4 a = 2;
5 H = z/(z-a);
6 F = H*z^(n-1)*(z-a);
7 h = limit(F,z,a);
8 disp(h, 'h[n]=')

```

---

**Example 10.25** LTi Systems characterized by Linear Constant Coefficient Difference equations Inverse Z Transform

```

1 //Example10.25:LTi Systems characterized by Linear
  Constant
2 //Coefficient Difference equations
3 //Inverse Z Transform
4 //z = %z;
5 syms n z;
6 H1 = z/(z-(1/2));
7 H2 = (1/3)/(z-(1/2));
8 F1 = H1*z^(n-1)*(z-(1/2));
9 F2 = H2*z^(n-1)*(z-(1/2));
10 h1 = limit(F1,z,1/2);
11 disp(h1, 'h1[n]=')

```

```

12 h2 = limit(F2,z,1/2);
13 disp(h2,'h2[n]=')
14 h = h1+h2;
15 disp(h,'h[n]=')
16 // Result
17 //h[n]= [(1/2)^n]+[2^(1-n)]/3
18 //Which is Equivalent to h[n] =[(1/2)^n]+[(1/2)^(n
    -1)]/3

```

---

**Example 10.33** Differentiation Property of Unilateral Ztransform  $x[n] = (a)^{(n+1)}.u[n+1]$

```

1 // Example10.33: Differentiation Property of
  Unilateral Ztransform
2 // x[n] = (a)^(n+1).u[n+1]
3 syms n z;
4 a = 0.5;
5 x =(a)^(n+1)
6 X = symsum(x*(z^(-n)),n,-1,%inf)
7 disp(X,'ans=')
8 // Result
9 //X(z)= 0.5*(2^(-%inf-1)*z^(-%inf-1)-2*z)/(1/(2*z)
    -1)
10 //Equivalent to z/(1-0.5*z^-1)

```

---

**Example 10.34** Unilateral Ztransform- partial fraction  $X(z) = (3 - (5/6) * (z^{-1})) / ((1 - (1/4) * (z^{-1})) * (1 - (1/3) * (z^{-1})))$

```

1 // Example10.34: Unilateral Ztransform- partial
  fraction
2 // X(z) =(3-(5/6)*(z^-1))/((1-(1/4)*(z^-1))*(1-(1/3)
    *(z^-1)))
3 z = %z;
4 s = %s;
5 syms n t;
6 a = 0.5;
7 [A]=pfs((3-(5/6)*(z^-1))/((1-(1/4)*(z^-1))*(1-(1/3)
    *(z^-1))))

```

```

8 x1 = horner(A(1),z)
9 x2 = horner(A(2),z)
10 x3 = A(3)
11 x = x1+x2+x3
12 disp(x1,"ans=")
13 disp(x2,"ans=")
14 disp(x3,"ans=")
15 disp(x,"ans=")
16 //Result
17
18 //      0.6666667
19 //      -----
20 // - 0.3333333 + z
21
22 //      0.25
23 //      -----
24 // - 0.25 + z
25
26 //3
27
28 //sum of these, gives the original value
29 //
30 //      2
31 //      - 0.8333333z + 3z
32 //      -----
33 //      2
34 //      0.0833333 - 0.5833333z + z

```

---

**Example 10.36** Output response of an LTI System

```

1 //Example 10.36:To find output response of an LTI
  System
2 syms n z;
3 H = z/(z+3)
4 X = z/(z-1)
5 Y = X*H
6 F1 = Y*(z^(n-1))*(z-1);
7 y1 = limit(F1,z,1);
8 F2 = Y*(z^(n-1))*(z+3);

```

```

9 y2 = limit(F2,z,-3);
10 disp(y1*"u(n)" + y2*"u(n)", 'y[n]=')
11 //Result
12 //y[n] = u(n)/4 - (-3)^(n+1)*u(n)/4
13 //Equivalent to = (1/4).u[n] - (3/4)(-3)^n.u[n]

```

---

**Example 10.37** Output response of an LTI System

```

1 //Example 10.37: To find output response of an LTI
  System
2 syms n z;
3 alpha = 8; //input constant
4 beta_b = 1; //initial condition y[-1] = 1
5 Y1 = -((3*beta_b*z)/(z+3))
6 Y2 = (alpha*z^2/((z+3)*(z-1)))
7 F1 = Y1*(z^(n-1))*(z+3);
8 y1 = limit(F1,z,-3);
9 F2 = Y2*(z^(n-1))*(z+3);
10 y2 = limit(F2,z,-3);
11 F3 = Y2*(z^(n-1))*(z-1);
12 y3 = limit(F3,z,1);
13 disp((y1+y2+y3)*'u(n)', 'y[n]=')
14 //Result
15 //y[n] = (2 - (-3)^(n+1))*u(n)

```

---

# Chapter 11

## Linear Feedback Systems

### 11.1 Scilab Codes

**Example 11.1** Root locus Analysis of Linear Feedback Systems Continuous Time Systems

