CS-215: Experiment 1B

Soumen Pradhan – 1912176

15 . 02 . 2021

1. Sampling Rates

Aim

To sample an analog signal $x(t) = 4\cos(2\pi t)$ at 3 different sampling rates to generate a discrete-time signal $x[n] = x(nT_s)$.

Theoretical Background

A discrete function is a function whose range is a set of countably finite or countably infinite points. It can be generated from an analog counterpart or from a list of arbitary data.

It is used to store and create analog signals in a digital system.

Methodology

- A vector of discrete points are generated using the sampling period T_s.
- Corresponding discrete values are calculated from the analog function, $x[T_s]$.
- The discrete (x, t) pairs are plotted using stem plot.
- Repeat for all sampling periods.

Code

```
clear all
clc

xMax = 3;
f = @(t) 4 * cos(2 * pi * t);

sample = [.1, .5, 1];
titles = strcat('Sampling-', num2str(sample', '%2.1f'), 's');
len = length(sample);

for i = [1: len]
    fplot(f, [0, xMax], '-r');
```

```
        hold on;

        Ts = [0: sample(i): xMax];
        xt = f(Ts);

        stem(Ts, xt, 'filled', ...
            'MarkerSize'    ,2.5    , ...
            'LineWidth'     ,0.8    );

        % title(titles(i, :));
        xlabel('time (s)');
        ylabel('x(t)');

        set(gca, ...
            'Box'         ,'off'            , ...
            'TickDir'     ,'out'            , ...
            'YGrid'       ,'on'             , ...
            'XTick'       ,[-10: .5: 10]    , ...
            'YTick'       ,[-12: 2: 12]     , ...
            'FontSize'    ,10               , ...
            'FontName'    ,'Bookman'        , ...
            'YLim'        ,[-4, 4]          );

        axis([-.125, 3.125, -4.25, 4.25]);
        pbaspect([3, 1, 1]);

        print(gcf, [titles(i, :), '.eps'], '-depsc');
        hold off;
    end
```

Input Description

3 Sampling Time Periods are given: 0.1s, 0.5s and 1s.

The range taken is $[0, 3]$, ensuring the analog signal will repeat itself more than once. This will make any pattern in discrete signal clearly visible.

Result

Discussion

The 3 graphs are generated.

It may be observed that information is lost with increasing sampling.

