ECE-S 631:
Fundamentals of Deterministic DSP

Prof. John Walsh

Project Report #1

Zexi Liu

Nov 1st, 2009

Electrical and Computer Engineering
Drexel University

Project I: Phase Tracking and Baud Timing Correction Systems

ECES 631, Prof. John MacLaren Walsh, Ph. D.

1 Purpose

In this lab you will encounter the utility of the fundamental Fourier and z-transform theory that you are learning in this course by using it to design carrier phase tracking and baud timing synchronization systems in a digital communications receiver. Feel free to work in groups. After you have completed this lab, you should know how to

- Plot a discrete time linear time invariant system's frequency response using freqz.
- Design a FIR filter using firls and firpm[1] .
- Follow the signal through a simple QAM communications system to determine its operating principles and design filters that are needed for it.
- Design and analyze a carrier phase tracking loop that is capable of tracking phase error signals that are constant offsets and ramps.
- Design and analyze a baud timing correction loop.

2 Getting Comfortable With Matlab

If you have never used Matlab before, carefully read through A Very Brief Introduction to Matlab, which can be found at http://www.ece.drexel.edu/walsh/eces631/matlabIntro.pdf. Make sure you type in the MATLAB commands as you read that document so that you can get used to the MATLAB syntax.

3 Plotting a Signal's DTFT Using freqz

It is easy to plot the frequency response of a finite impulse response filter with MATLAB – just use freqz. In particular, suppose the impulse response of the filter is $h[n]=5\delta[n]+3\delta[n-1]+4\delta[n-2]$. You can plot the frequency response of this filter with the MATLAB code

```
>> h=[5,3,4];

>> figure;

>> freqz(h);
```

which produces the plot shown in Figure 1. Note that because the impulse response has all real coefficients, MATLAB has exploited the symmetry of the DTFT to plot only the positive half of the frequency axis that we normally plot. The plot may be given labels on the x and y axes using the command xlabel and ylabel, respectively, and may be given a title using the command title. Read the help pages for these commands for further information.

In the case that the filter has complex coefficients, so that its frequency response is not necessarily conjugate symmetric, one can use the alternate command

```
>> freqz(h,1,'whole');
```

to force MATLAB to plot the full frequency response from $[0, 2\pi)$.

[1] The older command remez has now been replaced with firpm.

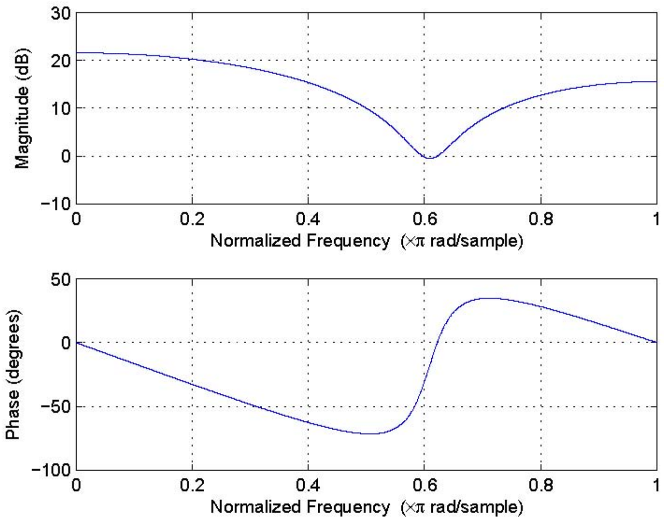


Figure 1: The frequency response plot generated by freqz.

3.1 Problems to do, print up, and hand in

1. What DTFT symmetry property do frequency responses of filter with purely real impulse responses have?

$$H(e^{j\omega}) = H^*(e^{-j\omega})$$
$$H_R(e^{j\omega}) = H_R(e^{-j\omega})$$
$$H_I(e^{j\omega}) = -H_I(e^{-j\omega})$$
$$|H(e^{j\omega})| = |H(e^{-j\omega})|$$
$$< H(e^{j\omega}) = -< H(e^{-j\omega})$$

2. Plot the frequency response of the filter whose impulse response is

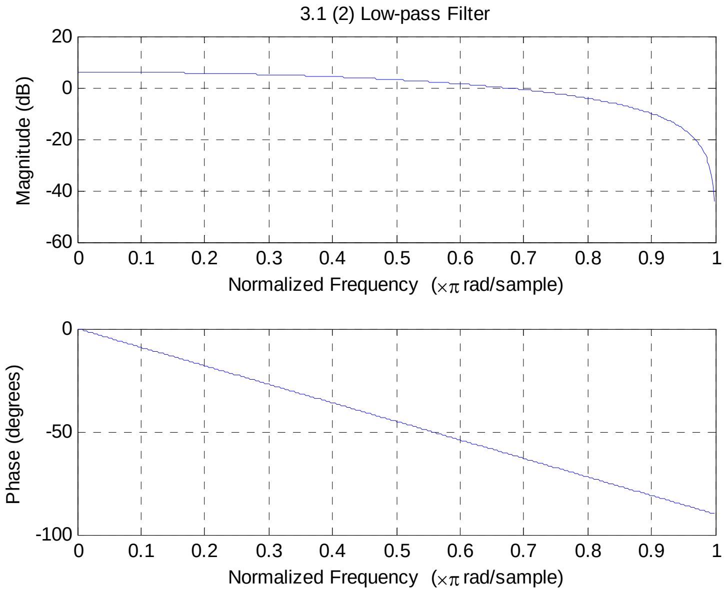


Figure 2: The frequency response plot generated by freqz.

Figure 2 shows a type II FIR linear phase low-pass filter.

$$h[n] := \begin{cases} 1 & n = 0,1 \\ 0 & otherwise \end{cases}$$

What type of filter is this? Label the x and y axes of your plots and include titles.

3. Plot the frequency response of the filter whose impulse response is

$$h[n] := \begin{cases} 1 & n = 0 \\ -1 & n = 1 \\ 0 & otherwise \end{cases}$$

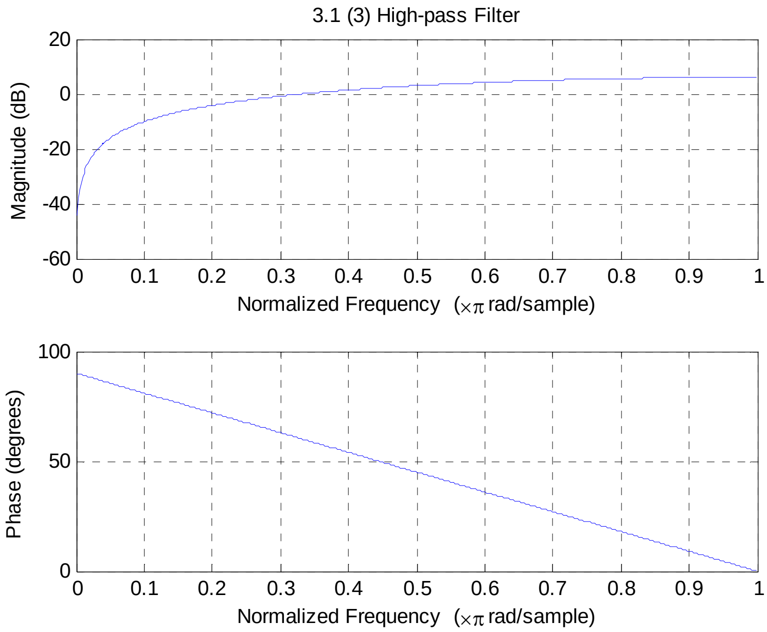


Figure 3: The frequency response plot generated by freqz.