```
1 //Example11.1:Root locus Analysis of Linear Feedback
   Systems
2 //Continuous Time Systems
3 //Refer figure 11.12(a) in Openhiem &Willksy page
   840
4 s = %s;
5 H = syslin('c',[1/(s+1)]);
6 G = syslin('c',[1/(s+2)]);
7 F = G*H;
8 clf;
9 evans(F,3)
```

---

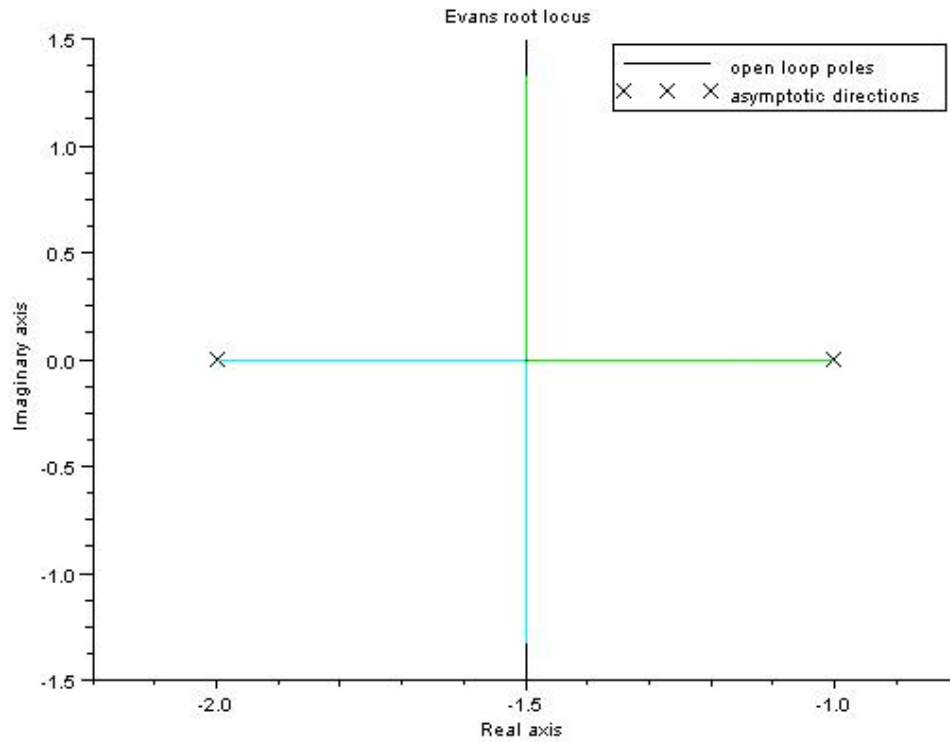


Figure 11.1: Results of Exa 11.1

**Example 11.2** Root locus Analysis of Linear Feedback Systems Continuous Time Systems

