Experiment 2B

Generation of Analog Signal using IIR Filters

Introduction

There are different techniques to generate sinusoidal signals using Digital Signal Processing. The first technique that crosses from the minds is using the pre-computed sine values stored in a table. An alternative technique would be using a digital oscillator. In the second technique, the frequency, resolution, amplitude and phase of the generated sinusoidal signal can be adjusted by changing the parameters of the iterative function.

In this experiment, the second technique will be used. An Infinite Impulse Response (IIR) Filter will be used for the design of the digital oscillator.

We will be using a technique, namely equation method, to design the IIR filter. The second order difference equation of a system with unit impulse response sin(nwT) can be given by

$$y_{(n)} = A\, y_{(n-1)} + B\, y_{(n-2)} + C\, x_{(n-1)}$$

By means of proper selection of the parameters A, B and C here, it is possible to generate a sinusoidal signal having the desired properties and the sampling period T. Generated sinusoidal signal can be converted to an analog signal which is continuous in time using a Digital to Analog Converter (DAC) and its quantization error can be minimized by applying a low pass filter. As a result, an analog sinusoidal signal can be generated by means of a digital technique.

The cut-off frequency of the output filter which is thought to be generated by using the discrete components (resistance, capacitance, inductance) must be greater than the frequency of the desired sinusoidal signal.

The coefficients of the difference equation given above are:

$$A = 2\cos(\theta_{Degree})$$

$$\theta_{Degree} = (f_{desired}/f_{sampling})*360$$

$$B = -1$$

$$C = \sin(\theta_{Degree})$$

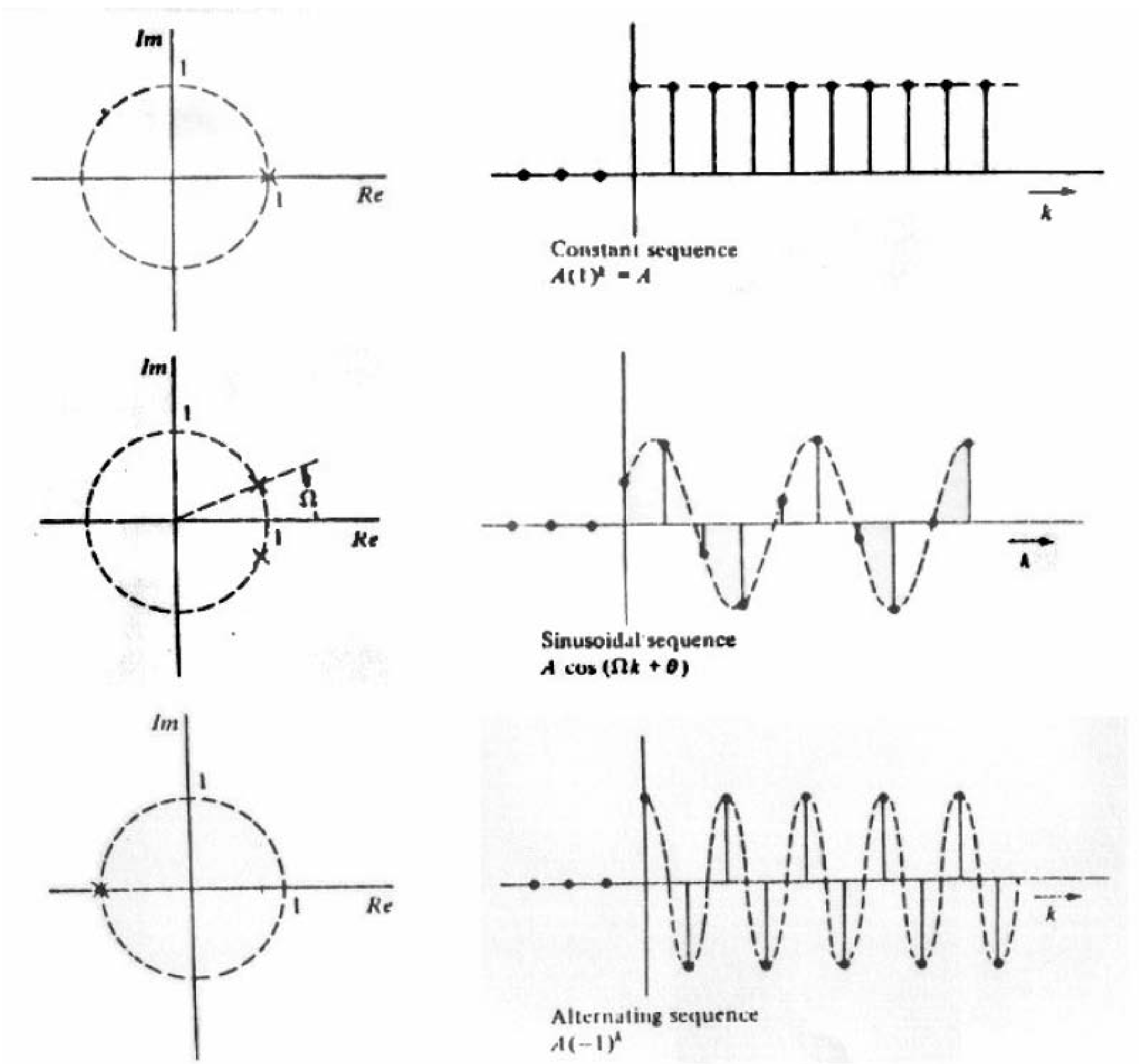
Now, let's analyze how these coefficients are obtained. If we are to take the Z-transform of the difference equation above, the resulting equation would be:

