Computer Assignment #1

ECE 405, Summer session 1, Cal Poly Pomona, CA
By Nasser M. Abbasi

PART(1) LOW PASS

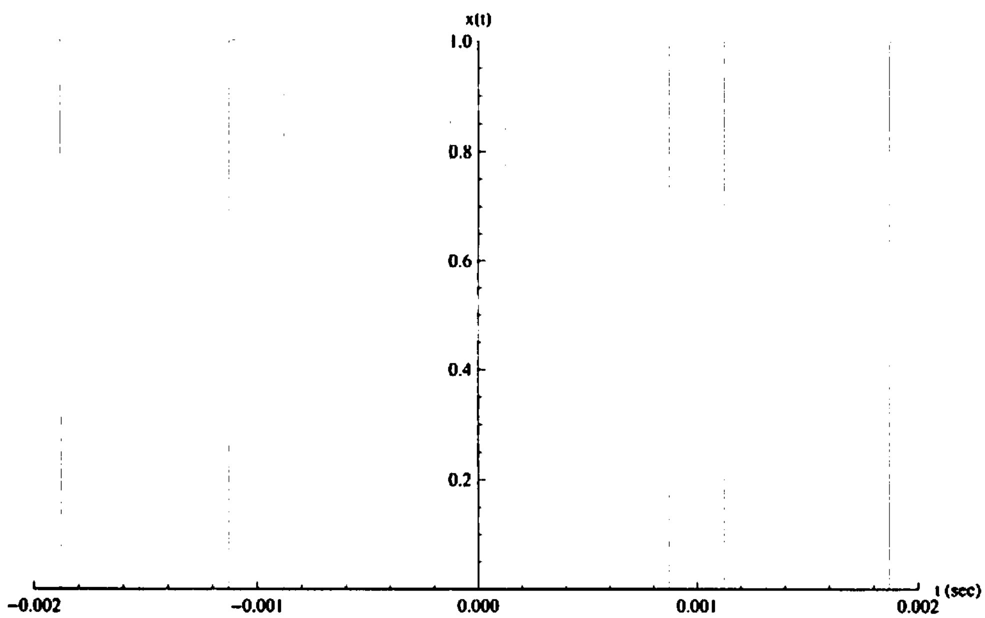
- **Load my DSP functions that I wrote for this course**

```
In[2]:= << dsp`
```

- **Plot the pulse train**

In[95]:=
```
delay = 0;
period = 1 * 10^-3;
range = 2 * 10^-3;
tao = .25 * 10^-3;
h = 1;
w0 = 2 Pi / period; (*rad/sec*)
f0 = 1 / period; (*hz*)
dutyCycle = tao / period;
numberOfCoeff = 20;
currentPulses = dsp`makePulseTrain[delay, period, range, tao, h];

Plot[0, {x, -range, range},
  PlotRange → {{-range, range}, {0, h}}, AxesLabel → {"t (sec)", "x(t)"},
  Epilog → {Thin, Red, currentPulses}
 ]
```

Out[105]=



- **Find the fourier series coefficients of the above pulse train**

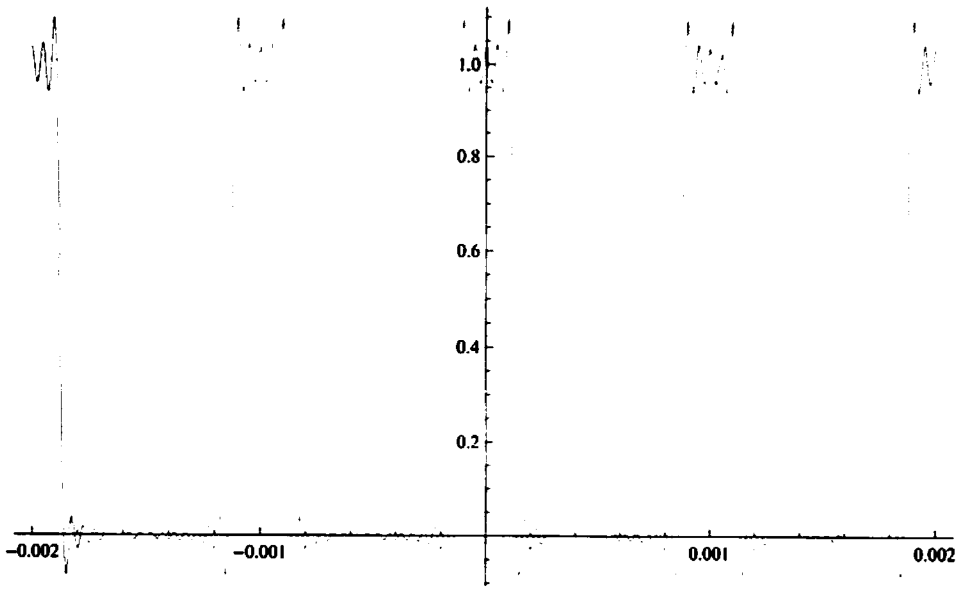
In[106]:=
```
xn = getFourierCoeffPulseTrain[h, tao, period, numberOfCoeff]
```

Out[106]= {0.25, 0.225079, 0.159155, 0.0750264, 9.74543 × 10^-18, -0.0450158, -0.0530516, -0.0321542, -9.74543 × 10^-18, 0.0250088, 0.031831, 0.0204617, 9.74543 × 10^-18, -0.0173138, -0.0227364, -0.0150053, -9.74543 × 10^-18, 0.0132399, 0.0176839, 0.0118463}

Plot fourier series approximation to the above pulse on top of it to compare

In[107]:=
```
Plot[getFourierApproximation[t, xn, period], {t, -range, range}]
```

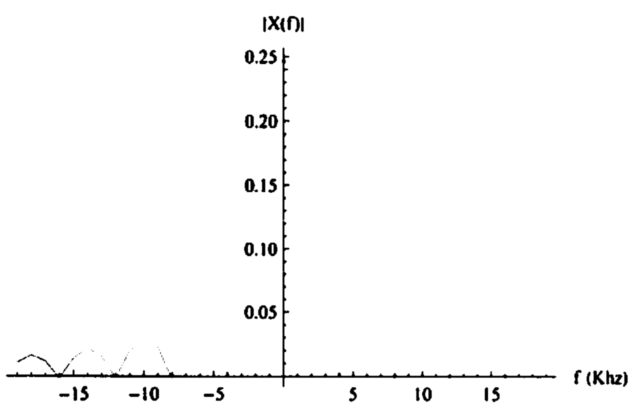
Out[107]=



- **Plot the spectrum of the pulse**

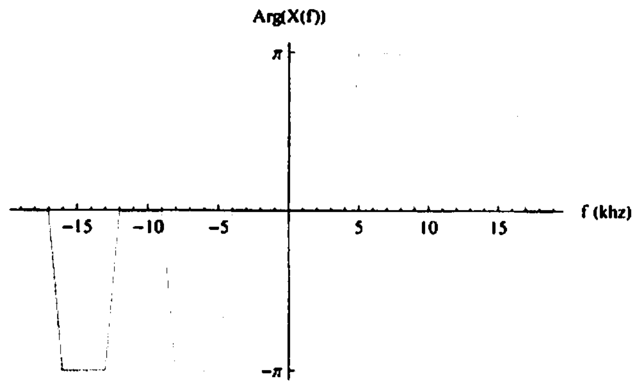
In[108]:=
```
data = getMagnitudeOfPulseTrainFourierCoeff [delay, period, range, dutyCycle, numberOfCoeff];
ListPlot[data, Joined → True, AxesLabel → {"f  (Khz)", "|X(f)|"}]
```

Out[109]=



In[110]:=
```
data = getPhaseOfPulseTrainFourierCoeff [delay, period, range, dutyCycle, numberOfCoeff];
ListPlot[data, Joined → True,
 AxesLabel → {"f (khz)", "Arg(X(f))"}, Ticks → {Automatic, {-Pi, Pi}}]
```

Out[111]=



- **Generate normalized low pass butterworth of order 4**

In[112]:=
```
Clear[s, form];
order = 4;
cuttoff = 1;
{poles, hs} = dsp`getButterworthPolynomial [order, cuttoff, s];
TraditionalForm@hs
```

Out[116]//TraditionalForm=

$$\frac{1.}{s^4 + 2.61313\, s^3 + 3.41421\, s^2 + 2.61313\, s + 1.}$$

- **convert the above to low pass butterworth with specified cuttoff**

In[117]:=
```
newHs = dsp`butterToLowPass [hs, 2 Pi/tao, s];
TraditionalForm@newHs
```

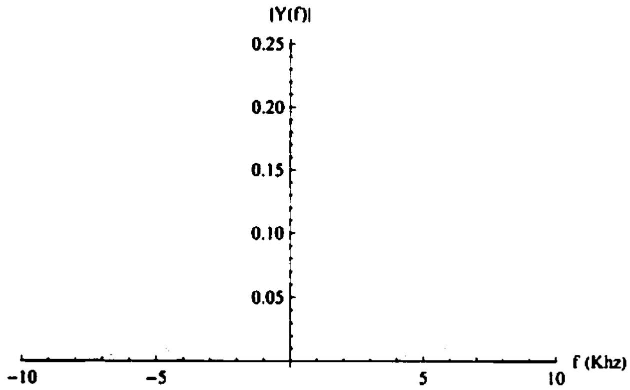
Out[118]//TraditionalForm=

$$\frac{1.}{2.50634\times 10^{-18}\, s^4 + 1.64604\times 10^{-13}\, s^3 + 5.40519\times 10^{-9}\, s^2 + 0.000103973\, s + 1.}$$

- **Multiply H(jwn) with Pulse fourier series y(n), and plot Y(f)**

In[119]:=
```
Clear[w];
xnFourier[n_] := h dutyCycle Sinc[Pi n dutyCycle]

tf[n_, w0_] := newHs /. s → (I w0 n)

yn[n_, w0_] := xnFourier[n] * tf[n, w0]

y[t_, w0_, numberOfCoeff_] := Sum[If[n == 0, yn[n, w0] * Exp[I w0 n t ],
    (yn[n, w0] * Exp[I w0 n t ] + yn[-n, w0] * Exp[-I w0 n t ])], {n, 0, numberOfCoeff}]

data = Table[{m, Abs[yn[m, w0]]}, {m, -20, 20}];
ListPlot[data, Joined → True,
 PlotRange → {{-10, 10}, All}, AxesLabel → {"f  (Khz)", "|Y(f)|"}]
```

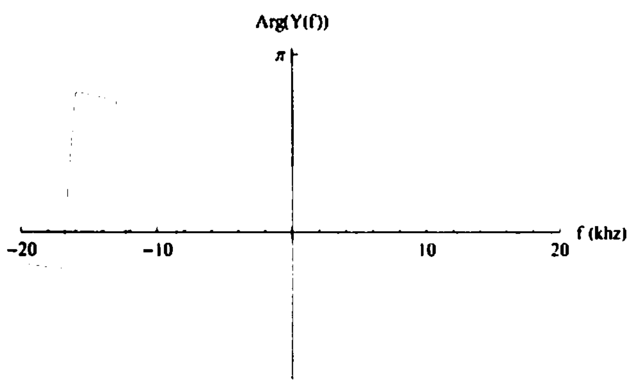
Out[125]=



- **Plot the phase spectrum**

In[126]:=
```
data = Table[{m, Arg[yn[m, w0]]}, {m, -20, 20}];
ListPlot[data, Joined → True, PlotRange → {{-20, 20}, All},
 AxesLabel → {"f  (khz)", "Arg(Y(f))"}, Ticks → {Automatic, {-Pi, Pi}}]
```

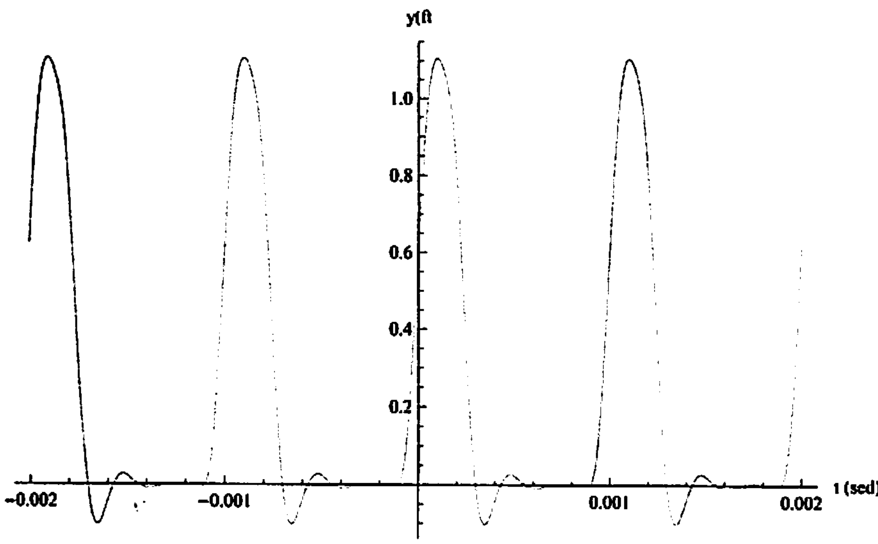
Out[127]=



- **Plot y(t)**

In[128]:=
```
Plot[y[t, w0, 10], {t, -range, range}, PlotRange → All, AxesLabel → {"t (sed)", "y(ft"}]
```

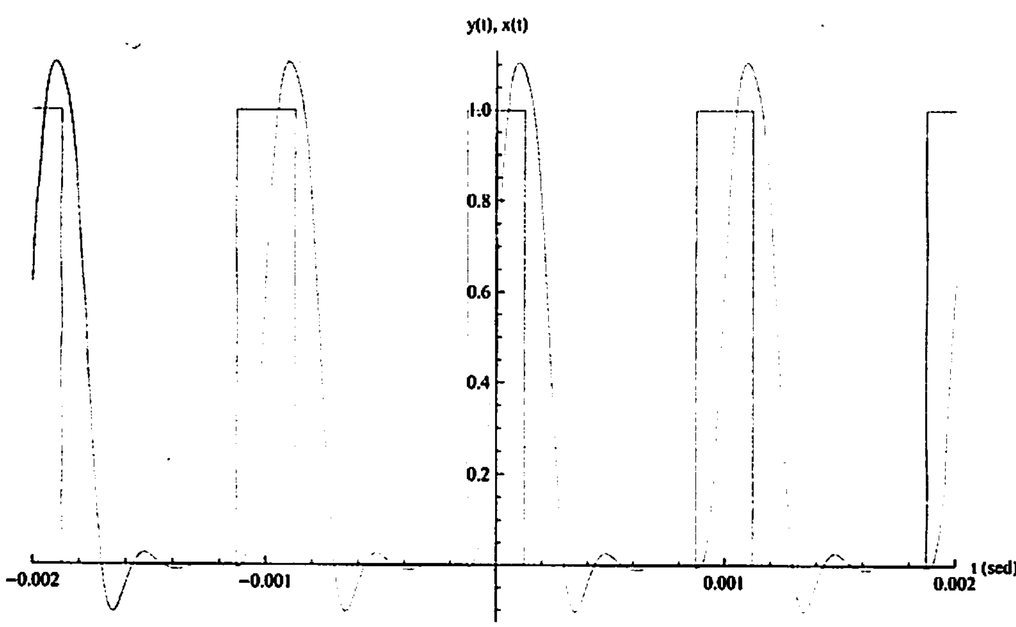
Out[128]=



- **Plot y(t) and x(t) on same plot to compare**

In[129]:=
```
Plot[y[t, w0, 10], {t, -range, range},
  PlotRange → {{-range, range}, All}, AxesLabel → {"t (sed)", "y(t), x(t)"},
    Epilog → {Thin, Red, currentPulses}
 ]
```