Figure 2 shows a type IV FIR linear phase high-pass filter.

What type of filter is this? Label the x and y axes of your plots and include titles.

4. Plot the frequency response from [0, 2π) of the filter whose complex impulse response coefficients are

$$h[n] := \begin{cases} e^{j\pi/4} & n = 0 \\ e^{-j\pi/4} & n = 1 \\ 0 & otherwise \end{cases}$$

What is the response of this filter to the signal x[n]= $e^{j\pi n/2}$? What is the response of this filter to the signal x[n]= $e^{-j\pi n/2}$? Why did the axis of this frequency response plot need to be of length 2π?

$$y_1[n] = h[n] * x_1[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = h[0]x[n] + h[1]x[n-1]$$
$$= e^{j\pi/4}e^{j\pi n/2} + e^{-j\pi/4}e^{j\pi(n-1)/2}$$

$$y_2[n] = h[n] * x_2[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = h[0]x[n] + h[1]x[n-1]$$
$$= e^{j\pi/4}e^{-j\pi n/2} + e^{-j\pi/4}e^{-j\pi(n-1)/2}$$

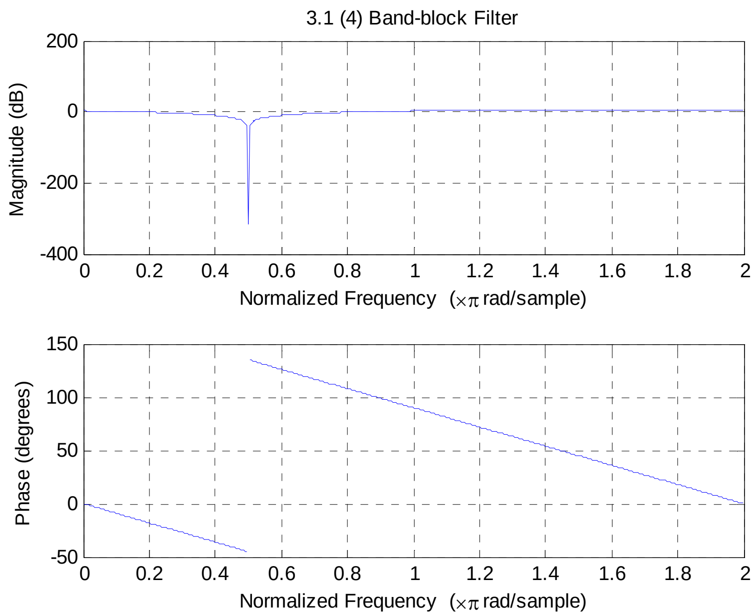


Figure 4: The frequency response plot generated by freqz.

The axis of this frequency response plot needs to be of length 2π because the filter impulse response has complex coefficient such that it doesn't have the conjugate symmetric.

Matlab code

```
clc;
clear;
% example
h0 = [5 3 4];
figure(1);
freqz(h0);
% freqz(h0, 1, 'whole');
% 3.1 (2)
h1 = [1 1];
figure(2);
% freqz(h1, 1, 'whole');
freqz(h1, 1);
title('3.1 (2) Low-pass Filter');
% 3.1 (3)
h2 = [1 -1];
figure(3);
freqz(h2, 1);
% freqz(h2, 1, 'whole');
title('3.1 (3) High-pass Filter');
% 3.1 (4)
```

```
h3 = [exp(1i*pi/4) exp(-1i*pi/4)];
figure(4);
freqz(h3, 1, 'whole');
% freqz(h3, 1);
title('3.1 (4) Band-block Filter');
% freqz(h3);
n = 0:25;
x1 = exp(1i*pi*n/4);
figure(5);
% y1 = filter(h3, 1, x1);
y1 = conv(h3, x1);    % the same as "filter"
subplot(4,1,1);
stem(x1);
xlabel('n');
ylabel('x1[n]');
title('x1[n]=exp(j*pi*n/4)');
subplot(4,1,2);
stem(y1);
xlabel('n');
ylabel('y1[n]');
title('y1[n]=x1[n]*h[n]');
x2 = exp(-1i*pi*n/4);
y2 = filter(h3, 1, x2);
subplot(4,1,3);
stem(x2);
xlabel('n');
ylabel('x2[n]');
title('x2[n]=exp(-j*pi*n/4)');
subplot(4,1,4);
stem(y2);
xlabel('n');
ylabel('y2[n]');
title('y2[n]=x2[n]*h[n]');
```

4 Designing FIR Filters Using firls and firpm

The MATLAB commands firls and firpm implement methods for designing optimal linear phase discrete time filter using the least-squares and equiripple (minimax) criteria, respectively. We will discuss the inner operation of these algorithms later on in the course. For now it suffices for you to know how to use the implementation of these commands in MATLAB in order to design a filter to match the frequency response constraints that you want.

Let's begin by reading the help page for firls by typing help firls [Enter] at the MATLAB prompt, and reading the text that is displayed. For our basic filter design needs in this course, the syntax

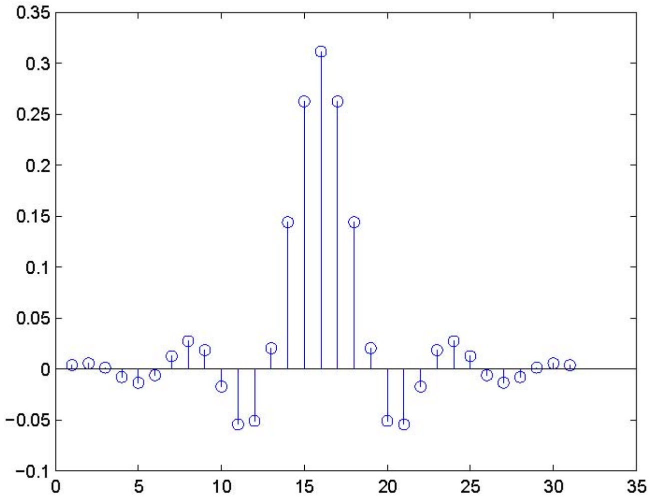


Figure 5: The impulse response of the designed filter.