```

1 //Example11.2:Root locus Analysis of Linear Feedback
  Systems
2 //Continuous Time Systems
3 //Refer figure 11.14(a) in Openhiem &Willksy page
  844
4 s = %s;
5 G = syslin('c',[(s-1)/((s+1)*(s+2))]);
6 clf;
7 evans(G,2)

```

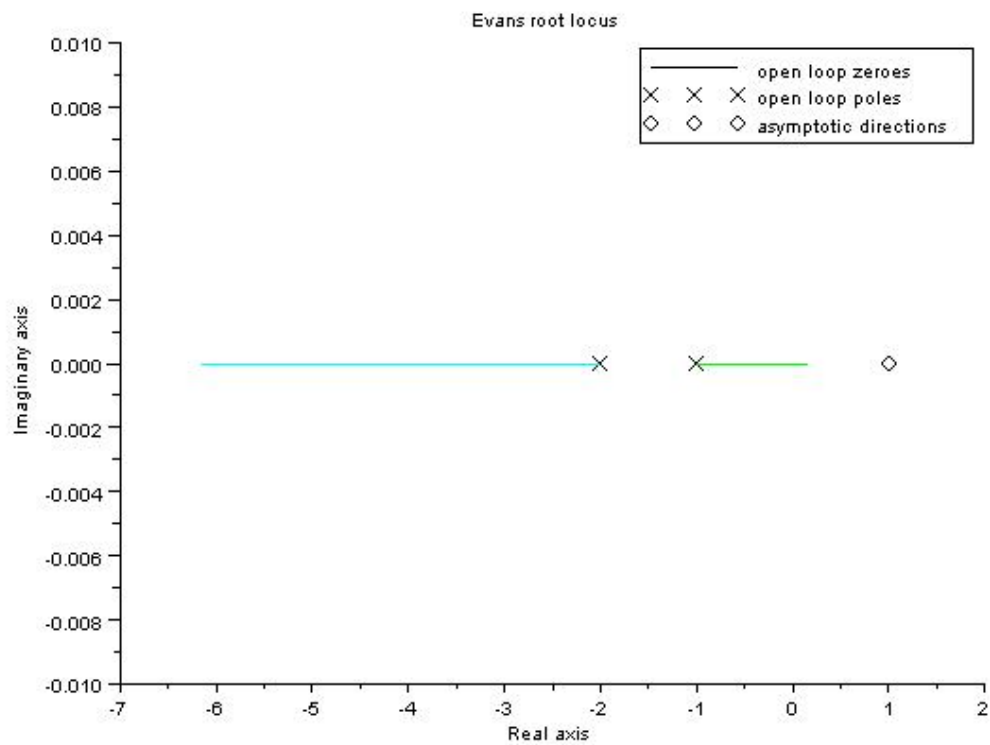


Figure 11.2: Results of Exa 11.2

**Example 11.3** Root locus Analysis of Linear Feedback Systems Discrete time system

```

1 //Example11.3:Root locus Analysis of Linear Feedback
  Systems
2 ////Discrete time system
3 //Refer figure 11.16(a) in Openhiem &Willksy page
  846
4 z = %z;
```



```

5 G = syslin('d',[z/((z-0.5)*(z-0.25))]);
6 clf;
7 evans(G,2)

```

---

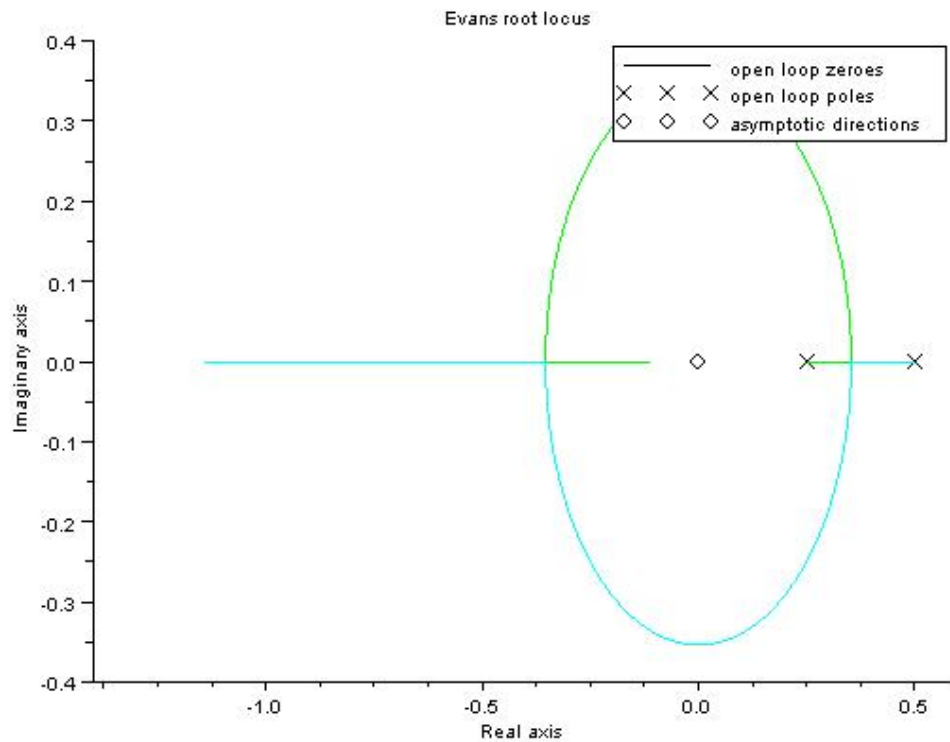


Figure 11.3: Results of Exa 11.3

**Example 11.5** Bode Nyquist criterion for Continuous Time Systems Bode Plot

```

1 //Example 11.5: Nyquist criterion for Continuous Time
  Systems
2 //Bode Plot
3 s = %s;

```