Out[129]=

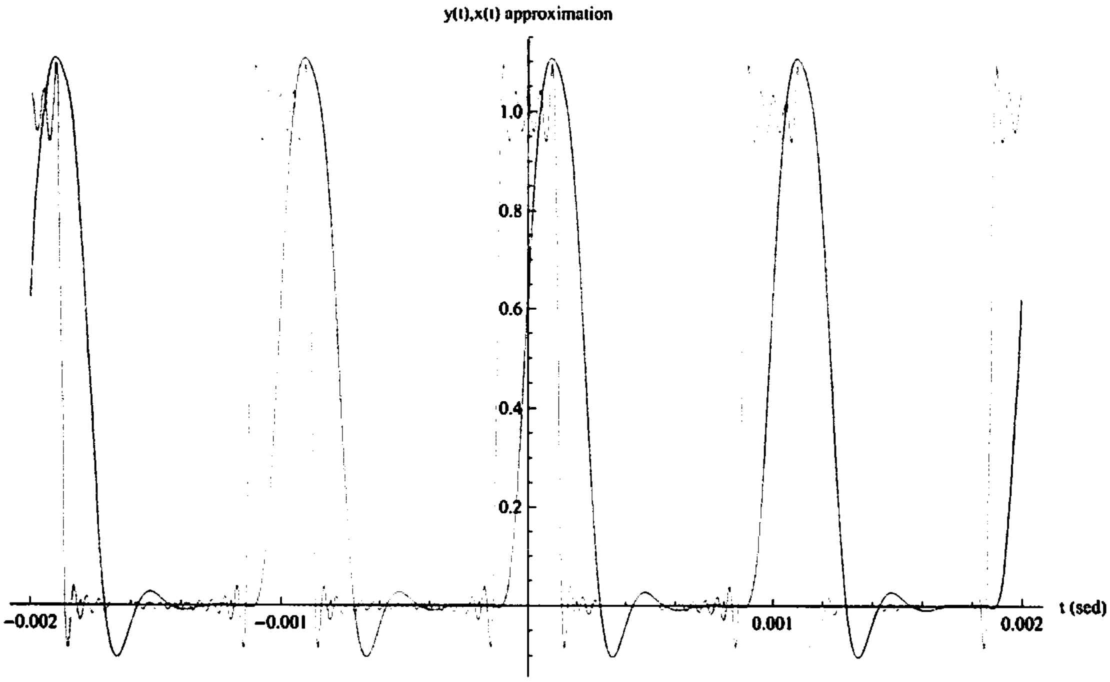


- **Plot y(t) on top of approximation of x(n) used**

In[50]:=

```
Plot[{getFourierApproximation[t, xn, period], y[t, w0, 10]}, {t, -range, range},
 PlotRange → All, AxesLabel → {"t (sed)", "y(t),x(t) approximation"}]
```

Out[50]=



Part (2) High Pass

- **convert normalized butterworth to high pass butterworth**

In[282]:=

```
newHs = dsp`butterToHighPass[hs, 2 Pi/tao, s];
newHs = Numerator[newHs] / Together[Denominator[newHs]];
TraditionalForm@newHs
```

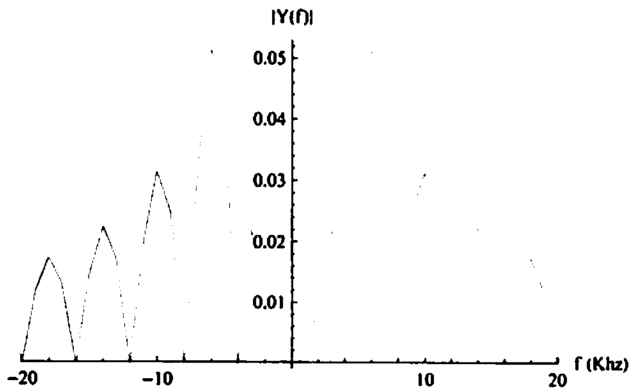
Out[284]//TraditionalForm=

$$\frac{1.s^4}{1.s^4 + 65675.s^3 + 2.1566\times 10^9 s^2 + 4.14839\times 10^{13} s + 3.98988\times 10^{17}}$$

- **Multiply H(jwn) with Pulse fourier series y(n), and plot Y(f)**

In[285]:=
```
Clear[w];
tf[n_, w0_] := newHs /. s → (I w0 n)

yn[n_, w0_] := dsp`fcPulseTrain[n, h, tao, period] * tf[n, w0]

y[t_, w0_, numberOfCoeff_] := Sum[If[n == 0, yn[n, w0] * Exp[I w0 n t],
    (yn[n, w0] * Exp[I w0 n t] + yn[-n, w0] * Exp[-I w0 n t])], {n, 0, numberOfCoeff}]

data = Table[{m, Abs[yn[m, w0]]}, {m, -40, 40}];
ListPlot[data, Joined → True,
 PlotRange → {{-20, 20}, All}, AxesLabel → {"f (Khz)", "|Y(f)|"}]
```

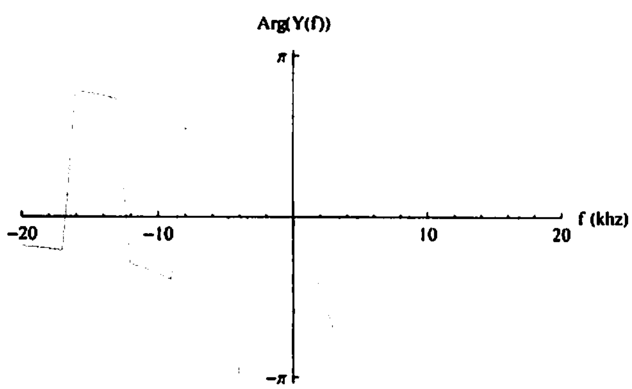
Out[290]=



- **Plot the phase spectrum**

In[227]:=
```
data = Table[{m, Arg[yn[m, w0]]}, {m, -20, 20}];
ListPlot[data, Joined → True, PlotRange → {{-20, 20}, All},
 AxesLabel → {"f (khz)", "Arg(Y(f))"}, Ticks → {Automatic, {-Pi, Pi}}]
```

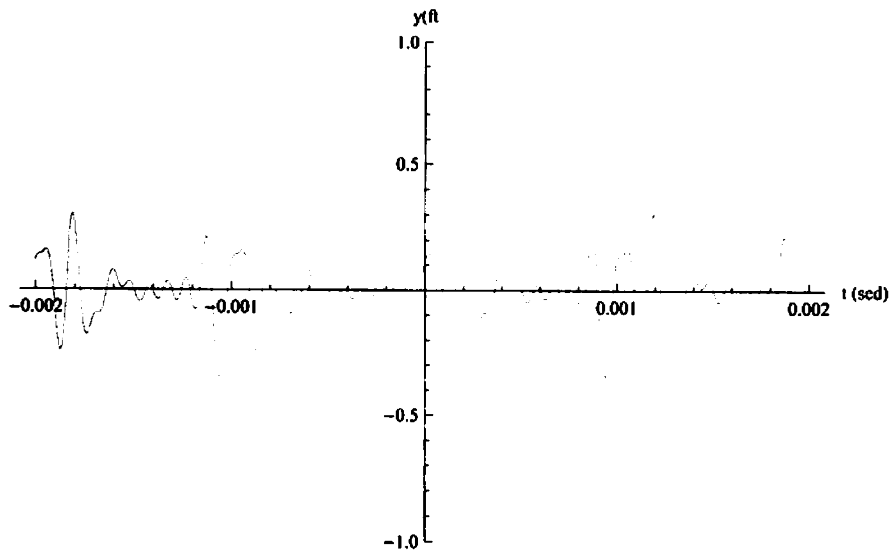
Out[228]=



- **Plot y(t)**

In[229]:=
```
Plot[y[t, w0, 10], {t, -range, range},
  PlotRange → {Automatic, {-h, h}}, AxesLabel → {"t (sed)", "y(ft"}]
```

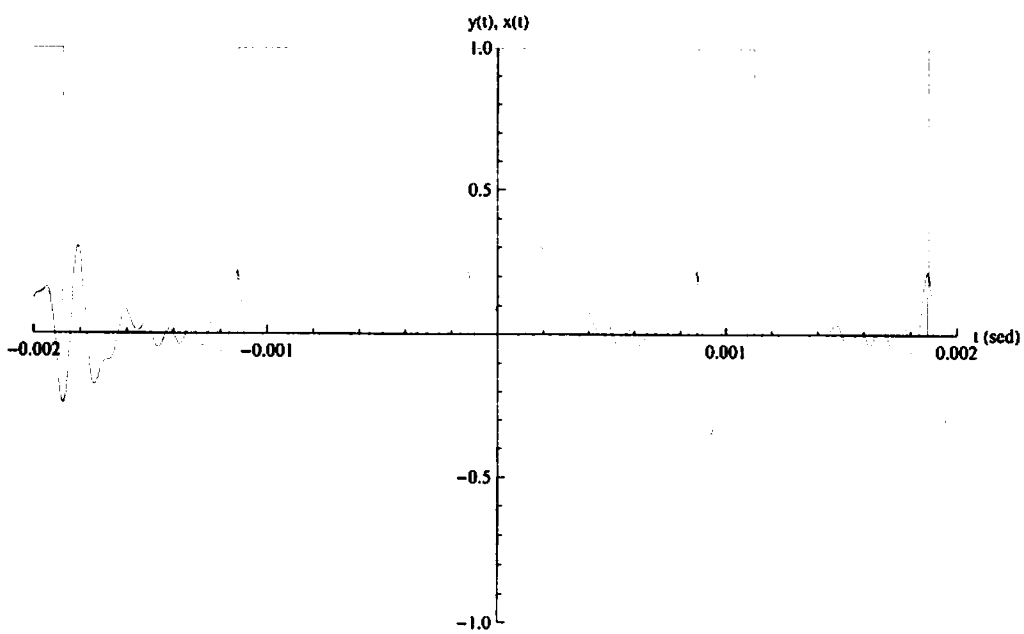
Out[229]=



- **Plot y(t) and x(t) on same plot to compare**

In[230]:=
```
Plot[y[t, w0, 10], {t, -range, range},
  PlotRange → {{-range, range}, {-h, h}}, AxesLabel → {"t (sed)", "y(t), x(t)"},
    Epilog → {Thin, Red, currentPulses}
 ]
```

Out[230]=

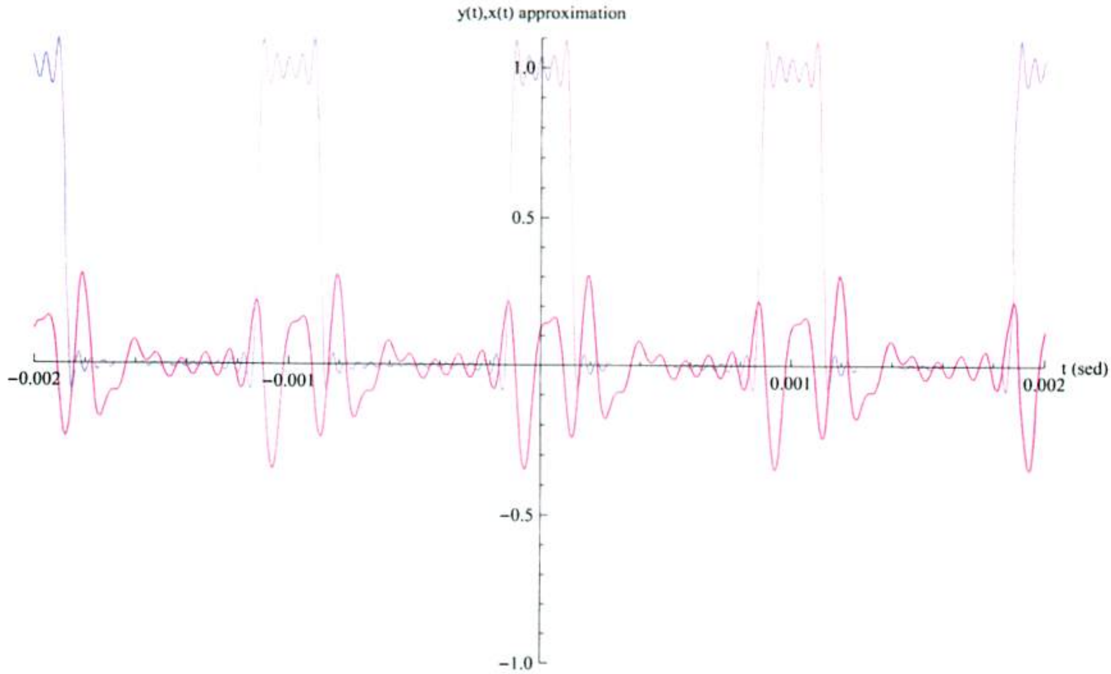


Plot y(t) on top of approximation of x(n) used

In[232]:=

```
Plot[{getFourierApproximation[t, xn, period], y[t, w0, 10]},
 {t, -range, range}, PlotRange → {{-range, range}, {-h, 1.1 h}},
 AxesLabel → {"t (sed)", "y(t),x(t) approximation"}]
```

Out[232]=



Part (3) BandPass filter

- **convert normalized butterworth to band pass butterworth**

In[273]:=

```
newHs = dsp`butterToBandPass[hs, 2 Pi/tao, 4 Pi/tao, s];
newHs = Numerator[newHs] / Together[Denominator[newHs]];
TraditionalForm@newHs
```

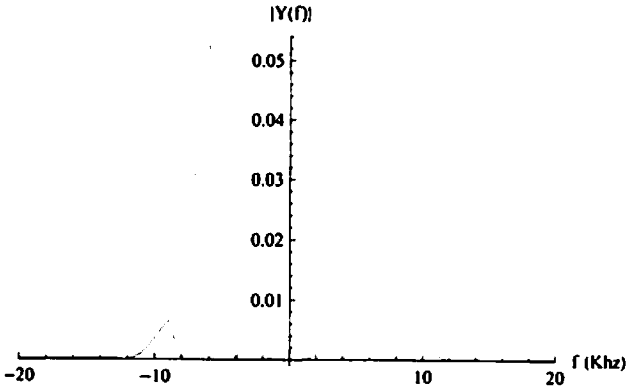
Out[275]//TraditionalForm=

$$(1.\,s^4)/(2.50634\times10^{-18}\,s^8+1.64604\times10^{-13}\,s^7+1.80703\times10^{-8}\,s^6+0.000727811\,s^5+38.6569\,s^4+919450.\,s^3+2.88394\times10^{10}\,s^2+3.31871\times10^{14}\,s+6.3838\times10^{18})$$