myFilter=firls(N,F,A);
will be sufficient. Here, N is the order of the filter (i.e. the length of its impulse response -1), which we will choose to be an even number for reasons we will discuss later. F is a vector of frequencies in between 0 and 1, where 1 corresponds to the radian frequency π and 0 corresponds the radian frequency 0. A is a list of desired frequency response magnitudes associated with these frequencies. firls attempts to design a real linear phase filter of length N whose magnitude response (magnitude of frequency response) best matches the line between A(k) at radian frequency F(k)π and A(k+1) at radian frequency F(k+1)π when k is odd and between 1 and N. For even k, F(k)π to F(k+1) is considered to be a “don’t care” region in which the designer does not care what the frequency response will be. The “best match” specified above, is in the sense of minimizing the squared error between the magnitude response and its desired values outside of the “don’t care regions, i.e. in the sense of minimizing

where $\hat{H}(e^{j\omega})$ is the frequency response of the filter being designed. The designed filter is returned by firls into the output vector myFilter.

The command firpm operates with the same syntax, i.e. myFilter=firls(N,F,A);. The only difference is that the filter it returns is the “best match” in a different sense, namely, it minimizes the maximum difference between the frequency response of the designed filter and the frequency constraints, i.e. it minimizes

Now that we have introduced firls and firpm, let’s do an example filter design. Suppose that we wish to design a length 31 linear phase low pass filter which passes (radian) frequencies below $\pi/4$ and blocks radian frequencies above $3\pi/8$, making a stem plot of the resulting impulse response and plotting the resulting frequency response. We could do so with the MATLAB code

```
>> myFilt=firls(30,[0 1/4 3/8 1],[1,1,0,0]);
>> figure; stem(myFilt);
>> figure; freqz(myFilt);
```

which produces the plots shown in Figure 2 and 3.

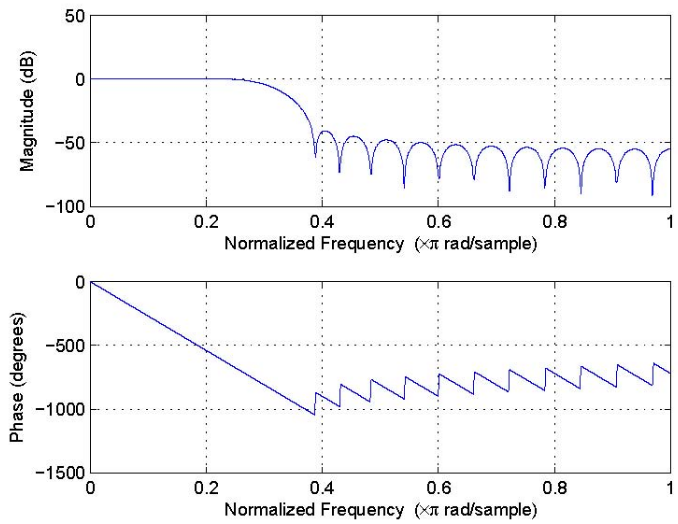


Figure 6: The frequency response of the designed filter.

4.1 Problems to Do, Print Up, and Hand In

1. Design a length 35 FIR bandpass filter which passes frequencies between 1π/8 and 3π/8 using firls. Make a stem plot of the resulting impulse response and a plot of its frequency response.

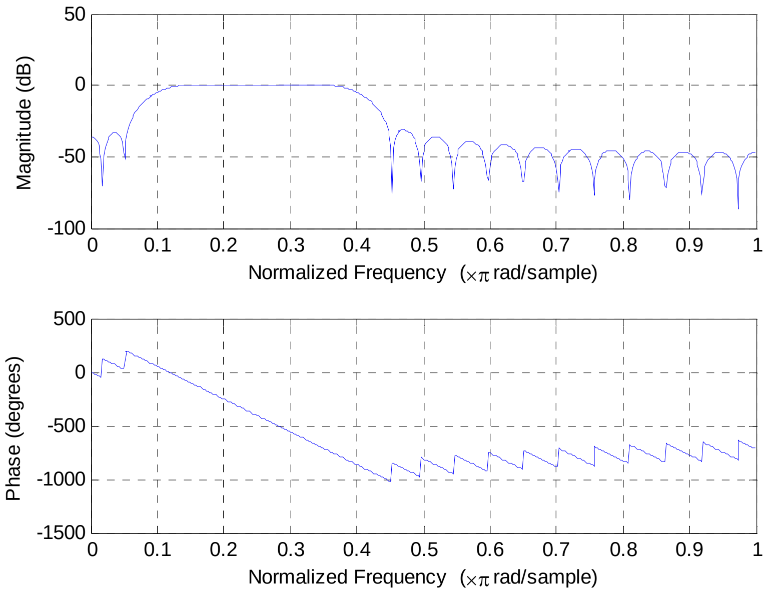


Figure 7: The frequency response of bandpass filter designed using firls.

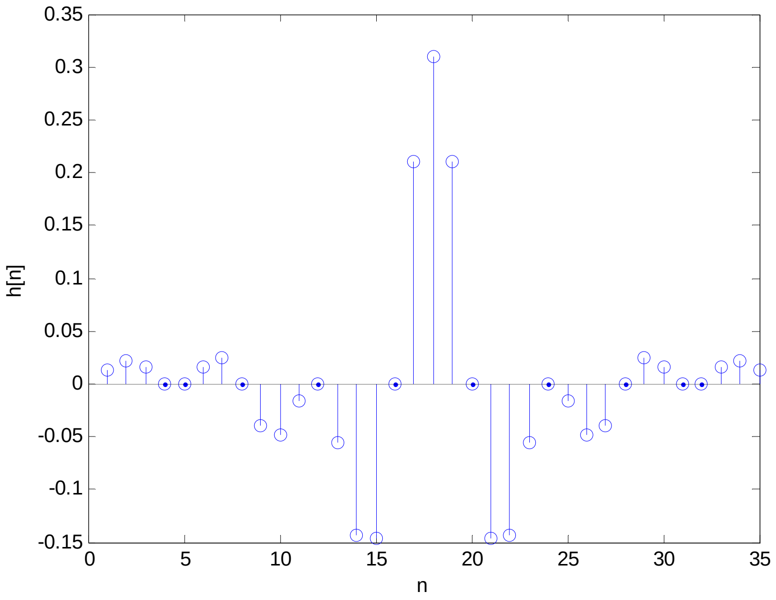


Figure 8: The impulse response of the designed filter.