```

4 //Open Loop Transfer Function
5 G = syslin('c',[1/(s+1)]);
6 H = syslin('c',[1/(0.5*s+1)]);
7 F = G*H;
8 clf;
9 bode(F,0.01,100)
10 show_margins(F)

```

---

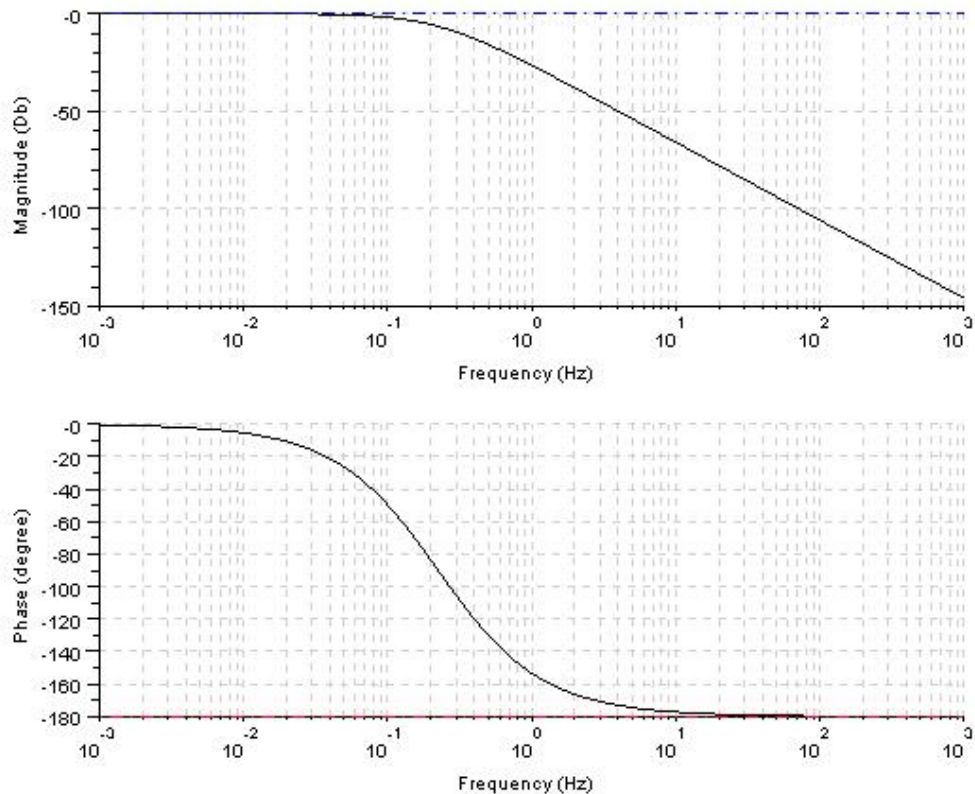


Figure 11.4: Results of Exa 11.5Bode

**Example 11.5Nyquist** Nyquist criterion for Continuous Time Systems Nyquist Plot

```

1 //Example 11.5:Nyquist criterion for Continuous Time
  Systems
2 //Nyquist Plot
3 s = %s;
4 //Open Loop Transfer Function
5 G = syslin('c',[1/(s+1)]);
6 H = syslin('c',[1/(0.5*s+1)]);
7 F = G*H;
8 clf;
9 nyquist(F)
10 show_margins(F,'nyquist')

```

---

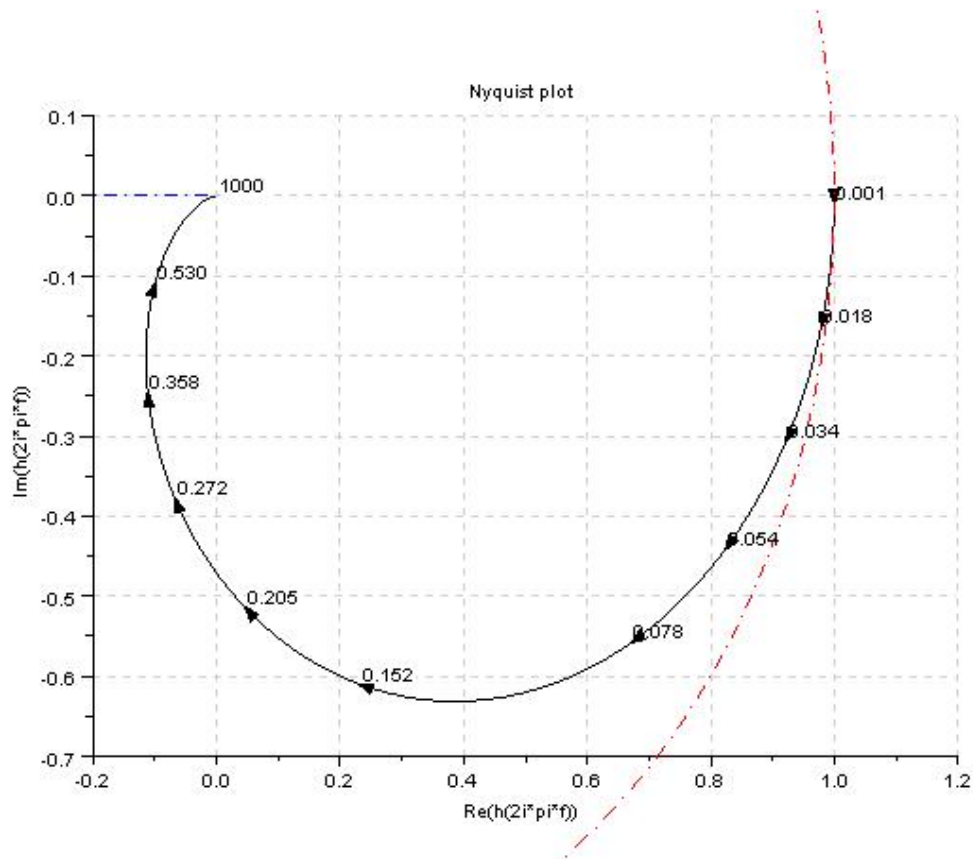


Figure 11.5: Results of Exa 11.5Nyquist

**Example 11.6** Nyquist criterion for Continuous Time Systems Nyquist Plot