- **Multiply H(jwn) with Pulse fourier series y(n), and plot Y(f)**

In[276]:=
```
Clear[w];
tf[n_, w0_] := newHs /. s → (I w0 n)

yn[n_, w0_] := dsp`fcPulseTrain[n, h, tao, period] * tf[n, w0]

y[t_, w0_, numberOfCoeff_] := Sum[If[n == 0, yn[n, w0] * Exp[I w0 n t],
    (yn[n, w0] * Exp[I w0 n t] + yn[-n, w0] * Exp[-I w0 n t])], {n, 0, numberOfCoeff}]

data = Table[{m, Abs[yn[m, w0]]}, {m, -40, 40}];
ListPlot[data, Joined → True,
 PlotRange → {{-20, 20}, All}, AxesLabel → {"f  (Khz)", "|Y(f)|"}]
```

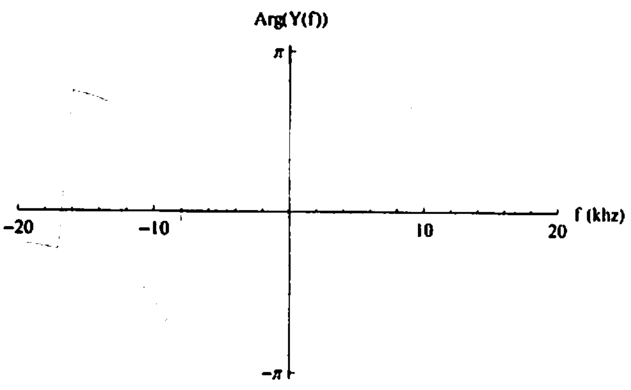
Out[281]=



- **Plot the phase spectrum**

In[242]:=
```
data = Table[{m, Arg[yn[m, w0]]}, {m, -20, 20}];
ListPlot[data, Joined → True, PlotRange → {{-20, 20}, All},
 AxesLabel → {"f  (khz)", "Arg(Y(f))"}, Ticks → {Automatic, {-Pi, Pi}}]
```

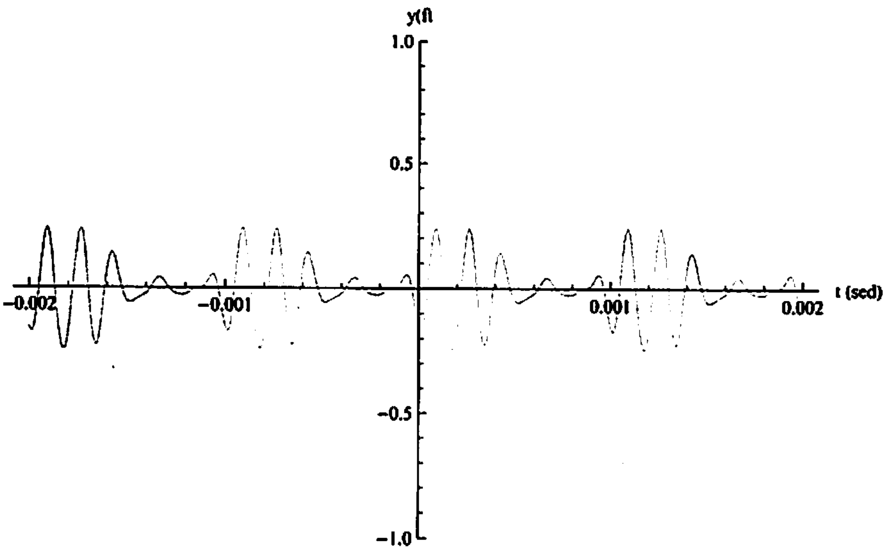
Out[243]=



▪ Plot y(t)

```
In[244]:= Plot[y[t, w0, 10], {t, -range, range},
    PlotRange → {Automatic, {-h, h}}, AxesLabel → {"t (sed)", "y(ft")}]
```

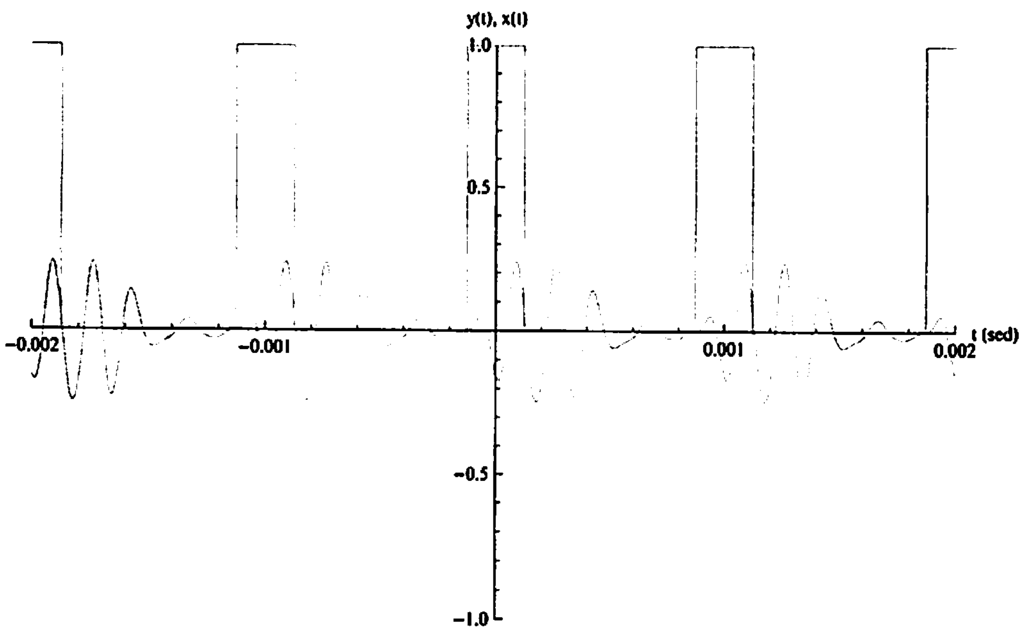
Out[244]=



▪ Plot y(t) and x(t) on same plot to compare

```
In[245]:= Plot[y[t, w0, 10], {t, -range, range},
    PlotRange → {{-range, range}, {-h, h}}, AxesLabel → {"t (sed)", "y(t), x(t)"},
      Epilog → {Thin, Red, currentPulses}
    ]
```

Out[245]=

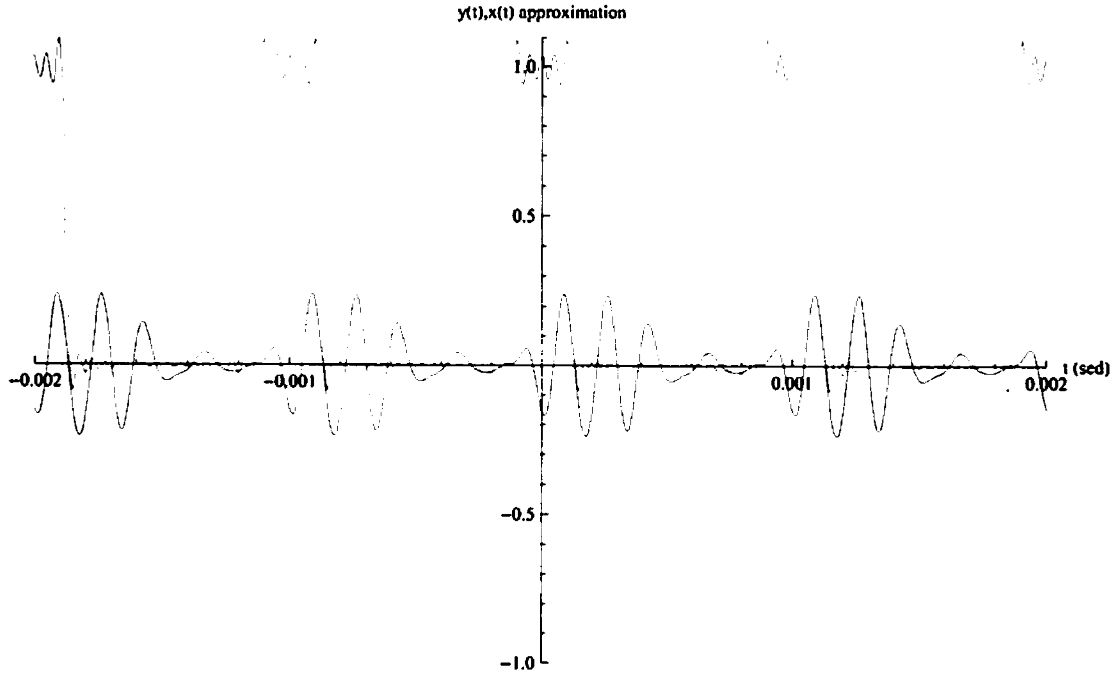


Plot y(t) on top of approximation of x(n) used

In[246]:=

```
Plot[{getFourierApproximation[t, xn, period], y[t, w0, 10]},
 {t, -range, range}, PlotRange → {{-range, range}, {-h, 1.1 h}},
 AxesLabel → {"t (sed)", "y(t),x(t) approximation"}]
```

Out[246]=



Part (4) BandStop filter

- **convert normalized butterworth to band stop butterworth**

In[293]:=

```
newHs = dsp`butterToBandStop[hs, 2 Pi/tao, 4 Pi/tao, s];
newHs = Numerator[newHs] / Together[Denominator[newHs]];
TraditionalForm@newHs
```

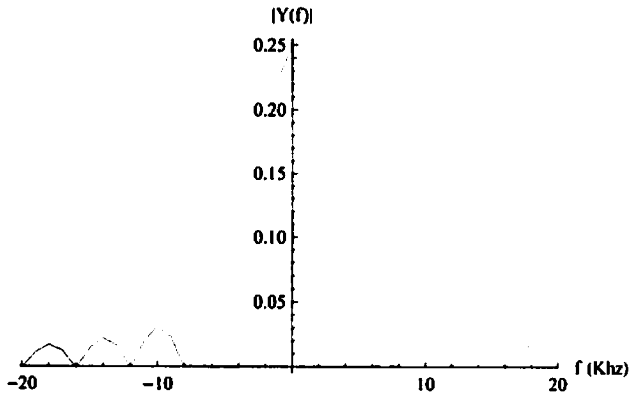
Out[295]//TraditionalForm=

$$(1.\,(s^2 + 1.26331\times 10^9)^4)/(1.\,s^8 + 65675.\,s^7 + 7.20984\times 10^9 s^6 + 2.90388\times 10^{14} s^5 + 1.54236\times 10^{19} s^4 + 3.66849\times 10^{23} s^3 + 1.15066\times 10^{28} s^2 + 1.32413\times 10^{32} s + 2.54706\times 10^{36})$$

- **Multiply H(jwn) with Pulse fourier series y(n), and plot Y(f)**

In[296]:=
```
Clear[w];
tf[n_, w0_] := newHs /. s → (I w0 n)

yn[n_, w0_] := dsp`fcPulseTrain[n, h, tao, period] * tf[n, w0]

y[t_, w0_, numberOfCoeff_] := Sum[If[n == 0, yn[n, w0] * Exp[I w0 n t ],
    (yn[n, w0] * Exp[I w0 n t ] + yn[-n, w0] * Exp[-I w0 n t ])], {n, 0, numberOfCoeff}]

data = Table[{m, Abs[yn[m, w0]]}, {m, -40, 40}];
ListPlot[data, Joined → True,
 PlotRange → {{-20, 20}, All}, AxesLabel → {"f  (Khz)", "|Y(f)|"}]
```

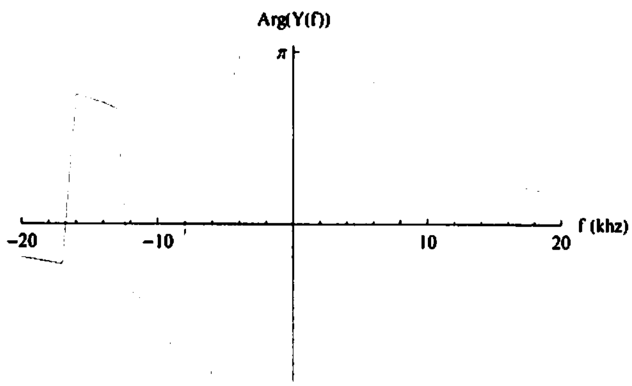
Out[301]=



- **Plot the phase spectrum**

In[302]:=
```
data = Table[{m, Arg[yn[m, w0]]}, {m, -20, 20}];
ListPlot[data, Joined → True, PlotRange → {{-20, 20}, All},
 AxesLabel → {"f  (khz)", "Arg(Y(f))"}, Ticks → {Automatic, {-Pi, Pi}}]
```

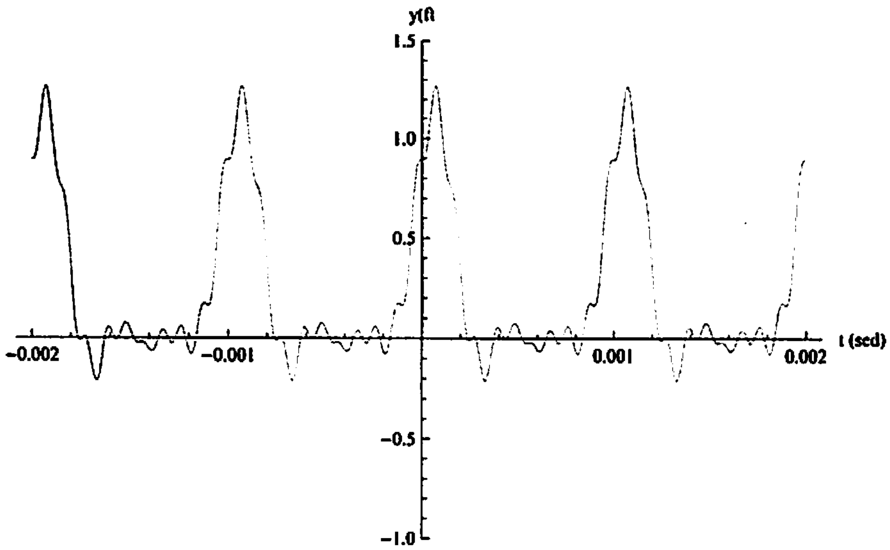
Out[303]=



- **Plot y(t)**

In[306]:=
```
Plot[y[t, w0, 10], {t, -range, range},
  PlotRange → {Automatic, {-h, 1.5 h}}, AxesLabel → {"t (sed)", "y(ft")}]
```