2. Design the same filter using firpm. Make a stem plot of the resulting impulse response and a plot of its frequency response.

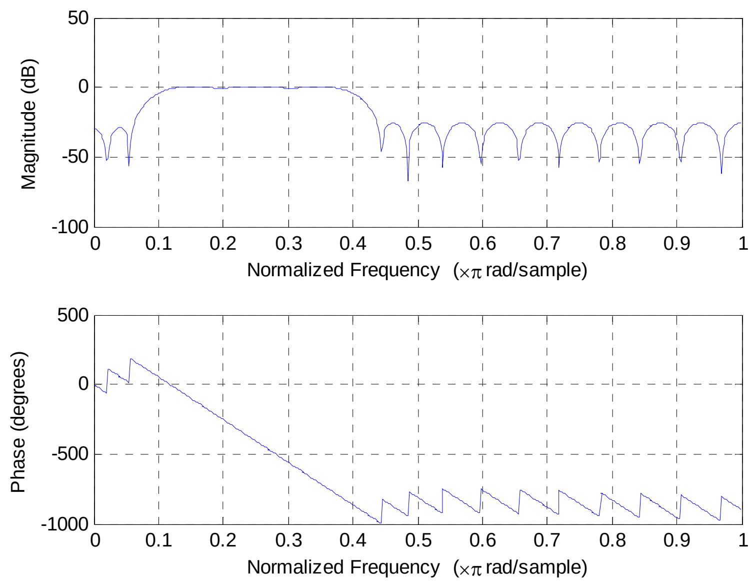
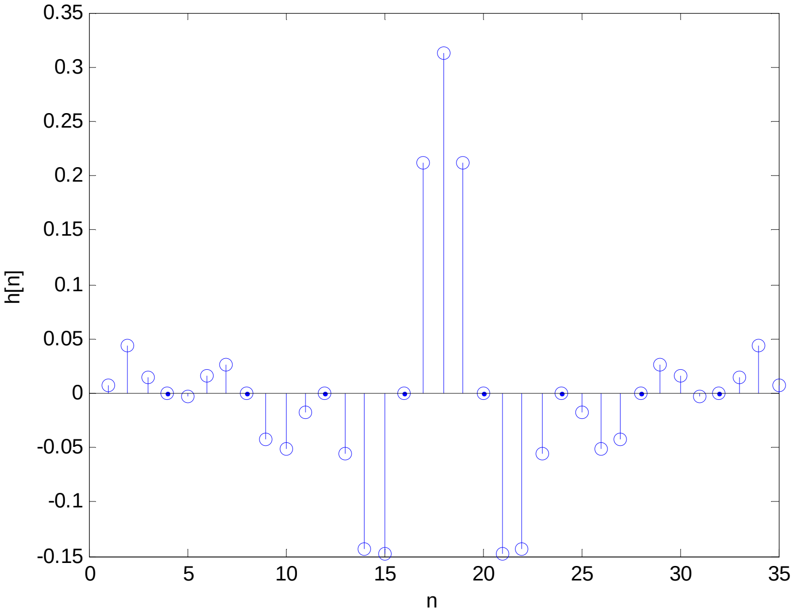


Figure 9: The frequency response of bandpass filter designed using firpm



.

Figure 10: The impulse response of the designed filter.

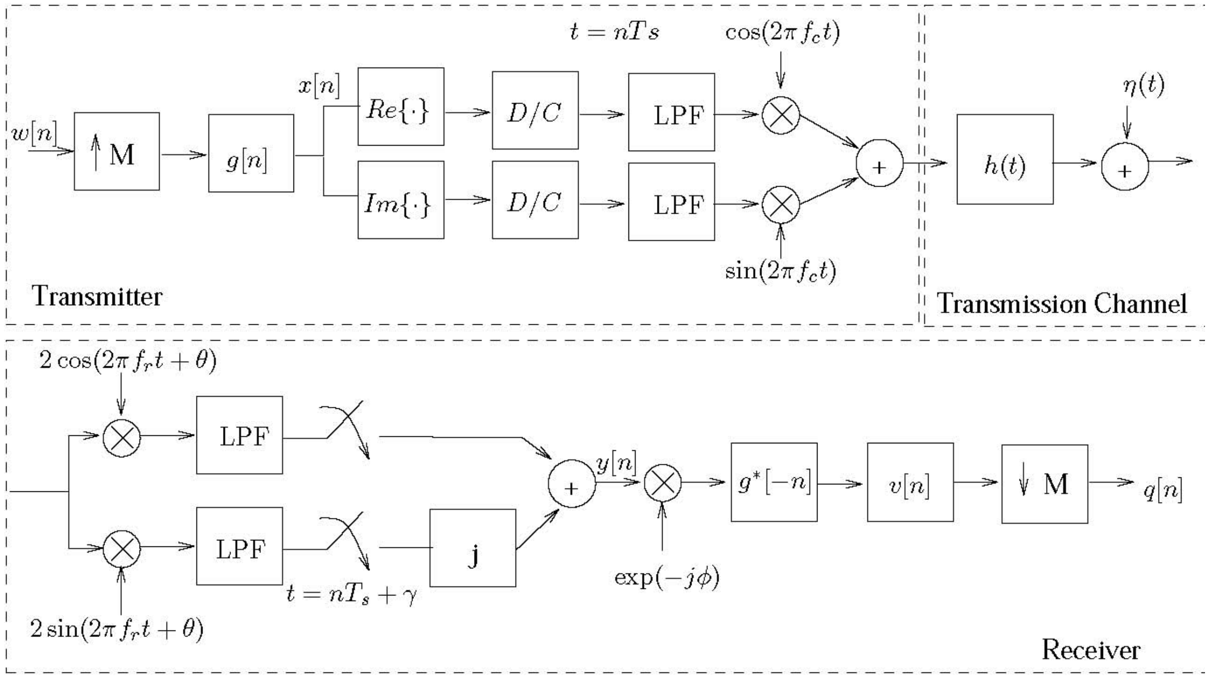


Figure 11: A simple digital transmitter and receiver.

5 Fourier Analysis of a Digital Transmission System

Now that you have become familiar with a few ways to do some of the most basic signal processing tasks in MATLAB (design a filter and plot a frequency response), it is time to apply this newly learned tool to enhance your understanding of the Fourier Transform and its properties within an applied domain. Let's consider the operation of the toy digital transmitter and receiver pictured in Figure 4. Suppose that the low pass filters are ideal and band-limited to. The box labeled j takes its input signal and multiplies by j = −1 to produce its output signal. The output from the upper chain (which is the transmitter and channel) is fed as an input into the lower chain (which is the receiver).