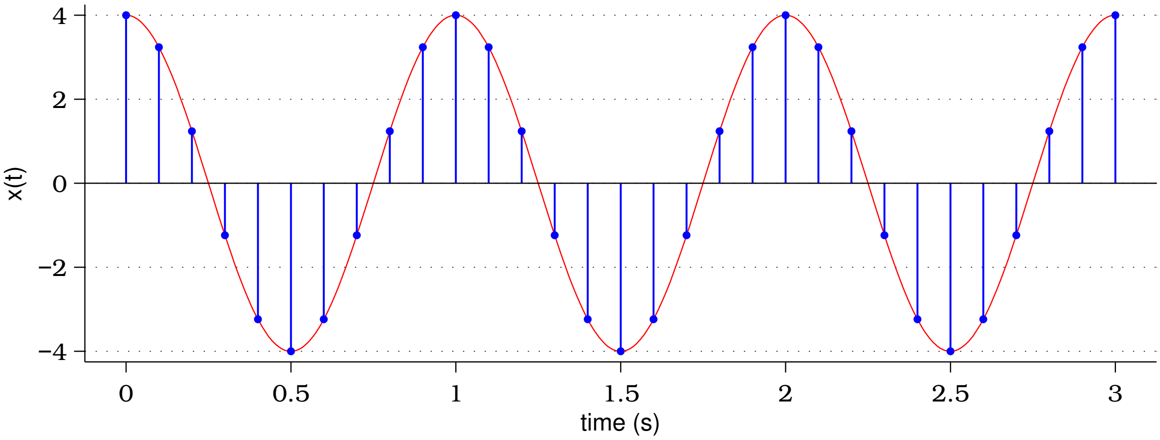

Figure 1.1: *Sampling Period 0.1s*

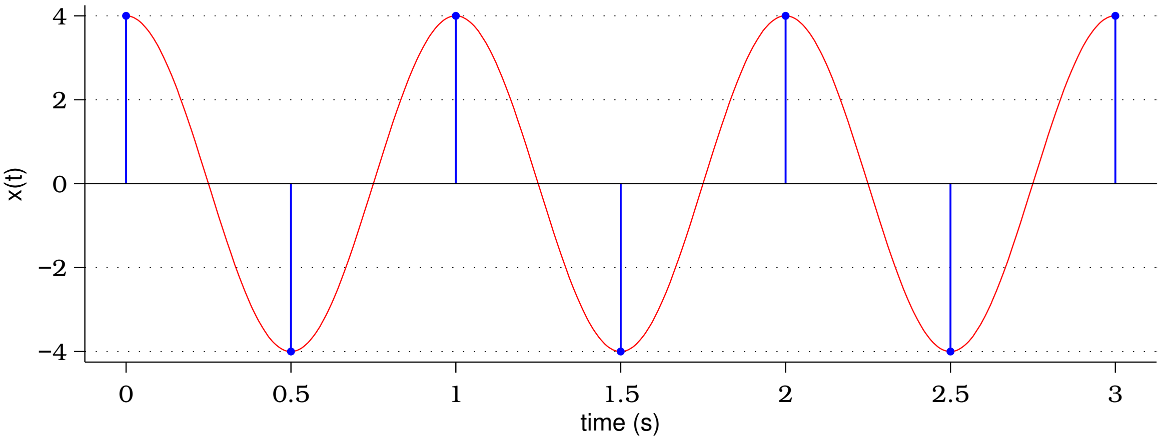

Figure 1.2: *Sampling Period 0.5s*

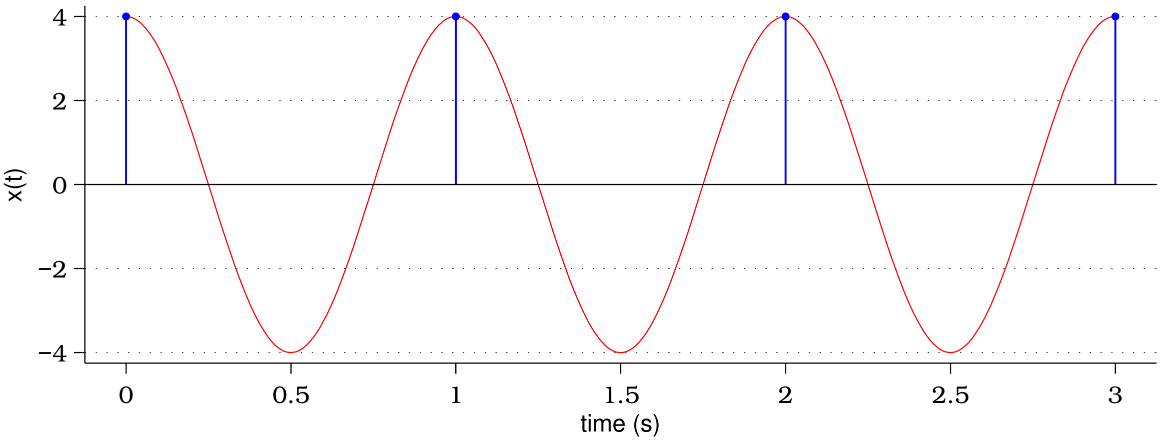

Figure 1.3: *Sampling Period 1.0s*

2. Comparison of Sampling Rates

Aim

To superimpose the 3 discrete-time signals and the original analog signal to compare information loss.

Theoretical Background

Generally, analog signals are sampled using DACs and later reconstructed by filtering the stored discrete-time signal. If the new generated signal differs heavily from original, information is said to be lost.

Hence, it's desirable to keep the no. of sample points as high as possible without consuming too much storage.

Methodology

- 3 sets of discrete points are generated from the sampling peroids.
- Corresponding sets of values are calculated.
- All sets of (x, t) pairs and the analog signal are plotted

Code

```
clear all
clc

xMax = 3;
f = @(t) 4 * cos(2 * pi * t);

sample = [.1, .5, 1];
len = length(sample);
leg = strcat('Sampling-', num2str(sample', '%2.1f'), 's');

colors = 'bmk';
```

```
styles = [' -'; '--'; '-.'];

fplot(f, [0, xMax], '-r');
hold on;

for i = [1: len]
    t = [0: sample(i): 3];
    xt = f(t);

    stem(t, xt, 'filled', ...
        'MarkerSize'    ,2.5                    , ...
        'LineWidth'     ,1.5 + 1.125*sample(i) , ...
        'Color'         ,colors(i)              , ...
        'LineStyle'     ,styles(i, :)           );
end

hold off;

% title('Comparison of Sampling Rates');
xlabel('time (s)');
ylabel('x(t)');

legend('Analog cosine', leg(1, :), leg(2, :), leg(3, :));

set(gca, ...
    'Box'       ,'off'          , ...
    'TickDir'   ,'out'          , ...
    'YGrid'     ,'on'           , ...
    'XTick'     ,[-10: .5: 10]  , ...
    'YTick'     ,[-5: 1: 5]     , ...
    'FontSize'  ,10             , ...
    'FontName'  ,'Times'        );

axis([-.125, 3.125, -4.25, 7.25]);
pbaspect([2, 1, 1]);

print(gcf, 'CompSampleRate.eps', '-depsc');
```

Input Description

3 Sampling Time Periods are given: 0.1s, 0.5s and 1s.

The range taken is $[0, 3]$, ensuring the analog signal will repeat itself more than once. This will make any pattern in discrete signal clearly visible.