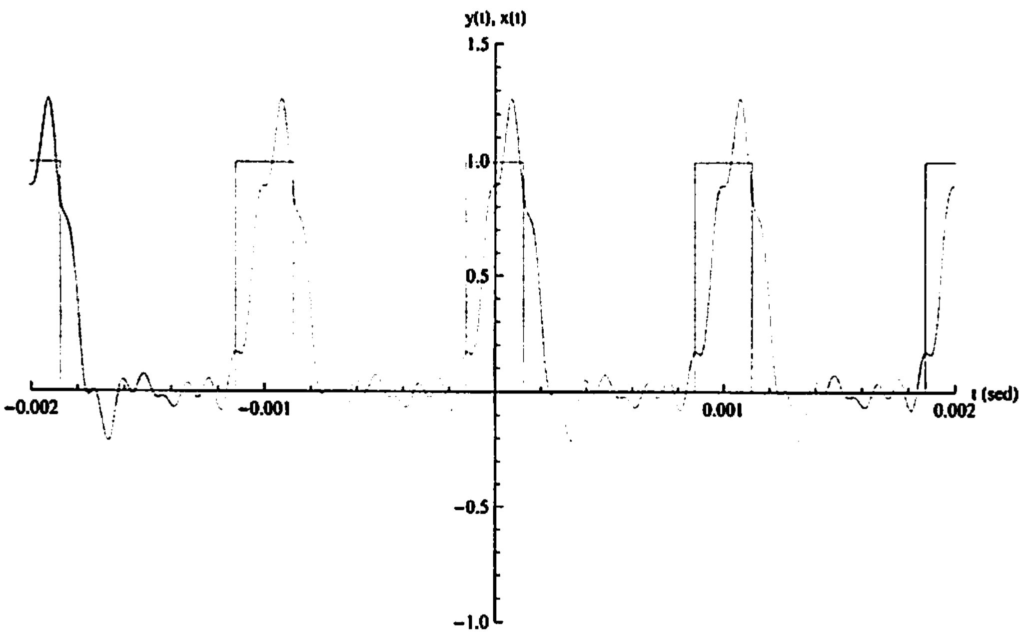
Out[306]=



- **Plot y(t) and x(t) on same plot to compare**

In[307]:=
```
Plot[y[t, w0, 10], {t, -range, range},
  PlotRange → {{-range, range}, {-h, 1.5 h}}, AxesLabel → {"t (sed)", "y(t), x(t)"},
    Epilog → {Thin, Red, currentPulses}
 ]
```

Out[307]=

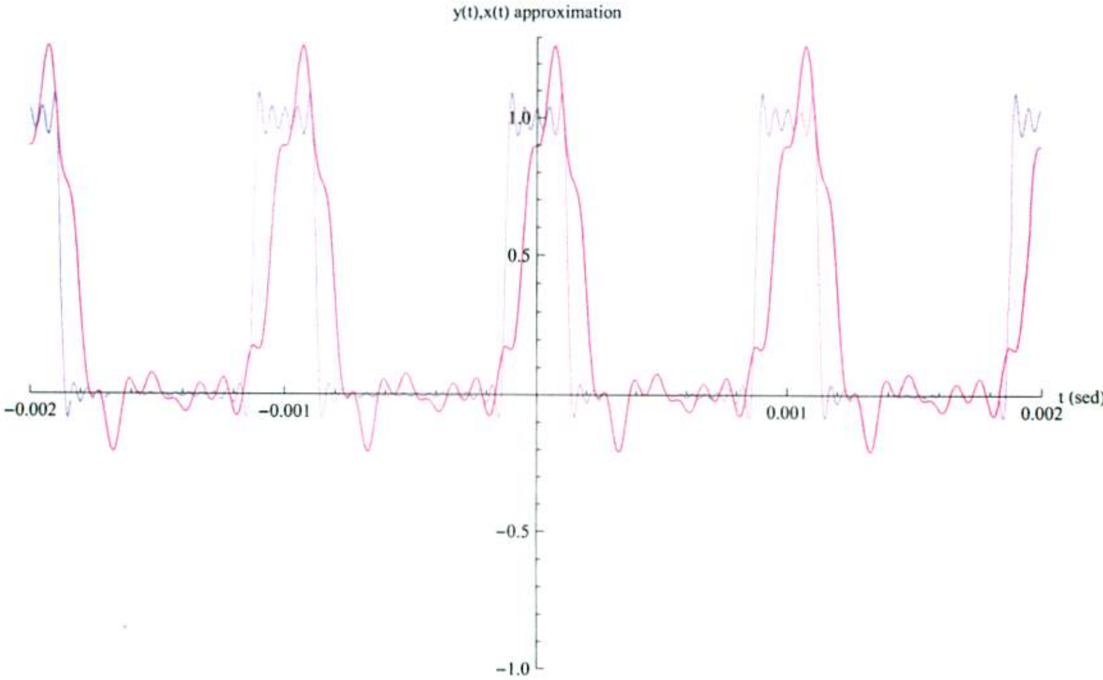


Plot y(t) on top of approximation of x(n) used

In[308]:=

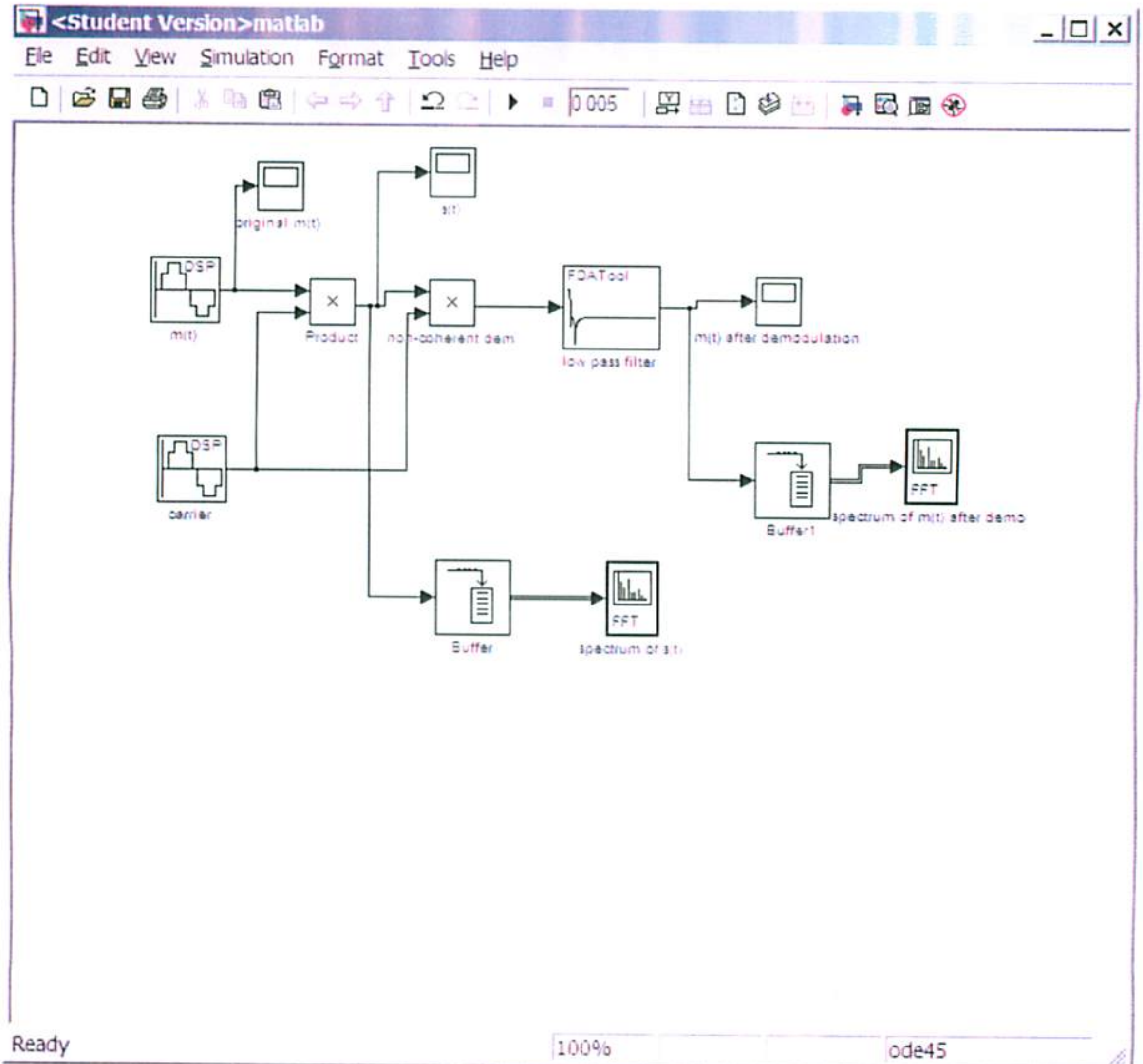
```
Plot[{getFourierApproximation[t, xn, period], y[t, w0, 10]},
 {t, -range, range}, PlotRange → {{-range, range}, {-h, 1.3 h}},
 AxesLabel → {"t (sed)", "y(t),x(t) approximation"}]
```

Out[308]=



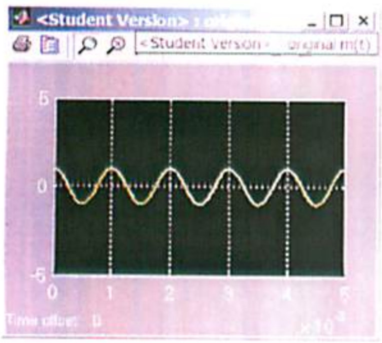
Computer Assignment #2
by Nasser M. Abbasi
ECE 405, summer session 1, Cal Poly Pomona

Simulink setup

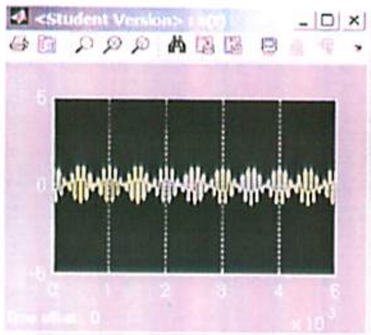


Part (1) *non-coherent demodulation*

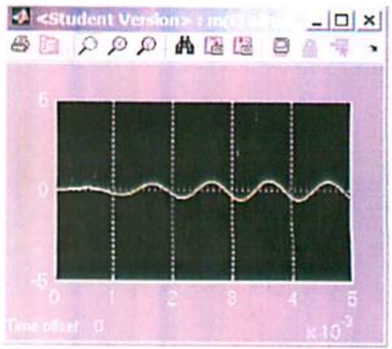
After run of the simulation, the following are the outputs time scope:



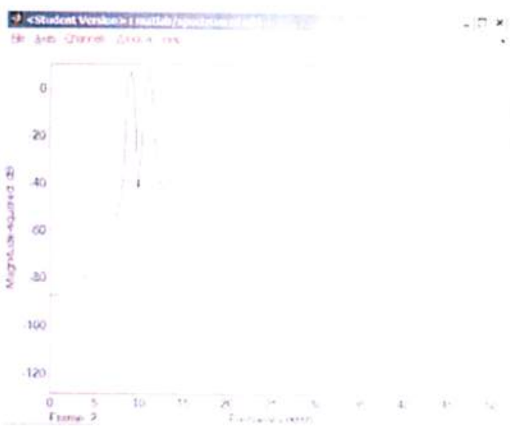
Original m(t)



S(t)=m(t)*cos(2 pi fc t)



Demodulated m(t)

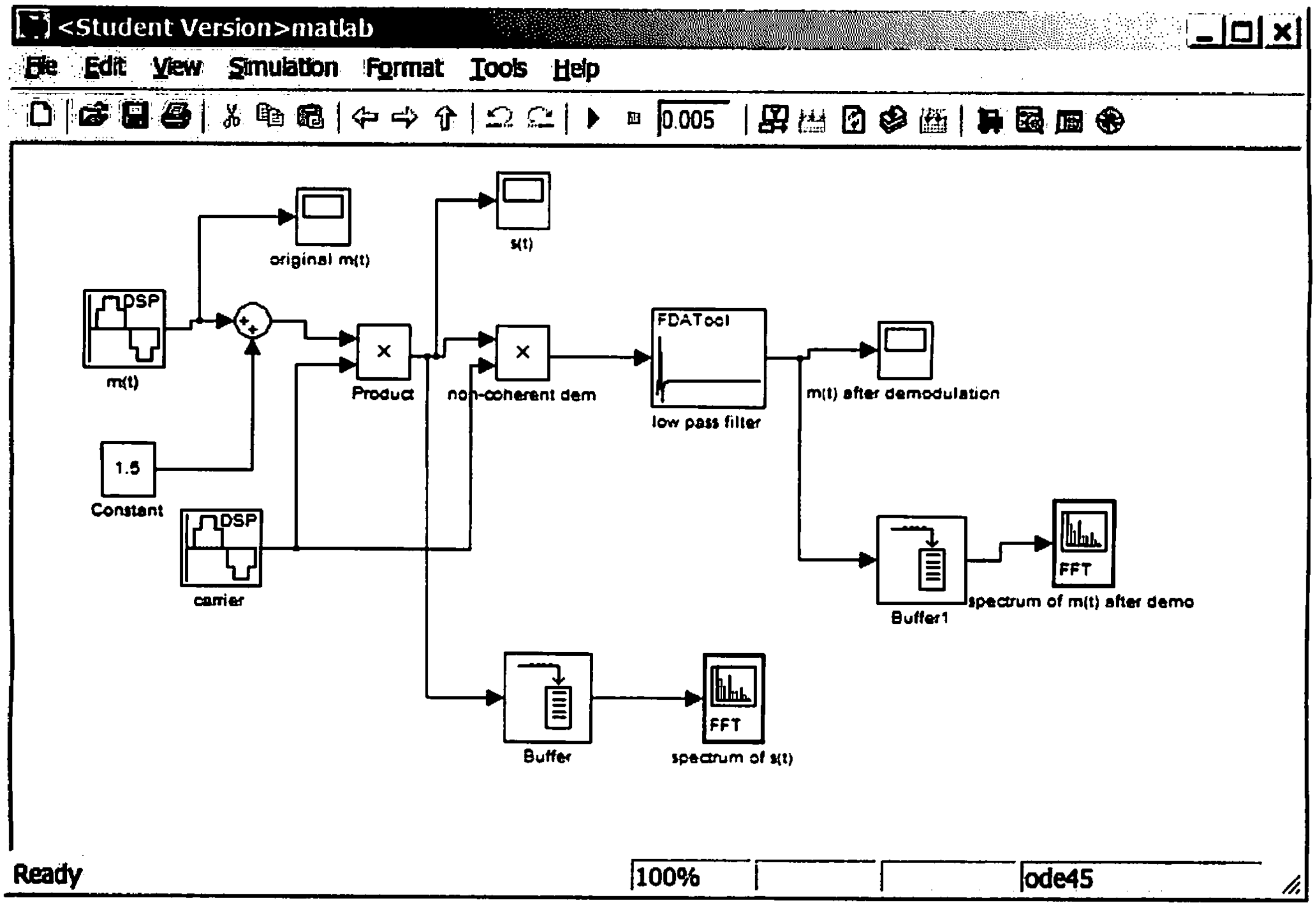


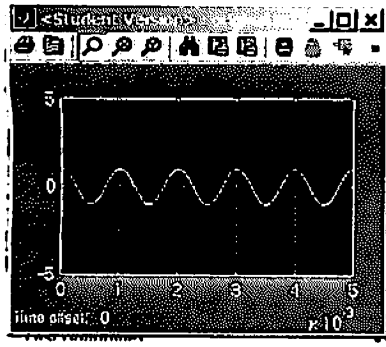
Magnitude Spectrum of s(t)