5.0.1 Problems to Solve and Hand In

1. Begin by assuming ideal channel conditions, so that the impulse response of the channel is h(t)= δ(t), and there is no receiver electronics noise η(t) = 0. Assume an idealized direct conversion receiver so that f_r = fc. Also assume perfect synchronization between the sampling clocks at the transmitter and receiver so that γ = 0. The part of this transmitter/receiver pair relating x[n] and y[n] is called quadrature amplitude modulation and demodulation.

   a) What is the transfer function between x[n] and y[n] under these ideal conditions?

   In transmitter

$$x_R(t) \cdot cos(2\pi f_c t) + x_I(t) \cdot sin(2\pi f_c t)$$

   In transmission channel

$$\int_{-\infty}^{+\infty} [x_R(\tau) \cdot cos(2\pi f_c \tau) + x_I(\tau) \cdot sin(2\pi f_c \tau)] \cdot \delta(t-\tau) d\tau$$

$$= x_R(t) \cdot cos(2\pi f_c t) + x_I(t) \cdot sin(2\pi f_c t)$$

In receiver

$$[x_R(t)\cdot cos(2\pi f_c t)+x_I(t)\cdot sin(2\pi f_c t)]\cdot 2cos(2\pi f_r t+\theta)$$
$$=x_R(t)\cdot[\cos(4\pi f_c t+\theta)+\cos(\theta)]+x_I(t)\cdot[\sin(4\pi f_c t+\theta)-\sin(\theta)]$$

$$[x_R(t)\cdot cos(2\pi f_c t)+x_I(t)\cdot sin(2\pi f_c t)]\cdot 2sin(2\pi f_r t+\theta)$$
$$=x_R(t)\cdot[\sin(4\pi f_c t+\theta)+\sin(\theta)]-x_I(t)\cdot[\cos(4\pi f_c t+\theta)-\cos(\theta)]$$

After LPF's

$$x_R(t)\cdot\cos(\theta)-x_I(t)\cdot\sin(\theta)$$
$$x_R(t)\cdot\sin(\theta)+x_I(t)\cdot\cos(\theta)$$

In the end

$$y[n]=x_R(t)\cdot\cos(\theta)-x_I(t)\cdot\sin(\theta)+j\cdot[x_R(t)\cdot\sin(\theta)+x_I(t)\cdot\cos(\theta)]|_{t=nT_s}$$
$$=x_R(t)\cdot\cos(\theta)+j\cdot x_I(t)\cdot\cos(\theta)-x_I(t)\cdot\sin(\theta)+j\cdot x_R(t)\cdot\sin(\theta)\,|_{t=nT_s}$$
$$=\cos(\theta)\cdot[x_R(t)+j\cdot x_I(t)]+j^2[x_I(t)\cdot\sin(\theta)-j\cdot x_R(t)\cdot\sin(\theta)]|_{t=nT_s}$$
$$=\cos(\theta)\cdot[x_R(t)+j\cdot x_I(t)]+j\cdot\sin(\theta)\cdot[x_R(t)+jx_I(t)]|_{t=nT_s}$$
$$=\cos(\theta)\cdot x[t]+j\cdot\sin(\theta)\cdot x[t]|_{t=nT_s}$$
$$=x[n]\cdot[\cos(\theta)+j\cdot\sin(\theta)]$$
$$=x[n]\cdot e^{j\theta}$$

b) What is the purpose of the multiplier following y[n]? What should φ be?

Based on the transfer function in (a), the multiplier $e^{-j\phi}$ is aimed to restore $x[n]\cdot e^{j\theta}$ to $x[n]$.

In order to do so, $let\ \phi=\theta$.

c) The filter g[n] is called a (digital) pulse shaping filter. Suppose first that g[n] is an ideal low pass filter and v[n] = δ[n]. What should the LPF g[n]'s cut off frequency be to allow for maximum bandwidth perfect transmission between w[n] and q[n] (i.e. q[n] = w[n])?

The cut off frequency of LPF g[n] should be $\frac{\pi}{M}$.

d) Again, supposing v[n] = δ[n], but let g[n] be non-ideal. What are the requirements on the pulse shaping filter g[n] for q[n] = w[n]? Sometimes these requirements are referred to as the Nyquist criterion.

Since g[n] is non-ideal now, the ω_p of g[n] should be $\omega_p\geq\frac{\pi}{T_s}$ and the ω_s of g[n] should be $\omega_p\leq(\frac{2\pi}{M}-\frac{\pi}{T_s})$. On the other hand, in order to satisfy Nyquist ISI criterion, g[n] should be a root-raised-cosine filter.

e) Many receivers down convert to an intermediate frequency, so that $f_r < f_c$. If we were to use this receiver, what would be some possible values of f_r which would still allow w[n]= q[n] under the assumptions above? (You may change the cutoff frequency of the low pass filters for this part.)

In receiver

$$[x_R(t)\cdot cos(2\pi f_c t)+x_I(t)\cdot sin(2\pi f_c t)]\cdot 2cos(2\pi f_r t+\theta)$$
$$=x_R(t)\cdot[\cos(2\pi(f_c+f_r)t+\theta)+\cos(2\pi(f_c-f_r)\,t-\theta)]+x_I(t)\cdot[\sin(2\pi(f_c+f_r)t+\theta)+\sin(2\pi(f_c-f_r)t-\theta)]$$

$$[x_R(t)\cdot cos(2\pi f_c t) + x_I(t)\cdot sin(2\pi f_c t)]\cdot 2sin(2\pi f_r t+\theta)$$
$$= x_R(t)\cdot[\sin(2\pi(f_c+f_r)t+\theta) + \sin(2\pi(f_c-f_r)t+\theta)]$$
$$- x_I(t)\cdot[\cos(2\pi(f_c+f_r)t+\theta) - \cos(2\pi(f_c-f_r)t-\theta)]$$

After LPS's

$$x_R(t)\cdot\cos(2\pi(f_c-f_r)t-\theta) + x_I(t)\cdot\sin(2\pi(f_c-f_r)t-\theta)$$
$$x_R(t)\cdot\sin(2\pi(f_c-f_r)t+\theta) + x_I(t)\cdot\cos(2\pi(f_c-f_r)t-\theta)$$

In the end

$$y[n] = [x_R(t) + j\cdot x_I(t)]$$
$$\cdot\{\cos(2\pi(f_c-f_r)t+\theta) + j\cdot\sin(2\pi(f_c-f_r)t+\theta)\}|_{t=nT_s}$$
$$= x(t)\cdot e^{j[2\pi(f_c-f_r)t+\theta]}|_{t=nT_s}$$
$$= x[nT_s]\cdot e^{j[2\pi(f_c-f_r)nT_s+\theta]}$$

$$(f_c-f_r)T_s = any\ integer$$

2. Now assume the channel is less ideal and delays and scales its input, so that h(t)= $\alpha\delta(t-\tau)$, but continue to assume $\eta(t) = 0$ and additionally let the baud timing synchronization error γ be non-zero.

a) What is the frequency response of the system between x[n] and y[n] now?