```
1 //Example 11.6:Nyquist criterion for Continuous Time
  Systems
2 //Nyquist Plot
3 s = %s;
4 //Open Loop Transfer Function
5 F = syslin('c',[(s+1)/((s-1)*(0.5*s+1))])
6 clf;
7 nyquist(F)
8 show_margins(F,'nyquist')
```

---

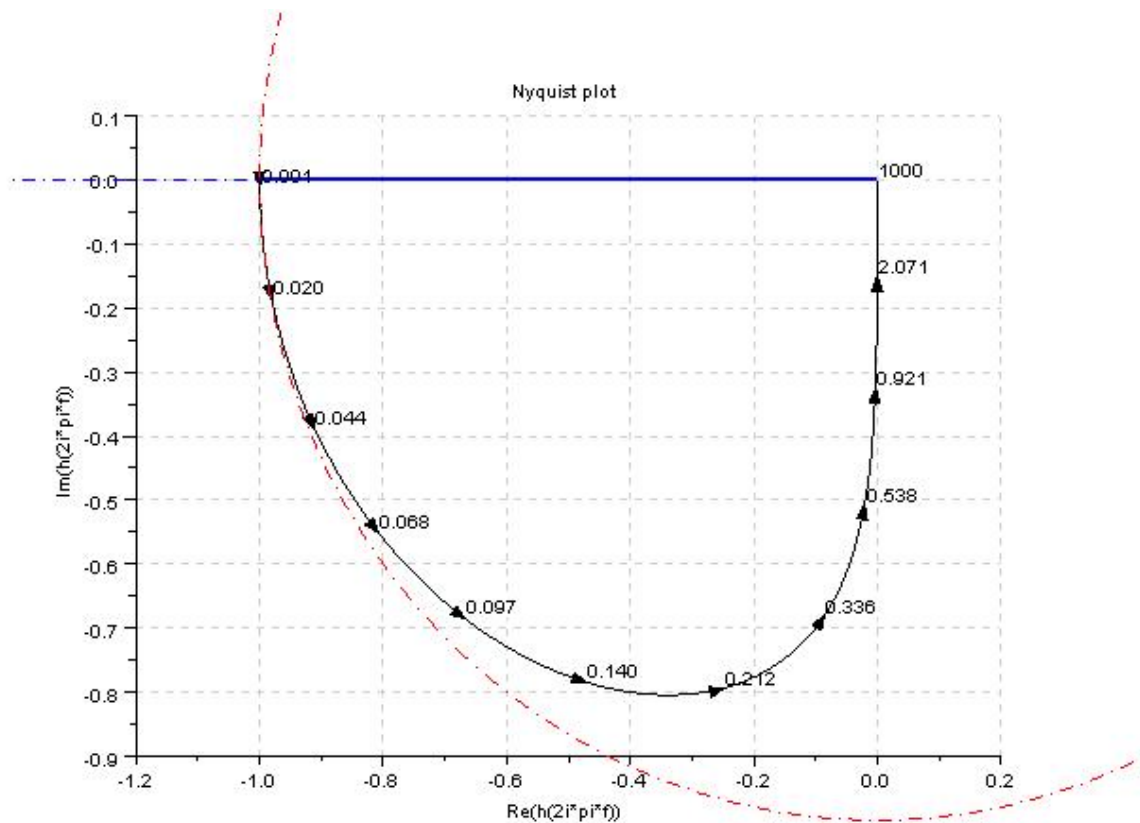


Figure 11.6: Results of Exa 11.6

### Example 11.7 Nyquist Plot