Magnitude Spectrum of m(t) after demodulation, notice fm=1000 hz

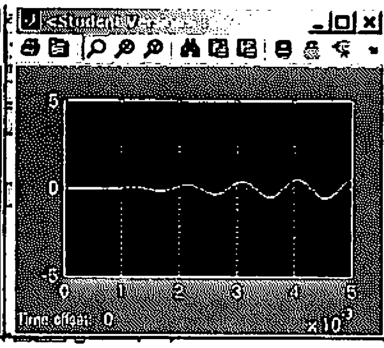
Computer Assignment #3
by Nasser M. Abbasi
ECE 405, summer session 1, Cal Poly Pomona

Simulink setup

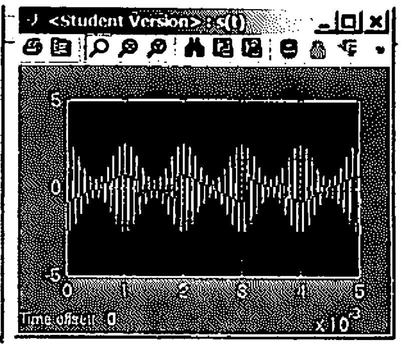




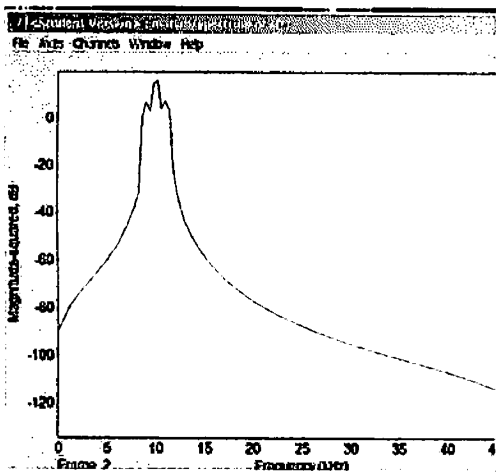
Original m(t)



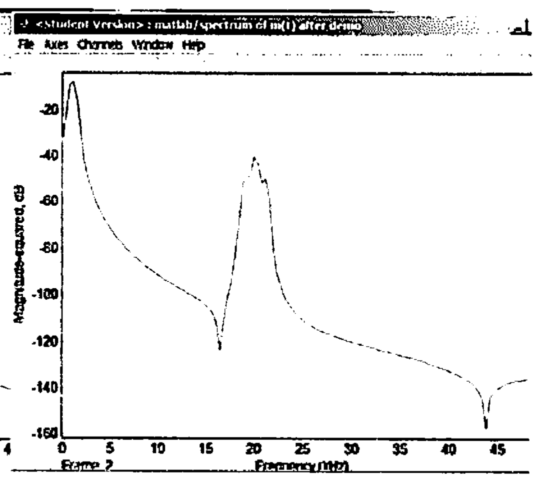
Demodulated m(t)



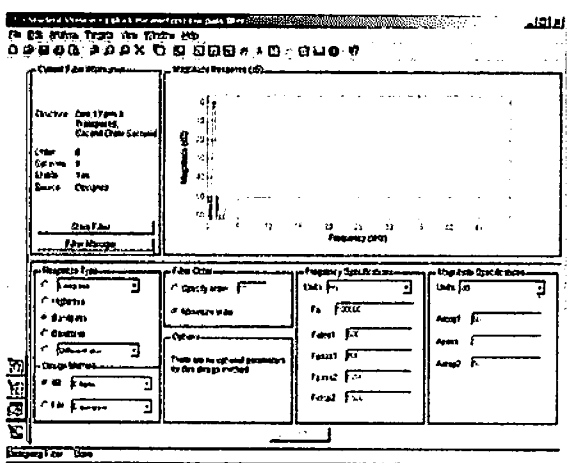
S(t)=(Ac+cos(2 pi fm t))*cos(2 pi fc t)



Magnitude Spectrum of s(t)



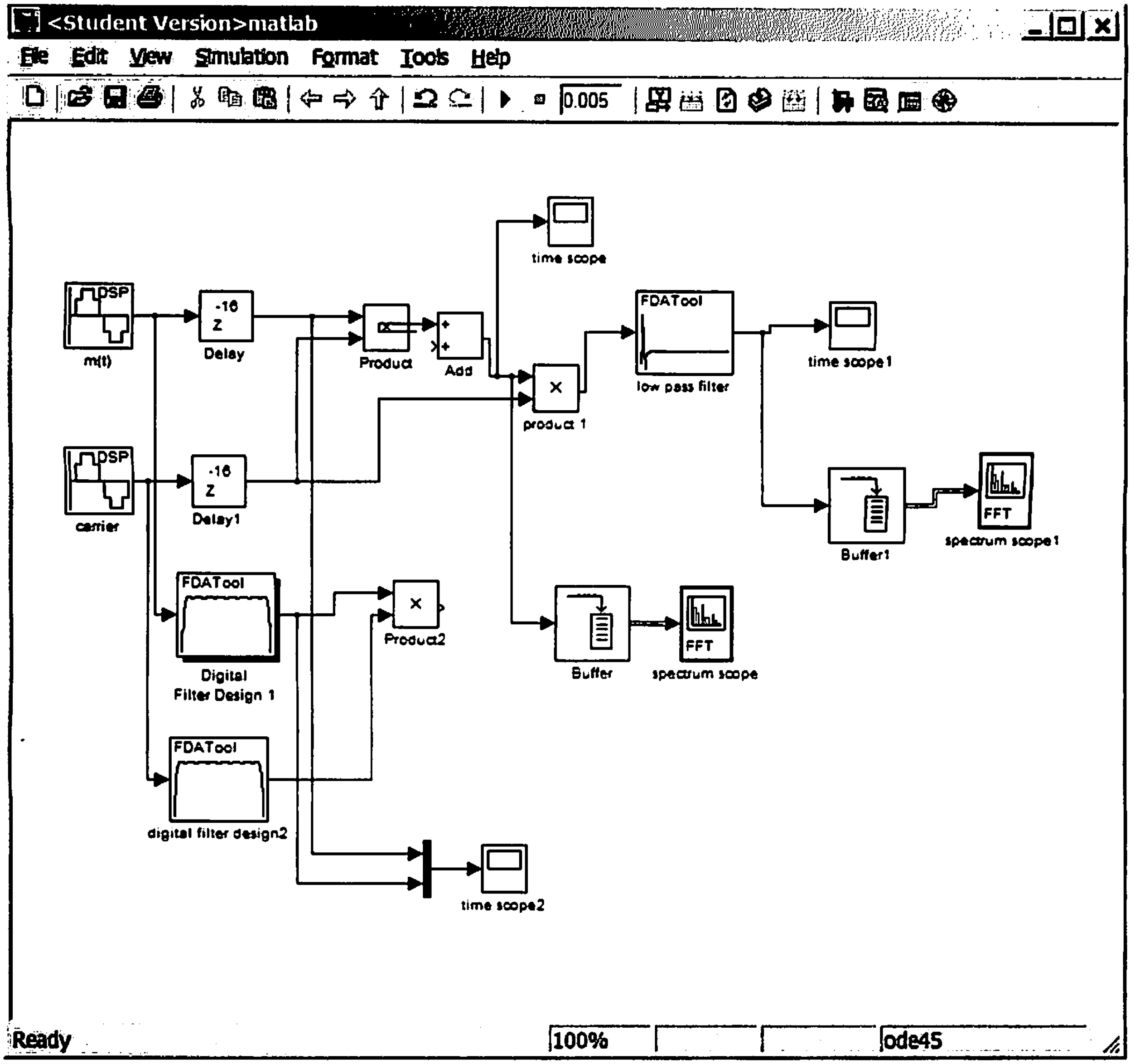
Magnitude Spectrum of m(t) after demodulation, notice fm=1000 hz

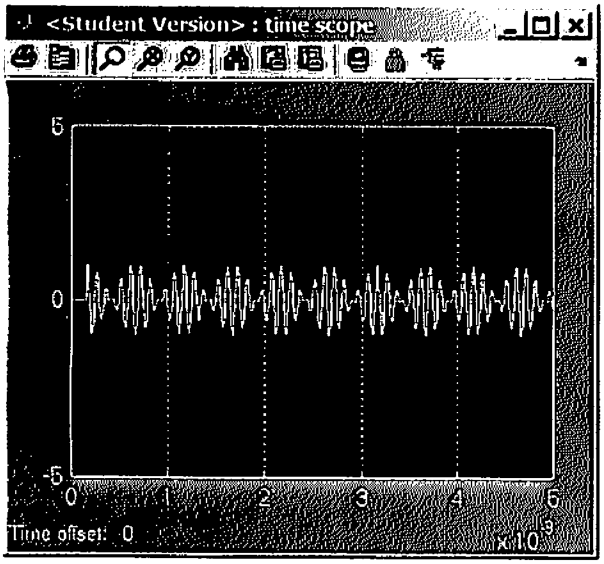


Computer #3
Part (1)

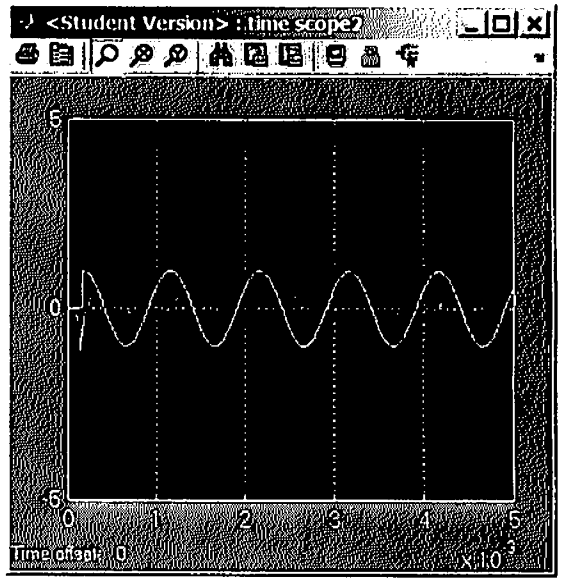
Computer Assignment #4
by Nasser M. Abbasi
ECE 405, summer session 1, Cal Poly Pomona