In transmitter

$$x_R(t)\cdot cos(2\pi f_c t) + x_I(t)\cdot sin(2\pi f_c t)$$

In transmission channel

$$\int_{-\infty}^{+\infty}[x_R(\sigma)\cdot cos(2\pi f_c\sigma) + x_I(\sigma)\cdot sin(2\pi f_c\sigma)]\cdot\alpha\cdot\delta(t-\tau-\sigma)d\sigma$$
$$= \alpha\cdot x_R(t-\tau)\cdot cos(2\pi f_c(t-\tau)) + \alpha\cdot x_I(t-\tau)$$
$$\cdot sin(2\pi f_c(t-\tau))$$

In receiver

$$\alpha\cdot\left[x_R(t-\tau)\cdot cos\big(2\pi f_c(t-\tau)\big) + x_I(t-\tau)\cdot sin\big(2\pi f_c(t-\tau)\big)\right]$$
$$\cdot 2cos(2\pi f_r t+\theta)$$
$$= \alpha\cdot x_R(t-\tau)$$
$$\cdot[\cos(2\pi(f_c-f_r)t - 2\pi f_c\tau - \theta)$$
$$+ \cos(2\pi(f_c+f_r)t - 2\pi f_c\tau + \theta)] +$$
$$\alpha\cdot x_I(t-\tau)$$
$$\cdot[\sin(2\pi(f_c-f_r)t - 2\pi f_c\tau - \theta) + \sin(2\pi(f_c+f_r)t - 2\pi f_c\tau$$
$$+\theta)]$$
$$\alpha\cdot\left[x_R(t-\tau)\cdot cos\big(2\pi f_c(t-\tau)\big) + x_I(t-\tau)\cdot sin\big(2\pi f_c(t-\tau)\big)\right]$$
$$\cdot 2sin(2\pi f_r t+\theta)$$
$$= \alpha\cdot x_R(t-\tau)$$
$$\cdot[\sin(2\pi(f_c-f_r)t + 2\pi f_c\tau + \theta)$$
$$+ \sin(2\pi(f_c+f_r)t - 2\pi f_c\tau + \theta)] +$$
$$\alpha\cdot x_I(t-\tau)$$
$$\cdot[\cos(2\pi(f_c-f_r)t + 2\pi f_c\tau + \theta) - \cos(2\pi(f_c+f_r)t - 2\pi f_c\tau$$
$$+\theta)]$$

After LPF’s

$$\alpha \cdot x_R(t-\tau) \cdot \cos(2\pi(f_c - f_r)t - 2\pi f_c \tau - \theta) + \alpha \cdot x_I(t-\tau) \cdot \sin(2\pi(f_c - f_r)t - 2\pi f_c \tau - \theta)$$

$$\alpha \cdot x_R(t-\tau) \cdot \sin(2\pi(f_c - f_r)t + 2\pi f_c \tau + \theta) + \alpha \cdot x_I(t-\tau) \cdot \cos(2\pi(f_c - f_r)t + 2\pi f_c \tau + \theta)$$

Sampling with $t = nT_s + \gamma$

$$y[n] = \alpha \cdot x[nT_s - \tau + \gamma] \cdot e^{j \cdot [2\pi(f_c - f_r)nT_s + 2\pi(f_c - f_r)(\tau+\gamma)+\theta]}$$

$$= \alpha \cdot x[nT_s - \tau + \gamma] \cdot e^{j \cdot [2\pi(f_c - f_r)(nT_s+\gamma-\tau) + 4\pi(f_c - f_r)\tau+\theta]}$$

$$Y(f) = \alpha \cdot X(\frac{f + f_c - f_r}{f_s}) \cdot e^{j2\pi\frac{f}{f_s}(\gamma-\tau)} \cdot e^{j(4\pi(f_c - f_r)\tau+\theta)}$$

$$Y(f) = \alpha \cdot X(\frac{f}{f_s}) \cdot e^{j2\pi\frac{f}{f_s}(\gamma-\tau)} \cdot e^{j\theta}$$

b) If the g[n] are again ideal low pass filters with appropriate bandwidth, what should the impulse response of v[n] be to make w[n]= q[n]? (Hint: get the Frequency response of the system between the output of the up-sampler and the input to the filter v[n], and use this to determine requirements for the frequency response of v[n]).

3. Bonus – Extra Credit: Now assume that the signal is wireless and that the received signal is the sum of a "line of sight" propagation between the transmit and received antennas and a second component which reflects off the ground between the transmitter and receiver, so that $h(t)= \alpha_1\delta(t - \tau_1)+ \alpha_2\delta(t - \tau_2)$. Such channels are called multipath channels. Continue to neglect receiver noise by setting $\eta(t) = 0$.

a) What will the frequency response between x[n] and y[n]?

b) Suppose the impulse response between x[n] and y[n] can be effectively neglected outside the window $|n| \leq L$ for some L. If g[n] is an ideal low pass filter of appropriate bandwidth, and M = 2, can you come up with conditions for the existence of a finite impulse response filter v[n] that guarantees q[n]= w[n − d]? Such a filter is called a fractionally spaced equalizer because it operates at a higher sample rate than w[n] and q[n].

5.1 Verification with MATLAB

In this section you will implement a matlab model of the system in figure 4. Since MATLAB works natively with discrete time signals, you will use the transfer function between x[n] and y[n] that you determined in problem 2 in section 5.0.1 to produce y[n] directly from x[n].

5.1.1 Code to Implement and Hand In

1. Write a MATLAB implementation of the communications system in Figure 4 under the previously discussed assumptions and by producing y[n] directly from x[n] using the transfer function you determined. You may design g[n] with firpm or firls. You will want to use the command conv which produces the convolution of two input arguments as its output argument. Choose as your input signal w[n] = exp(j.01πn), e.g. make a vector w = exp(j*.01*pi [0: 1000]), and make stem plots of the real and imaginary parts of the output q[n]. Hand in your matlab code and your plot.