```
1 //Example 11.7
2 //Nyquist Plot
3 s = %s;
4 T =1;
5 //Open Loop Transfer Function
6 G = syslin('c',[-%e^(-s*T)]);
7 clf;
8 nyquist(G)
9 show_margins(G,'nyquist')
```

---

### Example 11.8 Nyquist criterion for Discrete Time Systems Nyquist Plot Discrete Time System

```
1 //Example 11.8:Nyquist criterion for Discrete Time
  Systems
2 //Nyquist Plot
3 //Discrete Time System
4 z = %z;
5 //Open Loop Transfer Function
6 F = syslin('d',[1/(z*(z+0.5))])
7 clf;
8 nyquist(F)
9 show_margins(F,'nyquist')
```

---

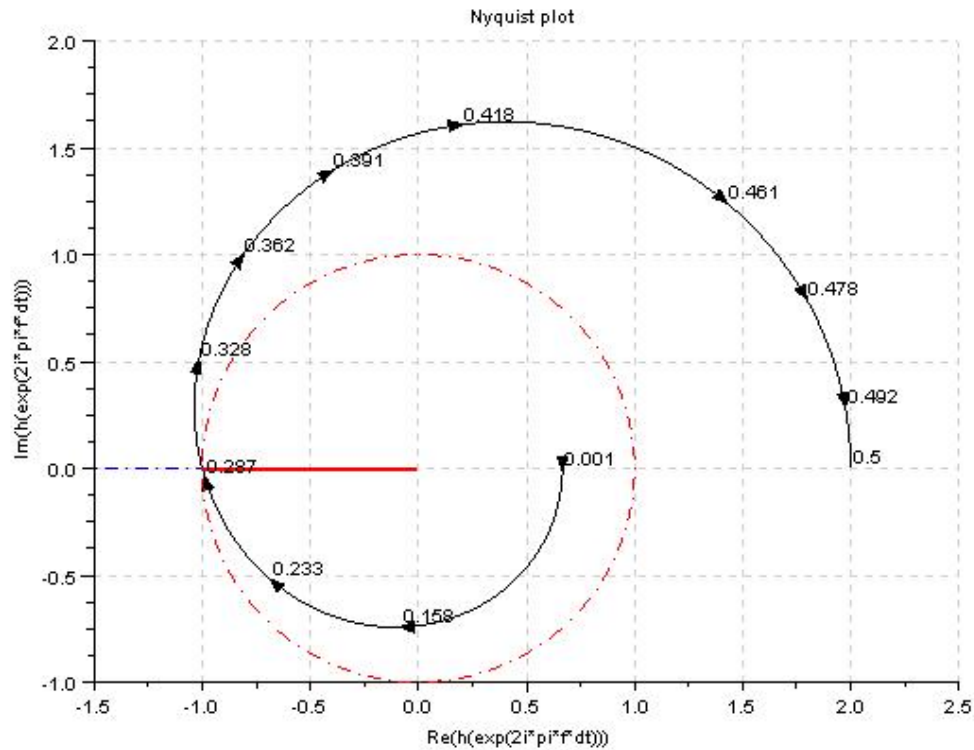


Figure 11.7: Results of Exa 11.8

**Example 11.9** Gain and Phase Margins and their associated cross over frequencies

```

1 //Example 11.9:Gain and Phase Margins and their
2 //associated cross over frequencies
3 s =poly(0,'s'); // Define ss as polynomial variable
4 //Create s transfer function in forward path
5 F = syslin('c',[ (4*(1+0.5*s))/(s*(1+2*s)*(1+0.05*s
    +(0.125*s)^2))]);
6 B = syslin('c',(1+0*s)/(1+0*s))
7 OL = F*B;
8 fmin = 0.01; // Min freq in Hz

```