Simulink setup

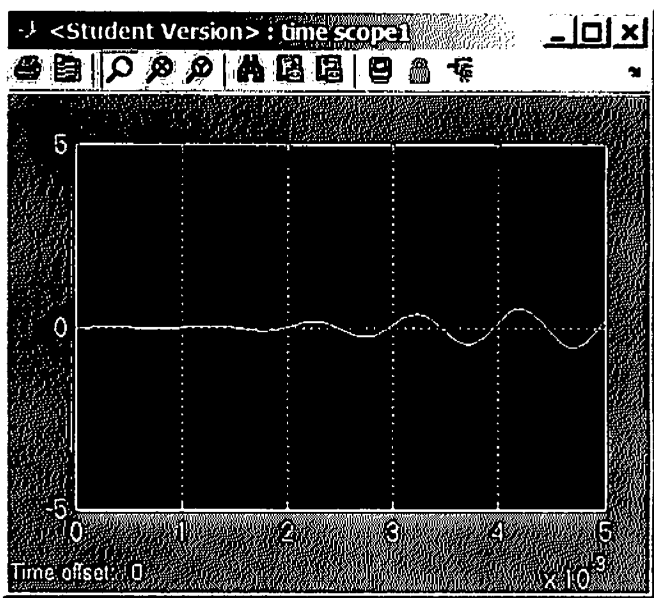




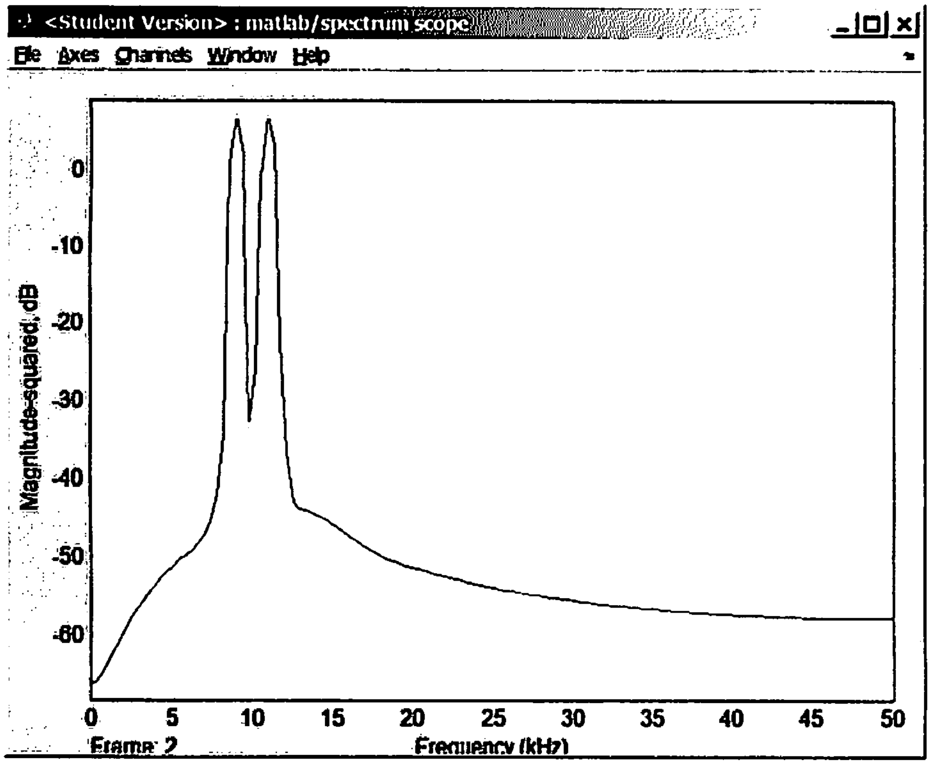
Time scope output



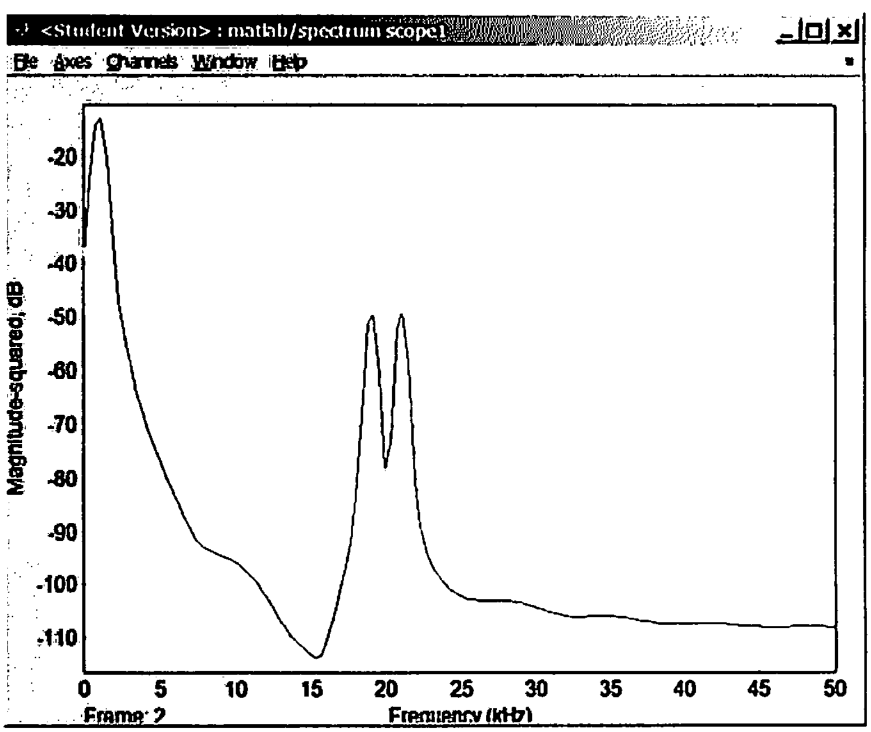
Time scope 2 output



Time scope 1 output



Spectrum scope output

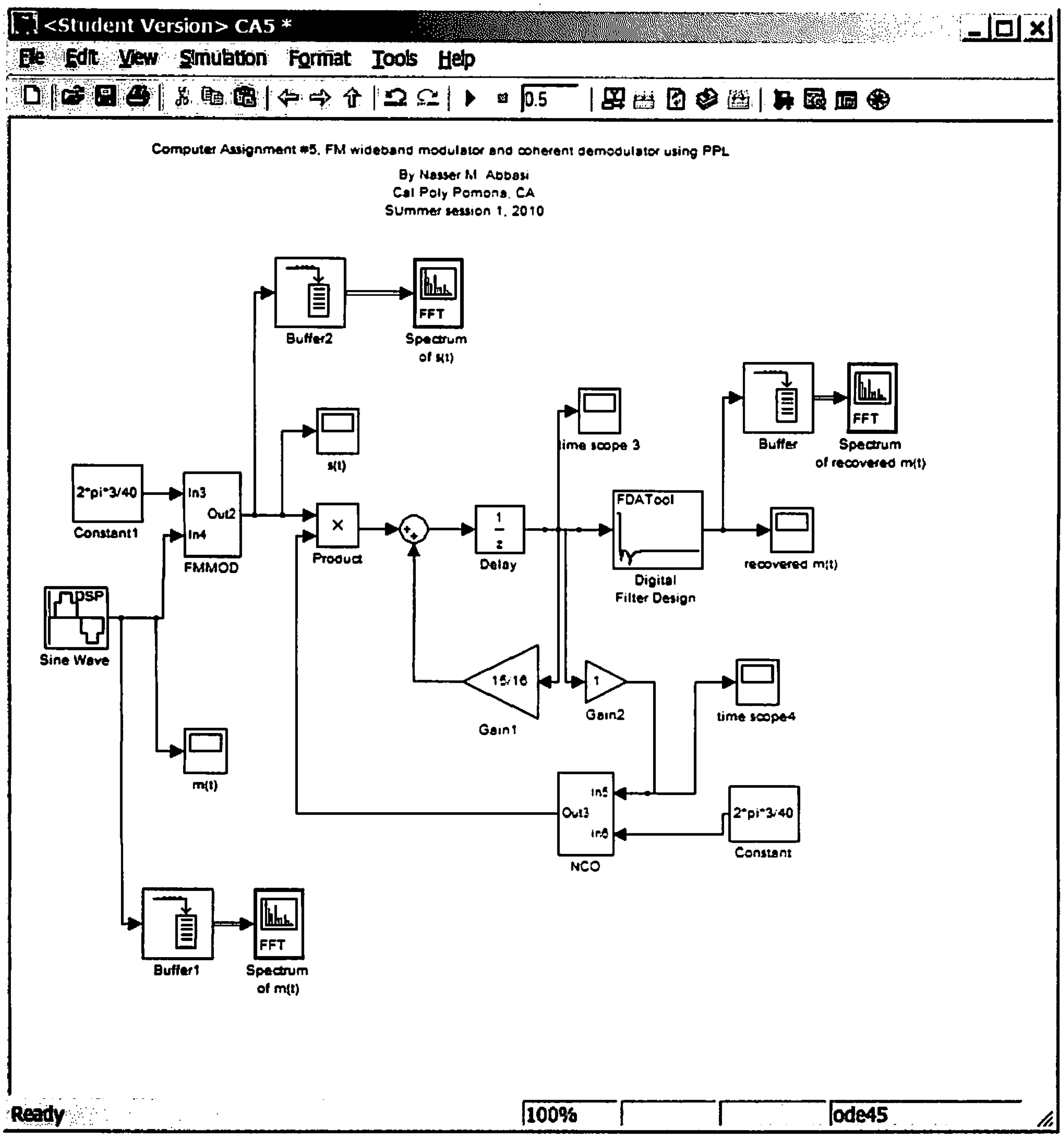


Spectrum scope 1 output

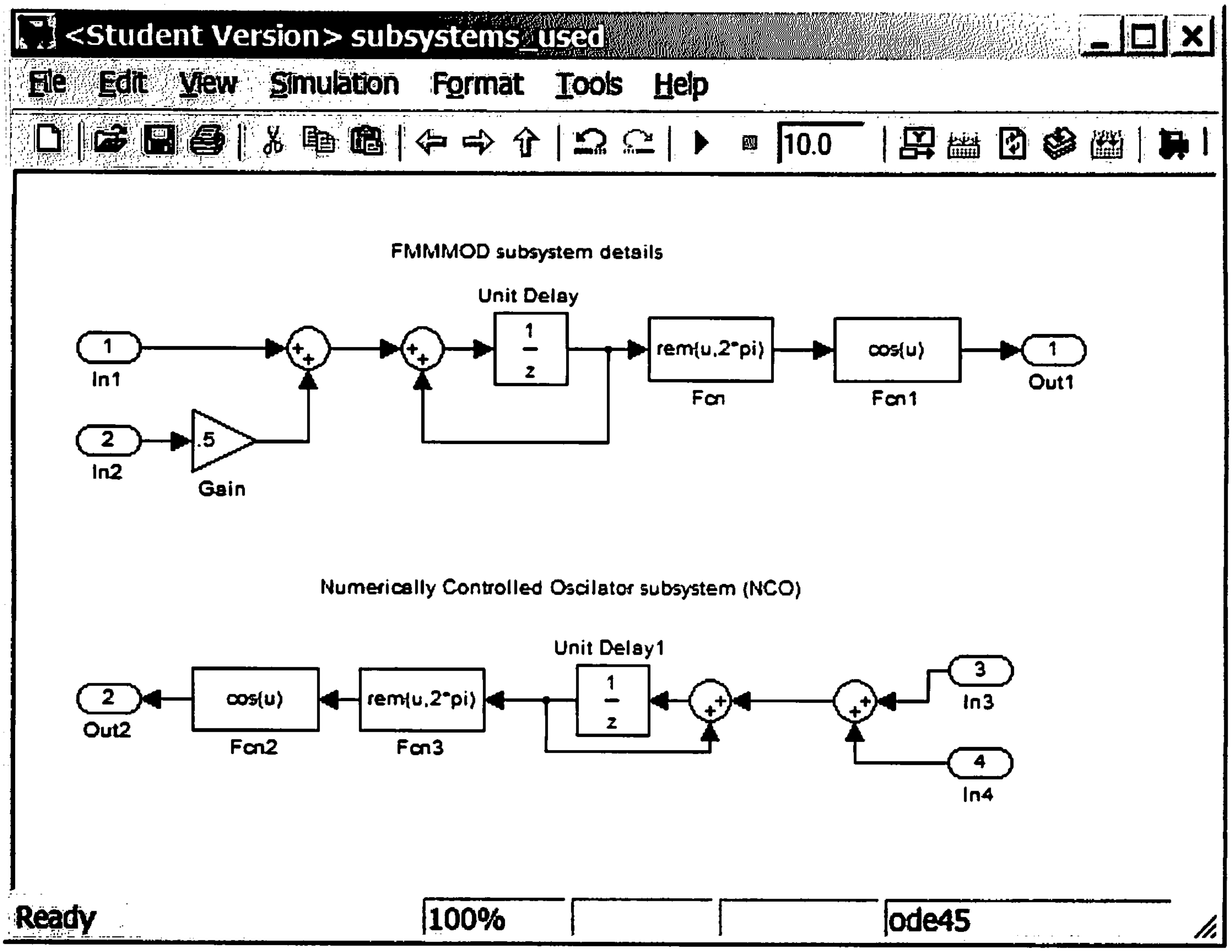
Computer Assignment #5, FM wideband modulator and coherent demodulator using PPL
Nasser M. Abbasi
Cal Poly Pomona, CA
SUmmer session 1, 2010

Simulink model

model is here

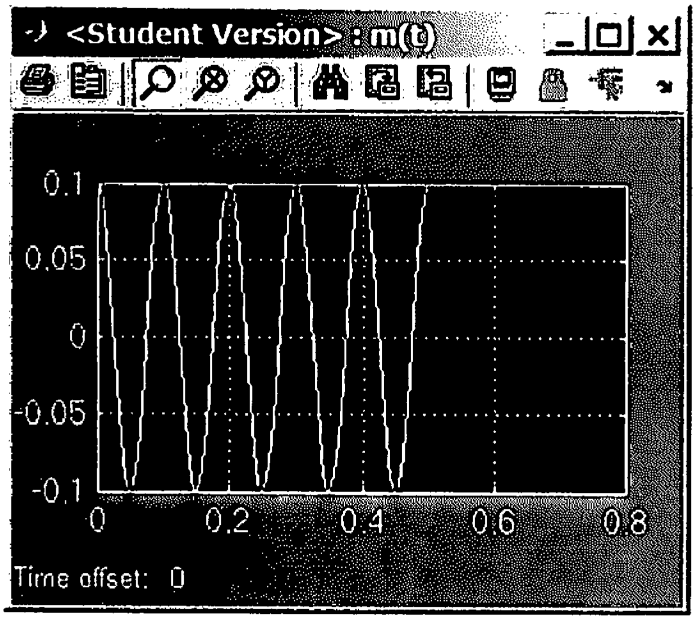


Models of subsystems

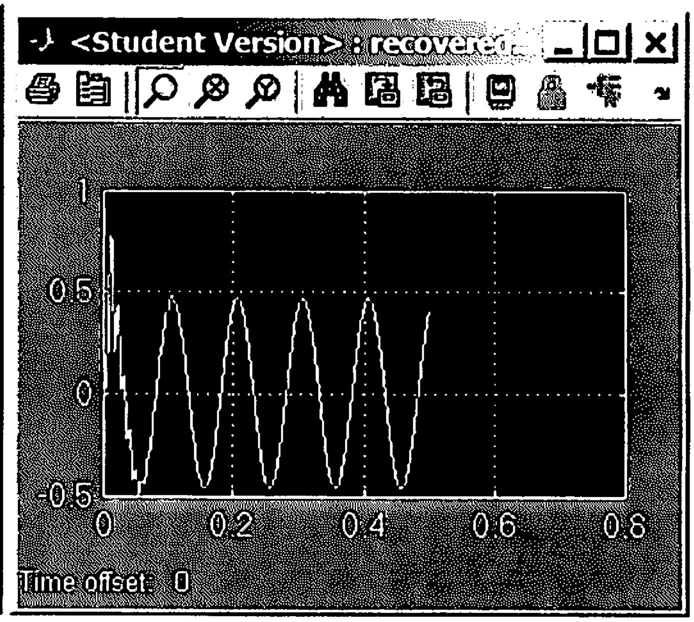


model file of the above is <u>here</u>

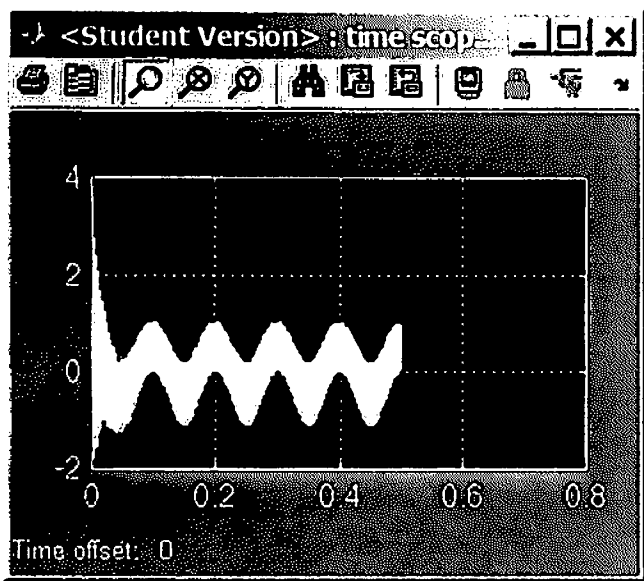
output



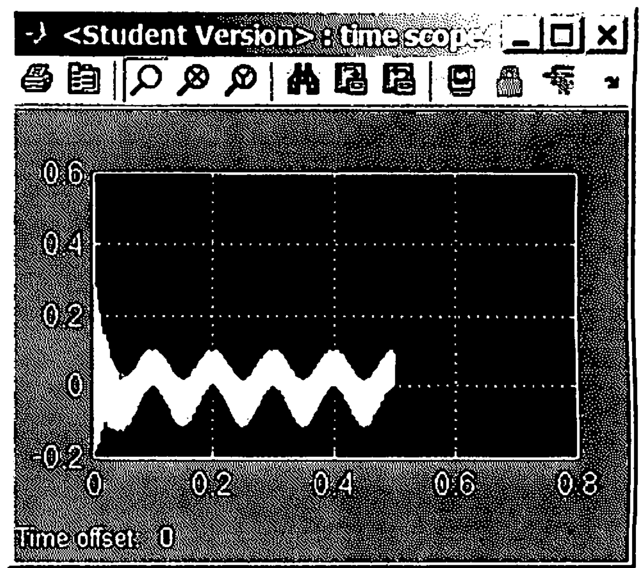
Original message m(t)



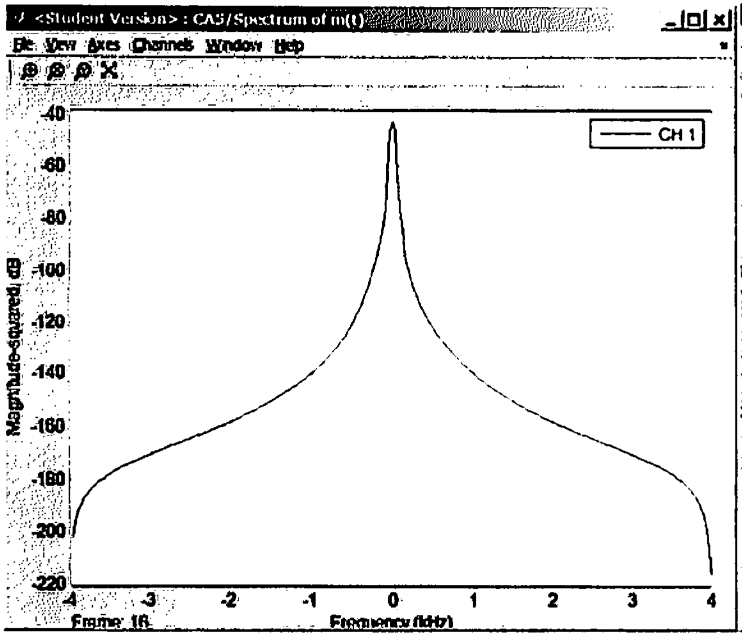
Recovered (demodulated) message m(t)