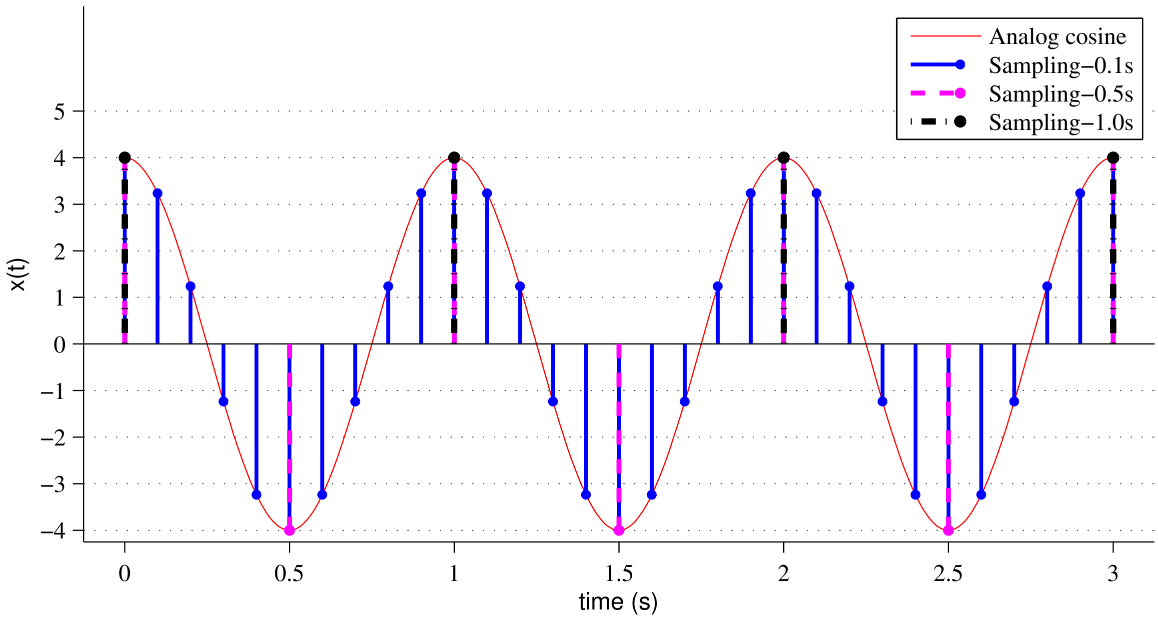

Figure 2.1: *Comparison of Sampling Rates*

Result

Conclusion

Note that, it's a coincidence that the discrete samplings coincide where they do and it is not necessary.

Here, if all 3 discrete signals are filtered to produce an analog signal, only the one with Time Period of 0.1s will resemble the original. The rest will produce a digital clock and a constant signal.

$\therefore$ Only 0.1s discrete-time signal retains the information.

3. Sampling of Analog and Discrete Signals

Aim

a. To compare analog exponenetial $x(t) = e^{at}$ for different values of a.

b. Draw relation between discrete sampling of e^{at} and $y[n] = \alpha^n$.

Theoretical Background

In certain systems, calculating powers of rational numbers is more efficient than powers of irratonal numbers. Hence, finding a tight approximation of the exponenetial function is desirable for efficient calculation in such systems

Methodology

- The continuous graphs of $e^{-\frac{t}{2}}$ and e^{-t} are plotted simultaneously.
- Then, e^{-t} is plotted in another graph, against which multiple graphs of α^n are drawn for comparison.