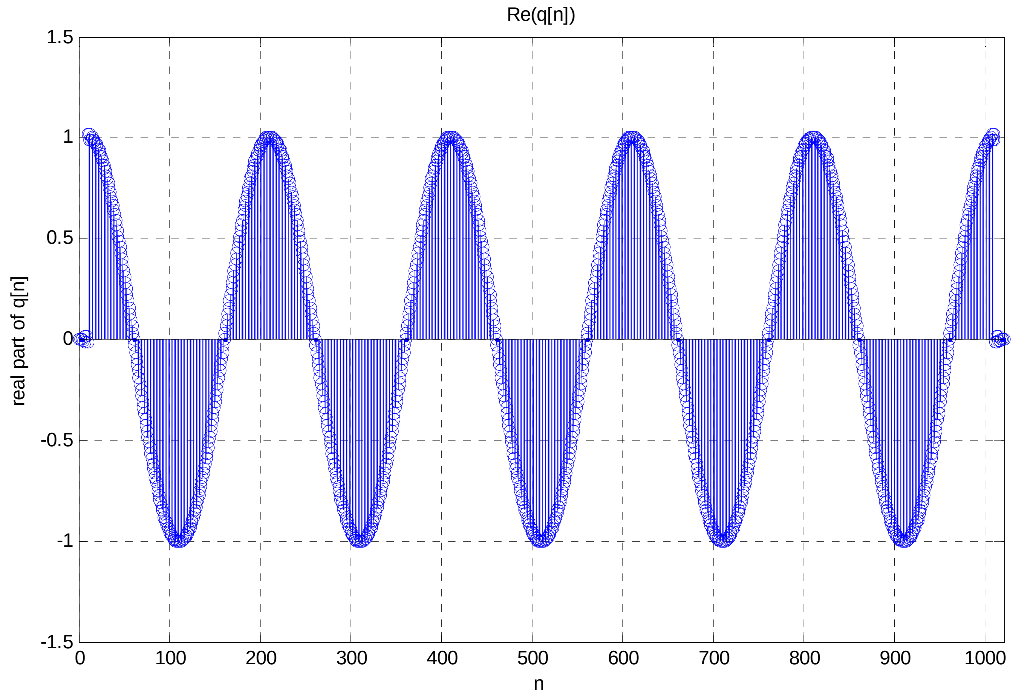


Figure 12: Real part of output q[n].

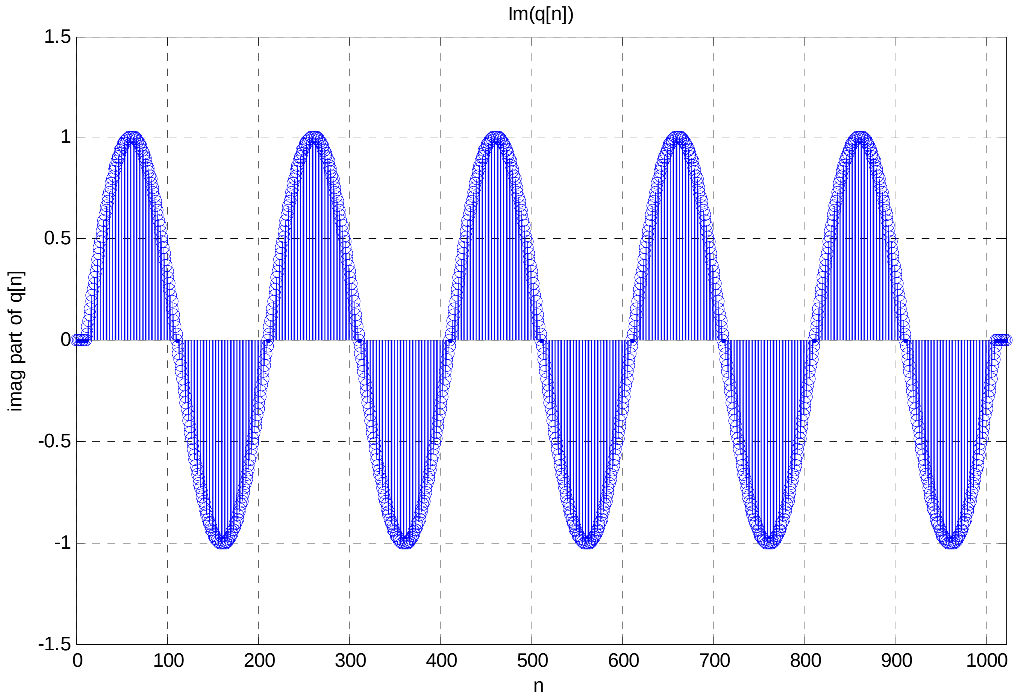


Figure 13: Imaginary part of output q[n].

Matlab code

```
clc;
clear;

M = 3;
theta = pi;
n = 0:1000;
w = exp(1i*0.01*pi*n);
w1 = upsample(w, M);
gn = firls( 30, [0 1/3 3/8 1], [1,1,0,0] );
x = conv(w1, gn);
y = x * exp(1i * theta);
y1 = y * exp(-1i * theta);
y2 = conv(y1, conj(fliplr(gn)));
vn = M;
y3 = conv(y2, vn);
q = downsample(y3, M);

figure(1);
stem(real(q));
grid on;
```

```
xlabel('n');
ylabel('real part of q[n]');
title('Re(q[n])');
figure(2);
stem(imag(q));
grid on;
xlabel('n');
ylabel('imag part of q[n]');
title('Im(q[n])');
```

2. Bonus – Extra Credit: Implement the system from problem 3 for a particular collection of $\alpha_1, \alpha_2, \tau_1, \tau_2$ and for your fractionally spaced equalizer. Demonstrate its proper operation with an input signal of your choice.

6 Z-Transform Analysis of a Phase Tracking System

Consider the system depicted in Figure 5, variations upon which are commonly used to deal with lack of synchronization between oscillators in systems such as the one studied in section 5. The overall goal of the system is to have the phase signal θ[n] match the phase signal φ[n].

6.0.2 Problems To Solve and Hand in

1. Define the signal $e[n] := \theta[n] - \varphi[n]$. Using the trigonometric identity

together with the small angle approximation

$$\sin(e[n]) \approx e[n]$$

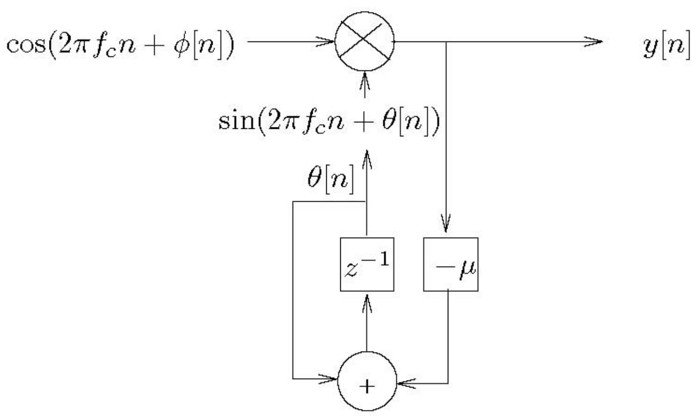


Figure 14: A model for a phase tracking system.

and neglecting contributions from the high frequency 2fc, write the rational transfer function H(z) (z-transform of the impulse response) through which e[n] is related to φ[n] in Figure 5. (i.e. e(z)= H(z)φ(z), where e(z) is the z-transform of e[n] and φ(z) is the z-transform of φ[n].)

$$H(z) = \frac{0.5}{1 + \frac{\mu}{1 - z^{-1}}} = \frac{0.5 \cdot (1 - z^{-1})}{1 - z^{-1} + \mu}$$

2. Suppose that θ[n] = 0 for n< 0, can this system track a small step change in φ? In other words, if μ is a small number, is the asymptotic value of the phase error e[∞] zero when φ[n]= αu[n] for some small phase α and the unit step u[n]? Prove your answer. (Use the linearized system with transfer function H(z) as discussed in the previous question and the final value theorem of unilateral z transforms).