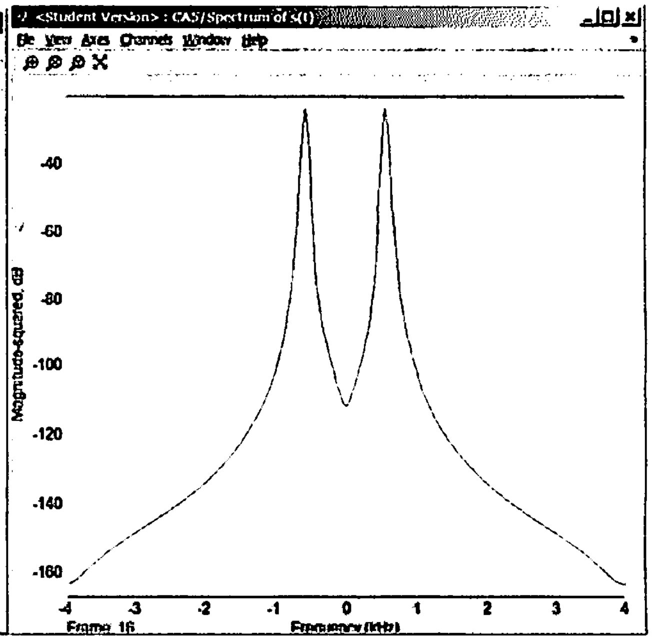
Time scope (3) in the model



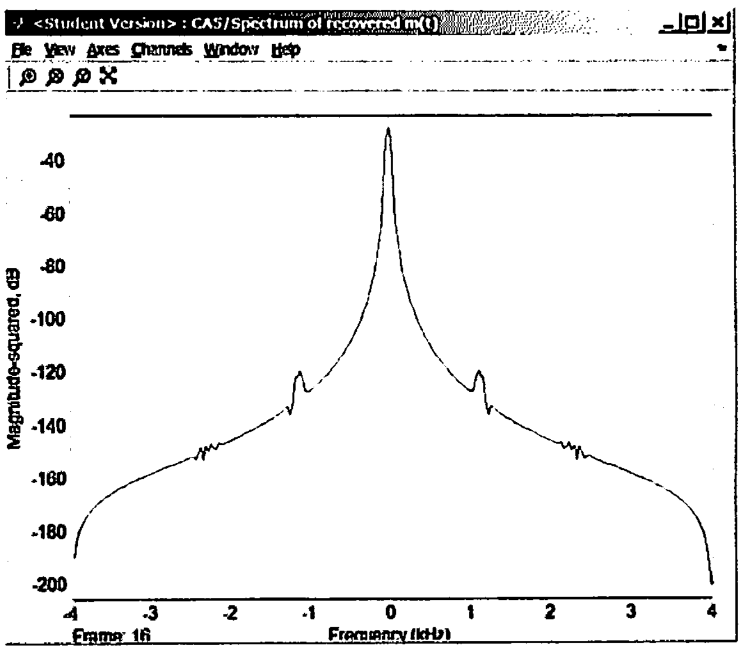
Time scope (4) in the model



Spectrum of original message m(t)



Spectrum of s(t), the modulated carrier (FM)



Spectrum of recovered message m(t) (compare to spectrum of original m(t))

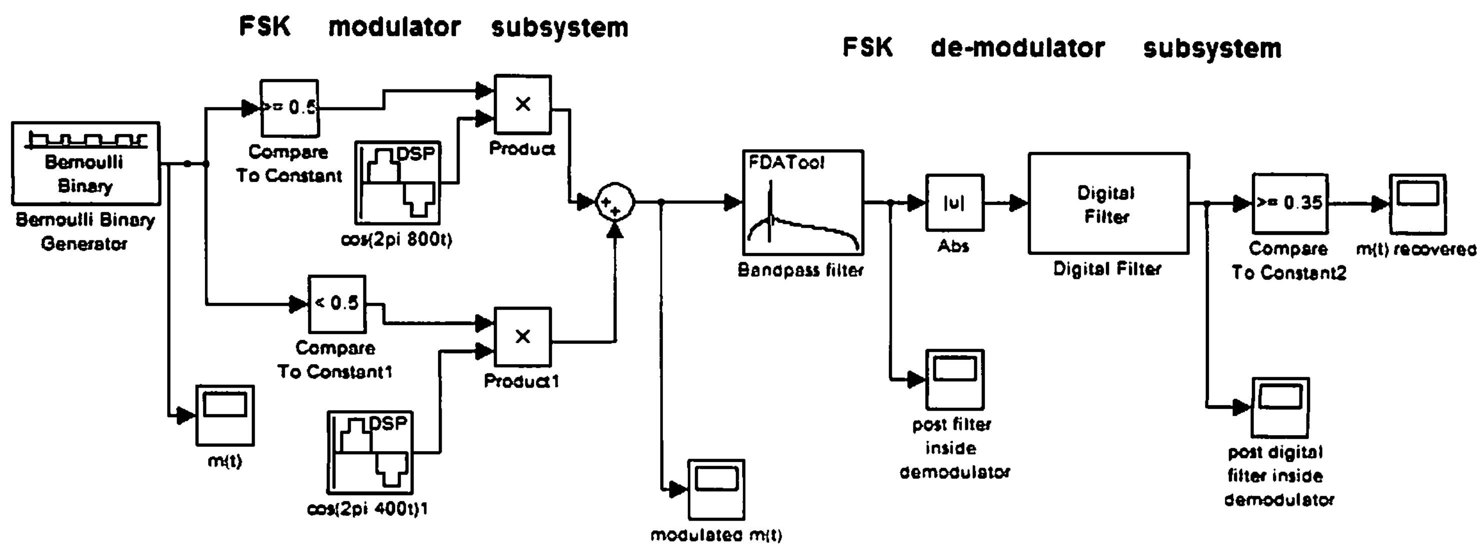
Computer Assignment #6
Nasser M. Abbasi
Cal Poly Pomona, CA
SUmmer session 1, 2010

problem description is here

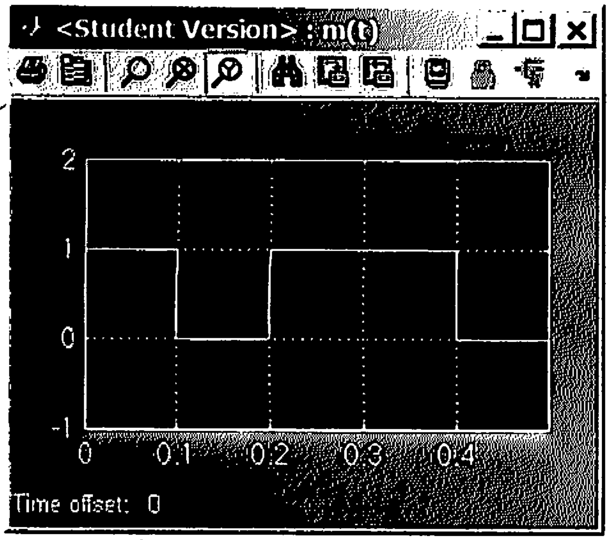
Simulink model

model is here

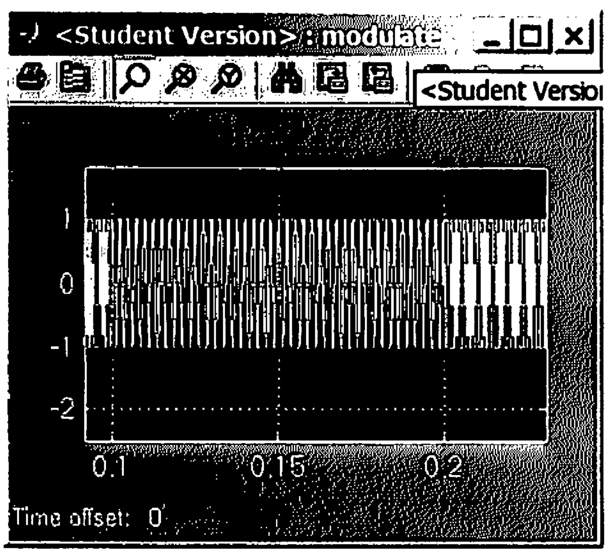
Computer assignment #6
modulation and demodulation of Binary FSK
by Nasser M. Abbasi. cal poly. summer session 1. 2010



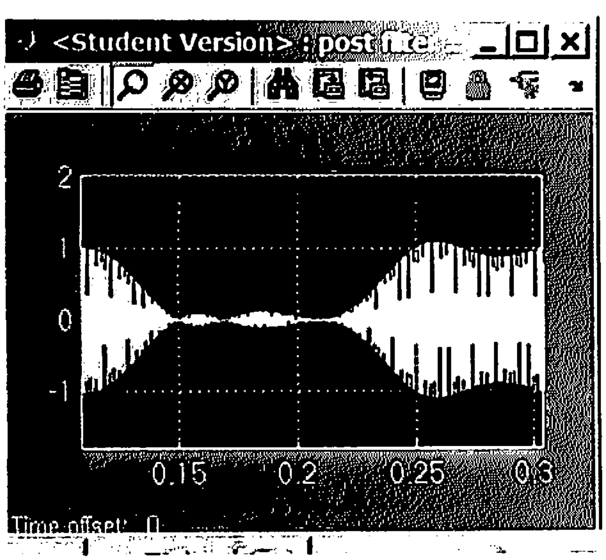
Output and result



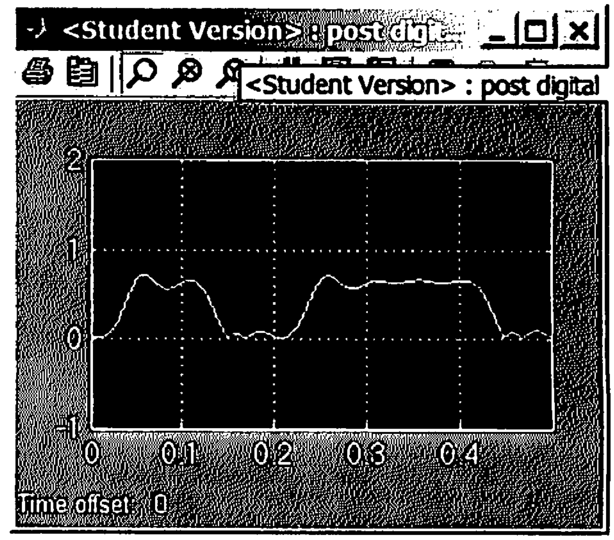
Original message m(t)



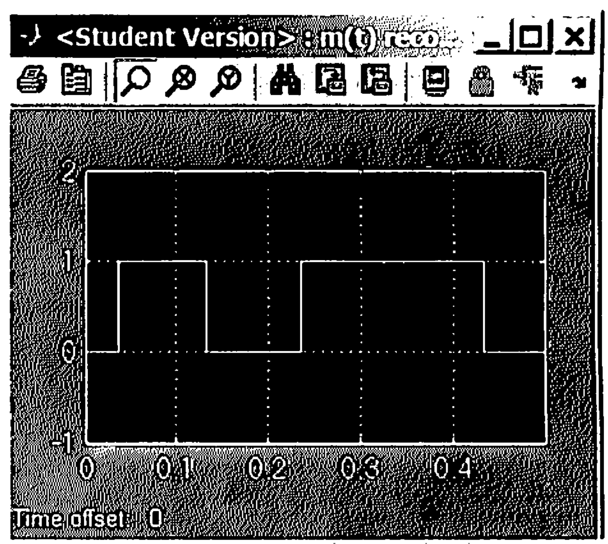
S(t), the modulated carrier message



Output of digital filter inside the demodulator



Output of bandpass filter in the demodulator



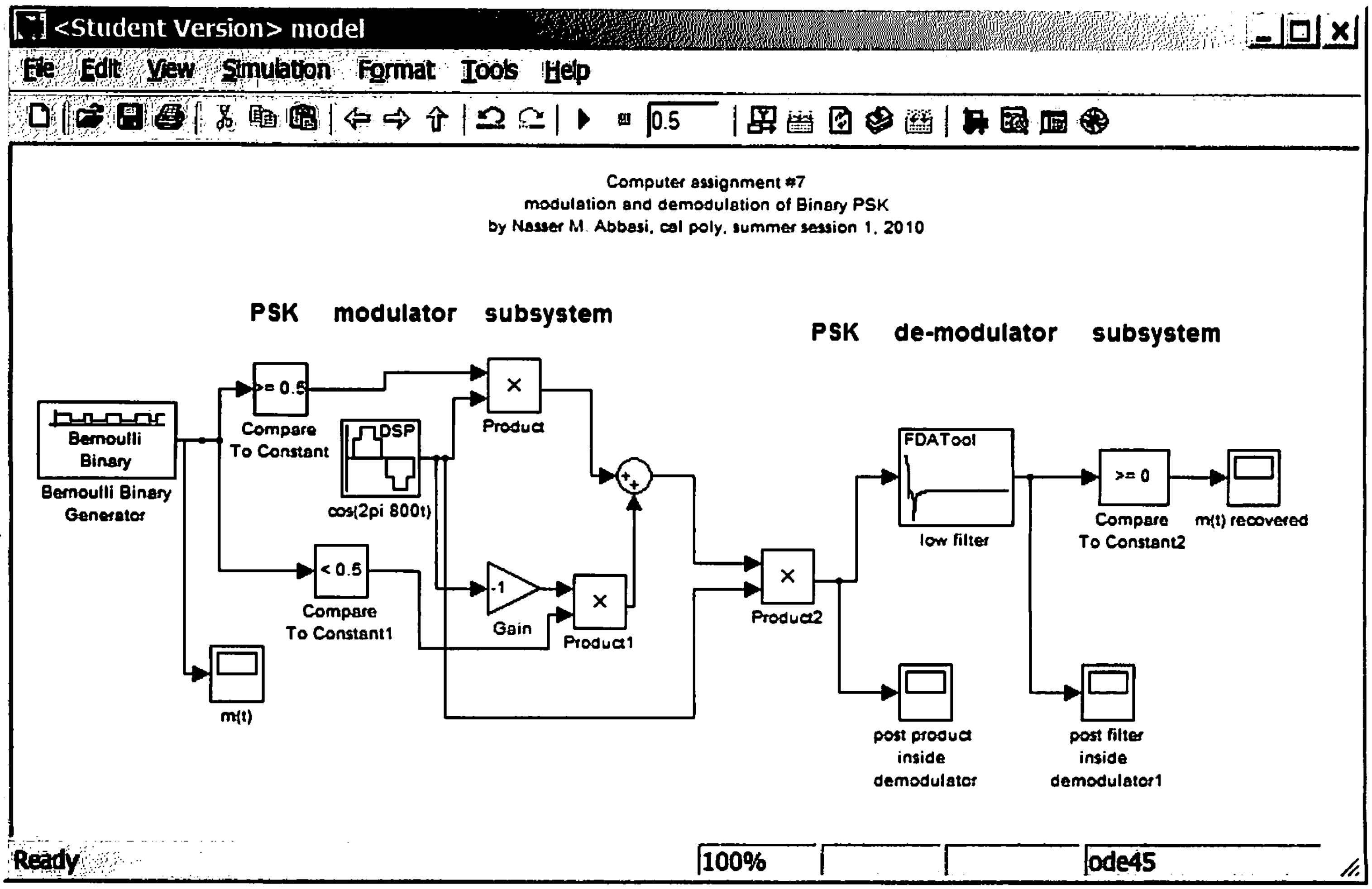
demodulated message m(t). Notice some delay at the start compared to original m(t)