Code

```
clear all
clc

xrange = [-2, 12];

f1 = @(t) exp(-.5 * t) * (t >= 0);
fplot(f1, xrange, 2e-5, ...
    'Color', 'blue');
hold on;

f2 = @(t) exp(-t) * (t >= 0);
fplot(f2, xrange, 2e-5, ...
    'Color', 'red');
hold off;
```

```
set(findobj(gca, 'Type', 'Line', 'Color', 'blue'), 'LineWidth', 1.25);
set(findobj(gca, 'Type', 'Line', 'Color', 'red'), 'LineWidth', 0.75);

legend('e^{-0.5t}', 'e^{-t}');

% title('Analog Exponential Functions');
xlabel('time (s)');
ylabel('x(t)');

set(gca, ...
    'Box'         ,'off'       , ...
    'TickDir'     ,'out'       , ...
    'YGrid'       ,'on'        , ...
    'XGrid'       ,'off'        , ...
    'XTick'       ,-10:2:30    , ...
    'YTick'       , -5:.25:5);

axis([-2.25, 12.25, -.125, 1.125]);
pbaspect([1.5, 1, 1]);

print(gcf, 'AnalogExpFunc.eps', '-depsc');


Ts = [-2: 12];
xt = exp(-0.5 .* Ts) .* (Ts >= 0);
yn = (0.7 .^ Ts) .* (Ts >= 0);
yn1 = (0.6 .^ Ts) .* (Ts >= 0);
yn2 = (0.5 .^ Ts) .* (Ts >= 0);

plot(Ts, xt, 'Color', 'red', 'Marker', '.');
hold on;
stem(Ts, yn, 'Color', 'blue', ...
    'Marker', 'o', 'LineWidth', 1.1);
stem(Ts, yn1, 'Color', 'black', ...
    'Marker', 'o', 'LineWidth', 1.1, 'LineStyle', '--');
stem(Ts, yn2, 'Color', 'magenta', ...
    'Marker', 'o', 'LineWidth', 1.1, 'LineStyle', '-.');
hold off;

legend('e^{-0.5n}', '0.7^n', '0.6^n', '0.5^n');

% title('Comparison of Exp. and Power Functions');
xlabel('time (s)');
ylabel('x(t)');
```

```
set(gca, ...
    'Box'        ,'off'      , ...
    'TickDir'    ,'out'      , ...
    'YGrid'      ,'on'       , ...
    'XGrid'      ,'off'       , ...
    'XTick'      ,-10:2:30   , ...
    'YTick'      , -5:.25:5);

axis([-2.25, 12.25, -.125, 1.125]);
pbaspect([1.5, 1, 1]);

print(gcf, 'AnalogDiscreteFuncComp.eps', '-depsc');
```

Input Description

The 2 exponenetial plots are for drawn for parameter $a = 0.5, 1$.

Since for $0 < \alpha < 1$, α^n is decreasing, multiple values from $(0, 1)$ were tested, ultimately $\alpha = (0.7, 0.6, 0.5)$ were chosen.