3. Now suppose that $\theta[n] = 0$ for $n < 0$ and $\varphi[n]=(\beta n + \alpha)u[n]$ for some small β and α. Is the asymptotic value of the phase error $e[\infty]$ zero? If not, can you use your answer from the previous question to draw the diagram for a new system design for which the asymptotic value of the phase error (under the simplifying approximations from question one above) will be zero?

6.1 Verification with MATLAB

In this section you will demonstrate the accuracy of your analytical results in the previous section. It is important to do this to verify that, at least in some situations, the approximations made did not introduce too many errors.

6.1.1 MATLAB Code To Write

1. Using a for loop, code up the system pictured in Figure 5 for $n \geq 0$. Let $f_c = .1$. Initialize the system at $\theta[0] = 0$, and define $\varphi[n] = .01$ for all n. Pick a small step size μ, say .005, and let the system run to sufficiently large n so that a plot of the error $\theta[n] - \varphi[n]$ becomes very small. Turn in your matlab code and a plot of the error e[n].

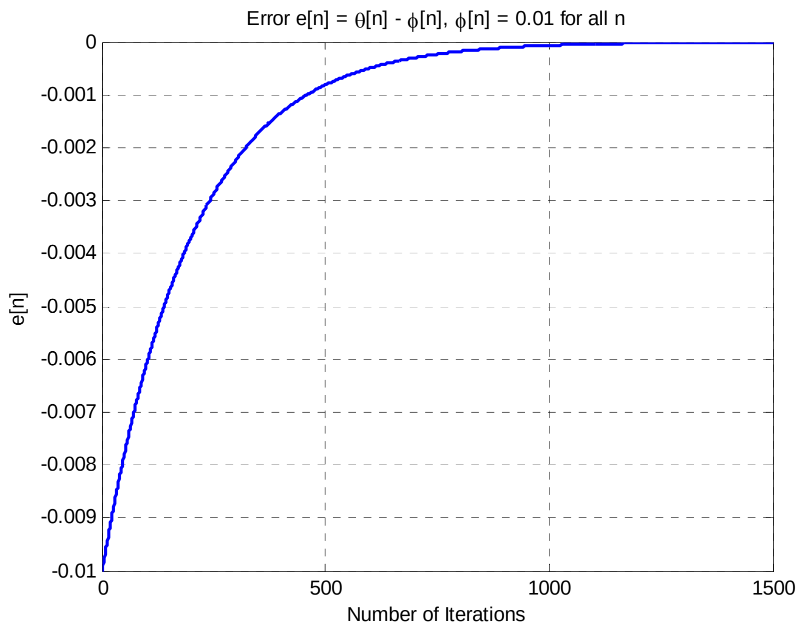


Figure 15: Plot of error e[n] when $\varphi = 0.01$ for all n

Matlab code

```
clc;
clear;

N = 1500;
phi(1, 1:N) = 0.01;
e(1, :) = 0;
theta(1, :) = 0;
for i = 1:N
    e(i) = theta(i) - phi(i);
    theta(i+1)=theta(i)-0.005*e(i);
end
figure(1);
plot(e);
grid on;
xlabel('Number of Iterations');
ylabel('e[n]');
title('Error e[n] = \theta[n] - \phi[n], \phi[n] = 0.01 for all n');
```

2. Repeat the previous exercise but changing φ[n]= βn. Demonstrate the tracking behavior you determined in problem 3 of section 6.0.2 by choosing a sufficiently small β and μ. Submit your matlab code and a plot of the error e[n].

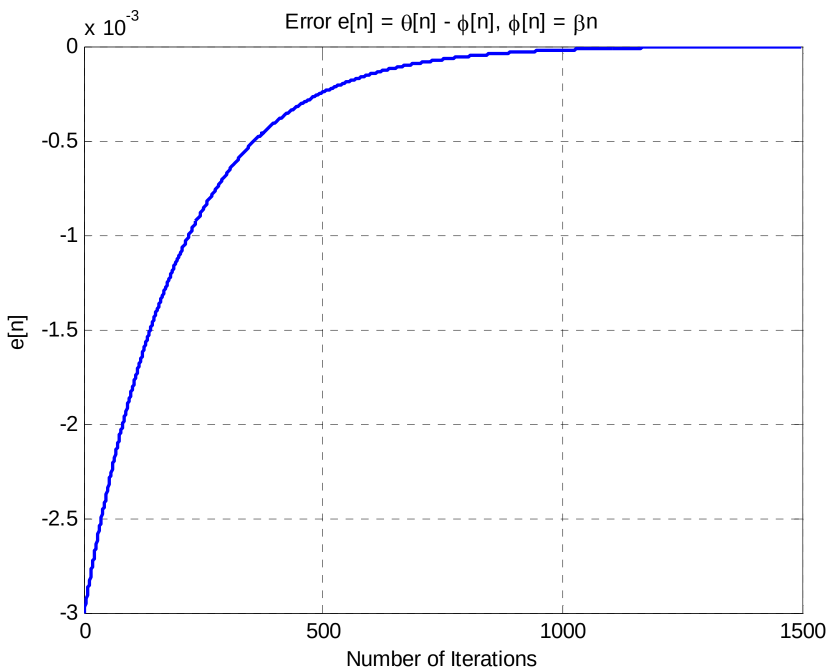


Figure 16: Plot of error e[n] when φ = βn

Matlab code

```
clc;
clear;

N = 1500;
phi(1, 1:N) = 0;
for i = 1:N
    phi(i) = 0.003*i;
end
e(1, :) = 0;
theta(1, :) = 0;
for i = 1:N-1
    e(i) = theta(i) - (phi(i+1) - phi(i));
    theta(i+1) = theta(i)-0.005*e(i);
end
figure(2);
plot(e);
grid on;
xlabel('Number of Iterations');
ylabel('e[n]');
title('Error e[n] = \theta[n] - \phi[n], \phi[n] = \betan');
```