Computer Assignment #7
Nasser M. Abbasi
Cal Poly Pomona, CA
SUmmer session 1, 2010

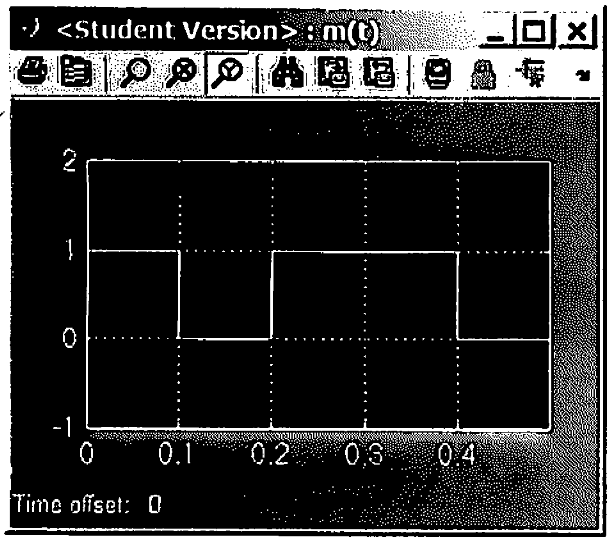
problem description is here

Simulink model

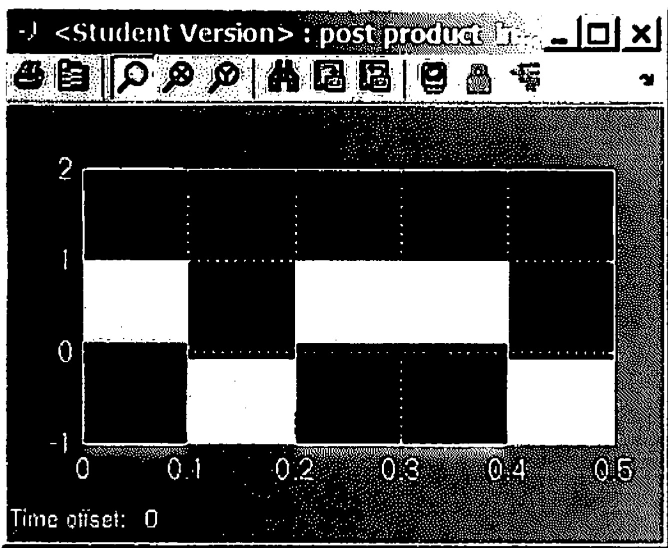
model is here



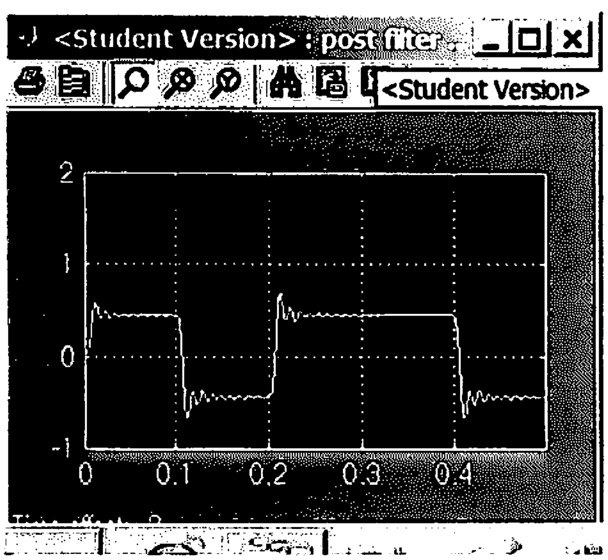
Output and result



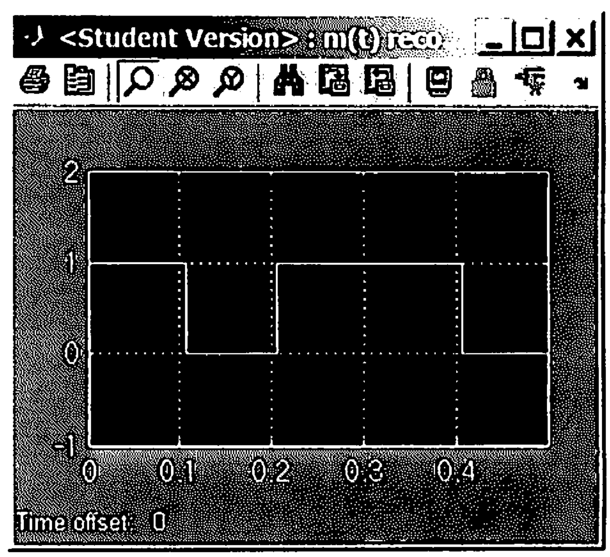
Original binary message m(t)



Modulated carrier s(t)



Message after passing the low pass filter, before compare



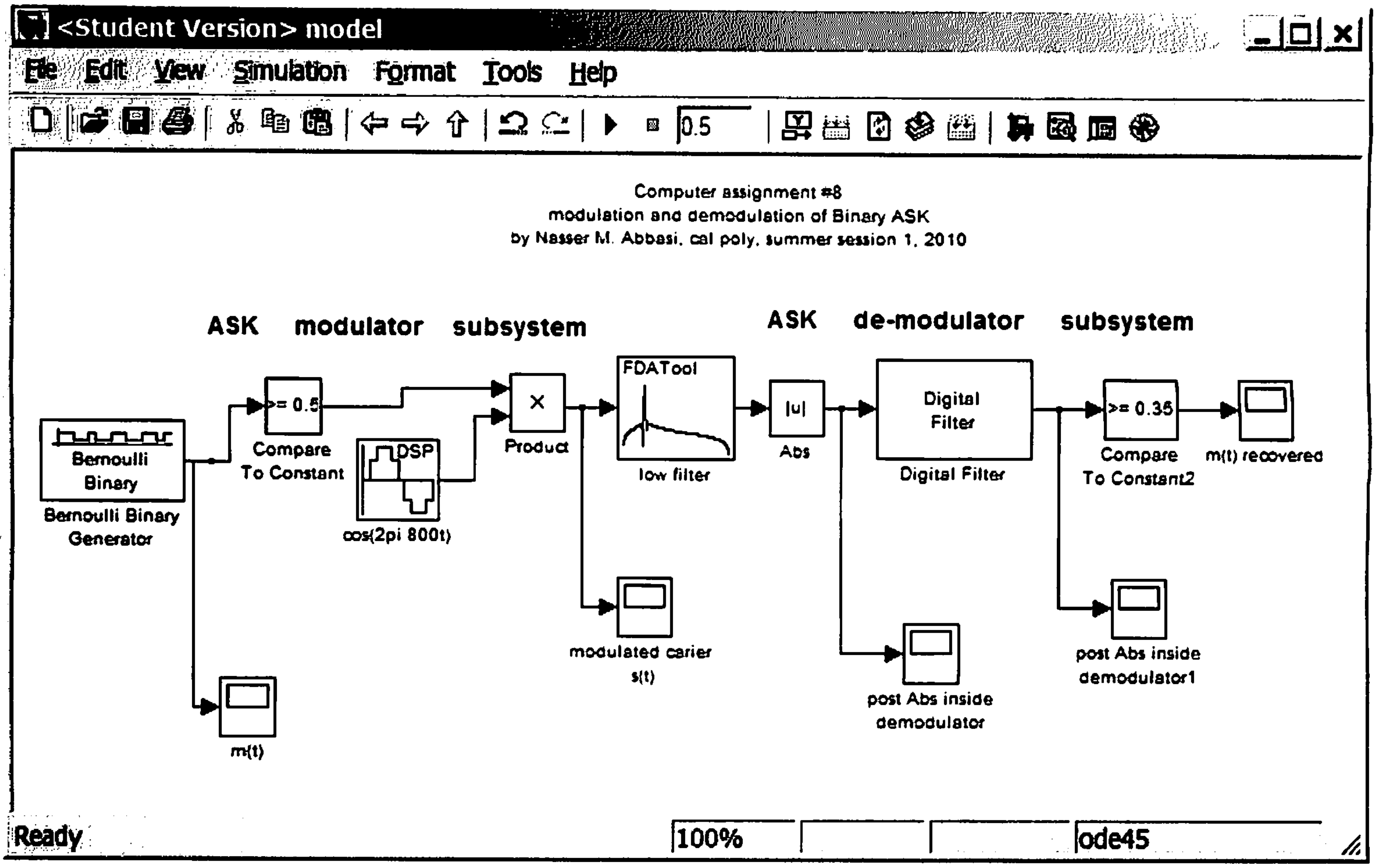
Recovered (demodulated) binary message m(t)

Computer Assignment #8
Nasser M. Abbasi
Cal Poly Pomona, CA
SUmmer session 1, 2010

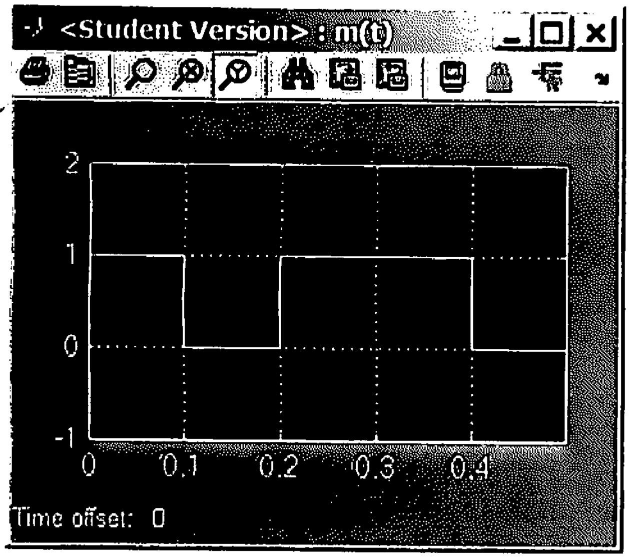
problem description is here

Simulink model

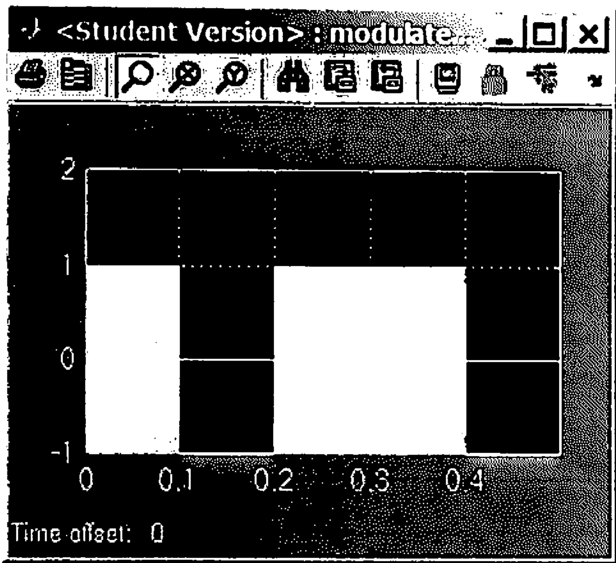
model is here



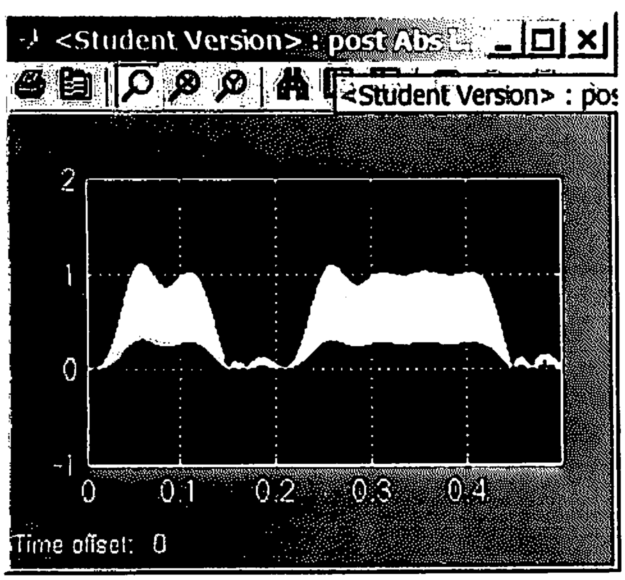
Output and result



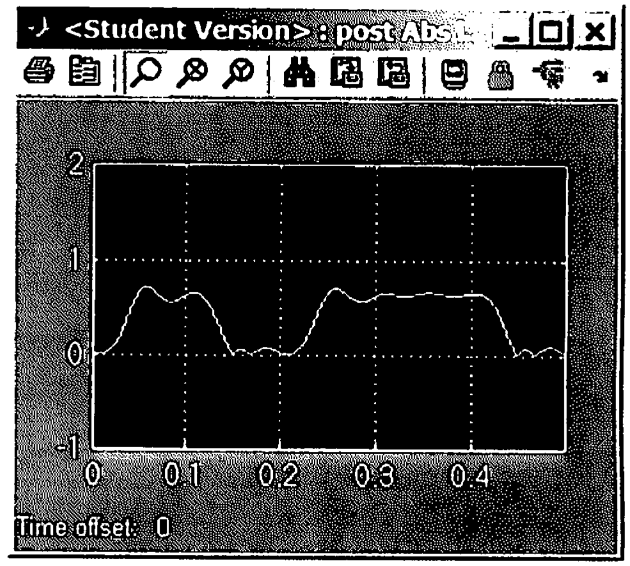
Original message m(t)



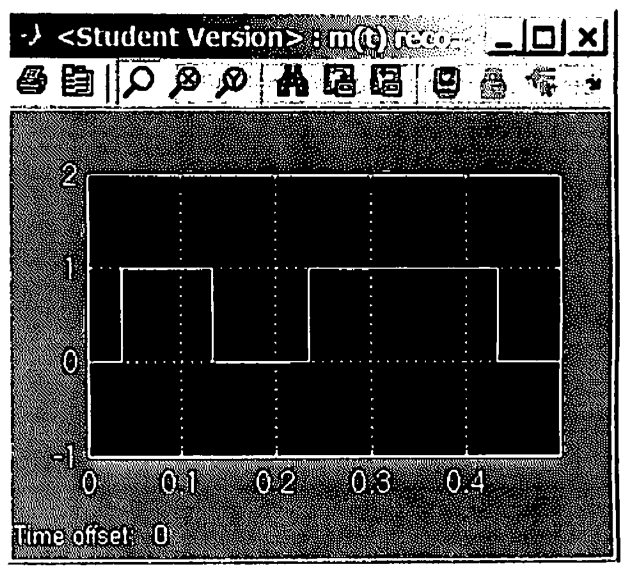
Modulated carier s(t)



After check on Abs value



After the digital filter



Demodulated message m(t) recovered. Notice delay at the start compared to original message m(t)