```

9 fmax = 10; // Max freq in Hz
10 scf(1);
11 //clf;
12 // Plot frequency response of open loop transfer
    function
13 bode(OL,0.01,10);
14 // display gain and phase margin and cross over
    frequencies
15 show_margins(OL);
16 [gm,fr1] = g_margin(OL)
17 [phm,fr2] = p_margin(OL)
18 disp(gm,'gain margin in dB')
19 disp(fr1,'gain cross over frequency in Hz')
20 disp(phm,'phase margin in dB')
21 disp(fr2,'phase cross over frequency in Hz')

```

---

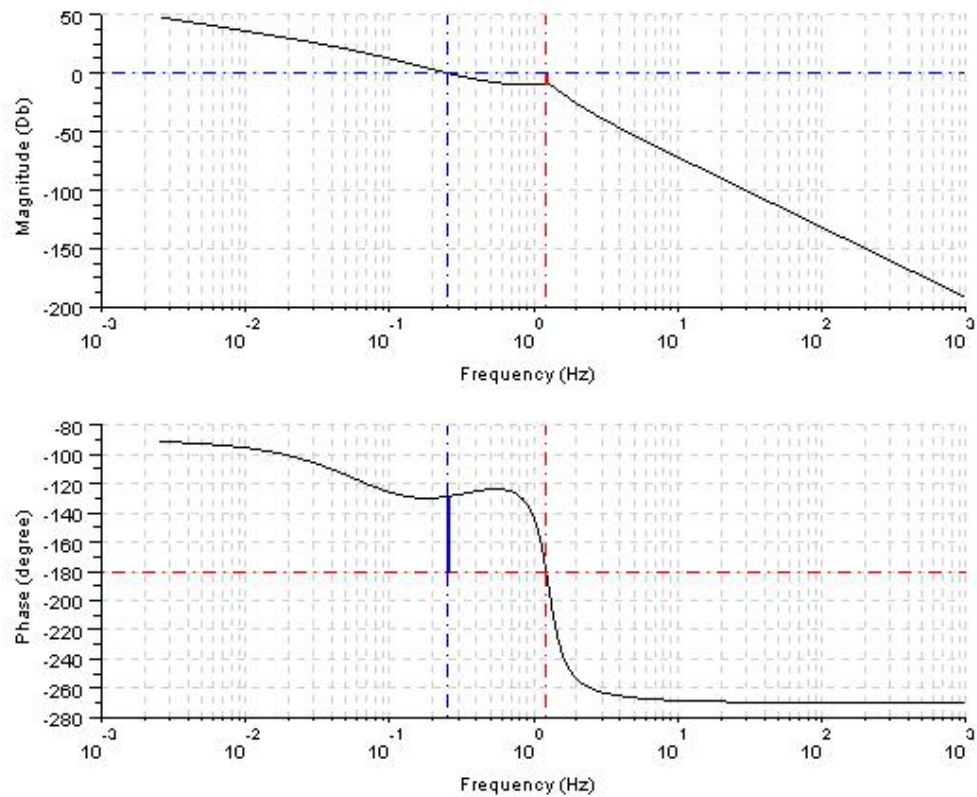


Figure 11.8: Results of Exa 11.9

**Example 11.9** //Figure11.9:Root locus analysis of  
Linear feedback systems

```

2 s = %s;
3 beta_b1 = 1;
4 beta_b2 = -1;
5 G1 = syslin('c',[2*beta_b1/s]);
6 G2 = syslin('c',[2*beta_b2/s]);
7 H = syslin('c',[s/(s-2)]);
8 F1 = G1*H;
9 F2 = G2*H;
10 clf;

```