Result

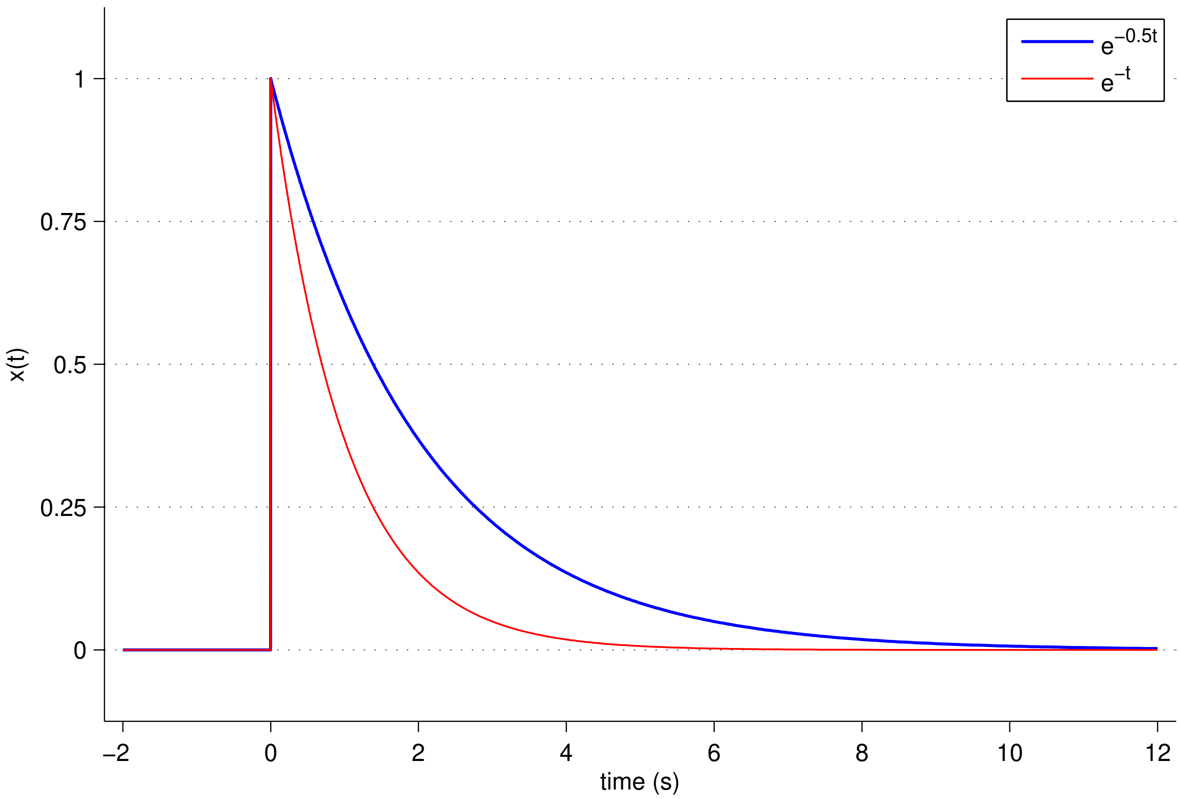

Figure 3.1: *Analog Exponential Functions*

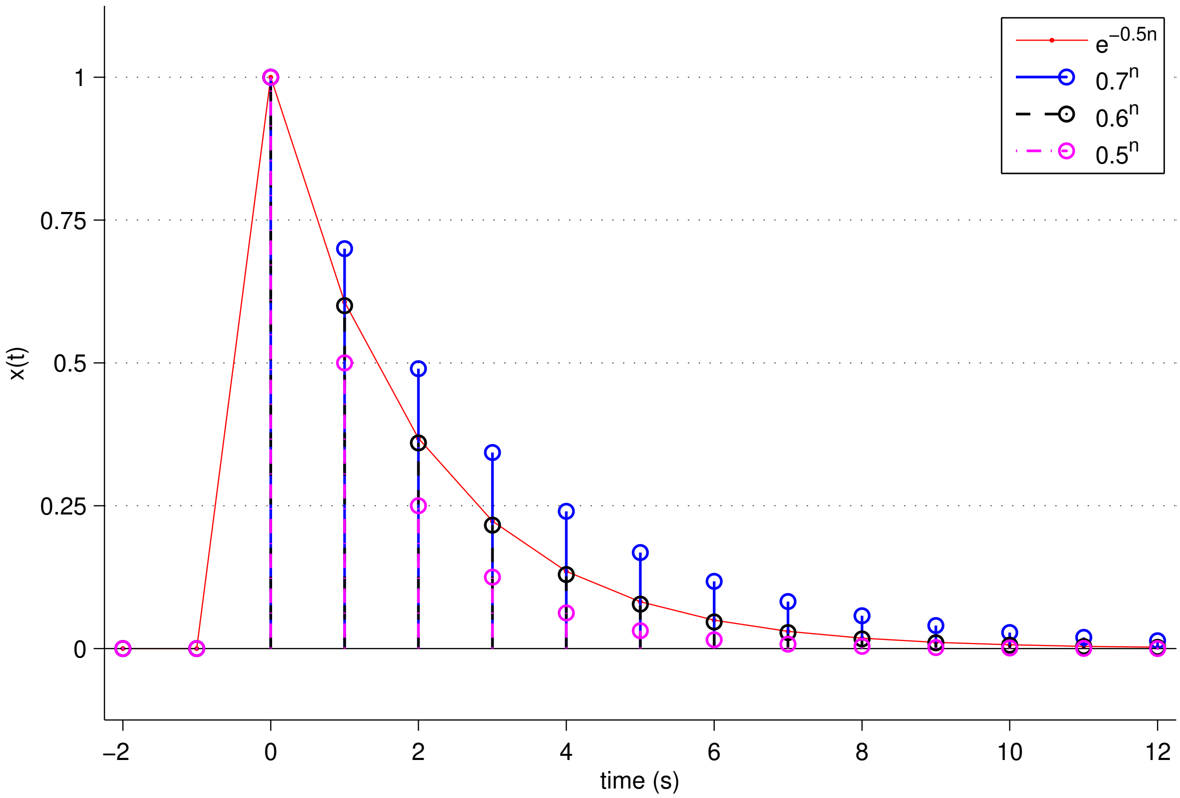

Figure 3.2: *Comparison of Exp. and Power Functions*

Discussion

$y[n]$ creates a tight bound on the exponenetial function $e^{-\frac{n}{2}}$. $y[n] = 0.6^n$ approximates it the best.

Note that, better approximations can be found around 0.6. Hence, an aproximation of any decreasing exponenetial function can be made using suitable conatants.