```
11 evans(F1,2)
12 figure
13 evans(F2,2)
```

---

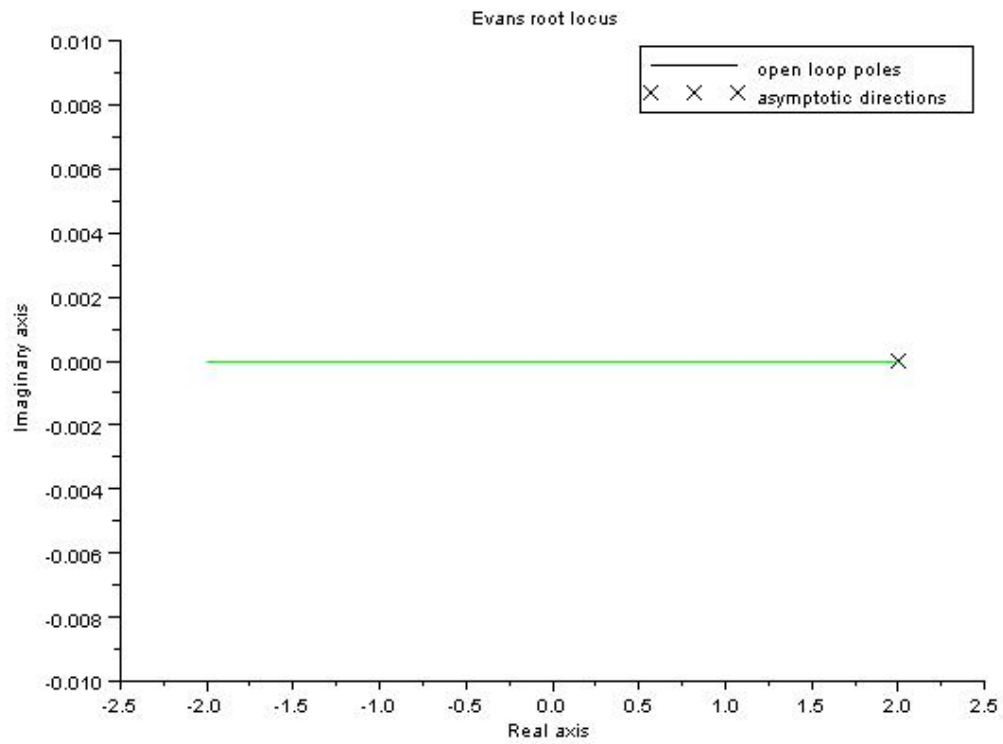


Figure 11.9: Results of Exa [11.9](#)

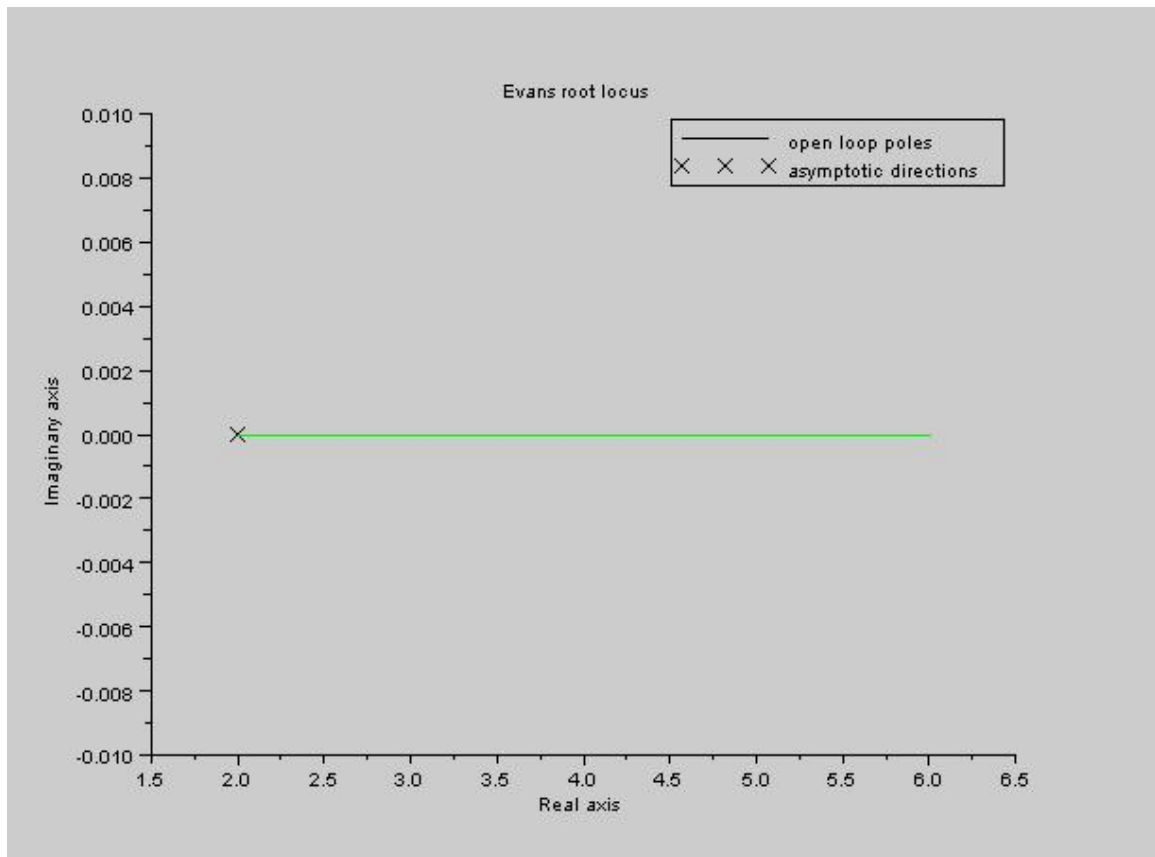


Figure 11.10: Results of Exa [11.9](#)