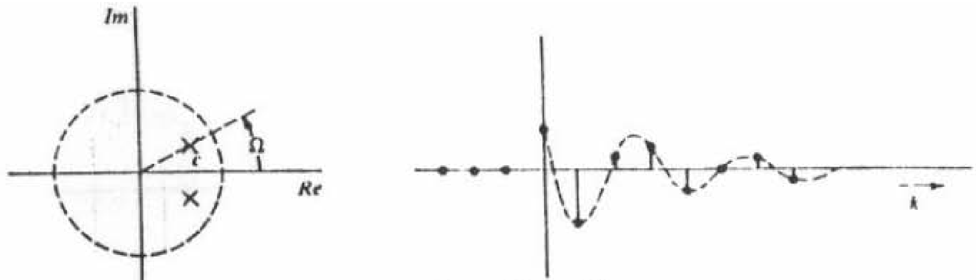
$$Y(Z) = AY(Z)Z^{-1} + BY(Z)Z^{-2} + CX(Z)Z^{-1}$$

Based on the equation above, the transfer function T(Z) would be:

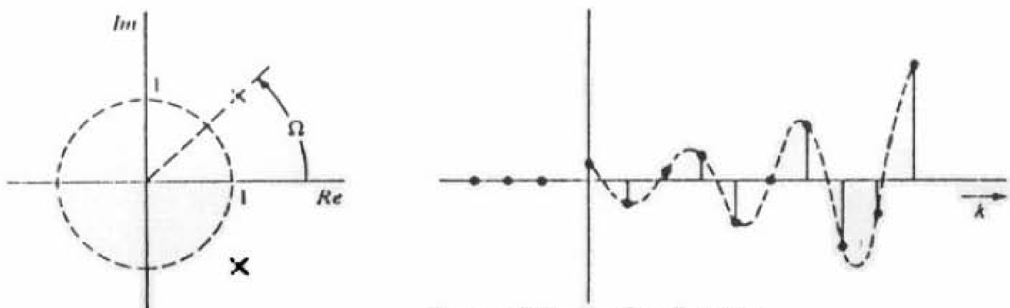
$$T(Z) = Y(Z) / X(Z) = CZ / (Z^2 - AZ - B)$$

The roots of the numerator and denominator are called as ***zeroes*** and ***poles*** respectively. Based on their locations on the identity circle, zeroes and poles change the behavior of the function.

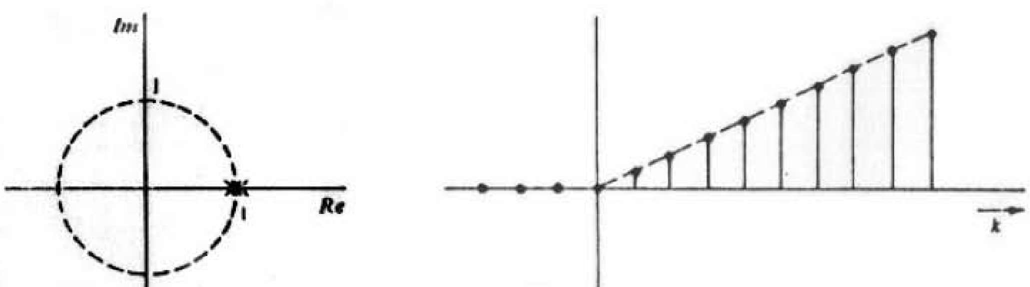




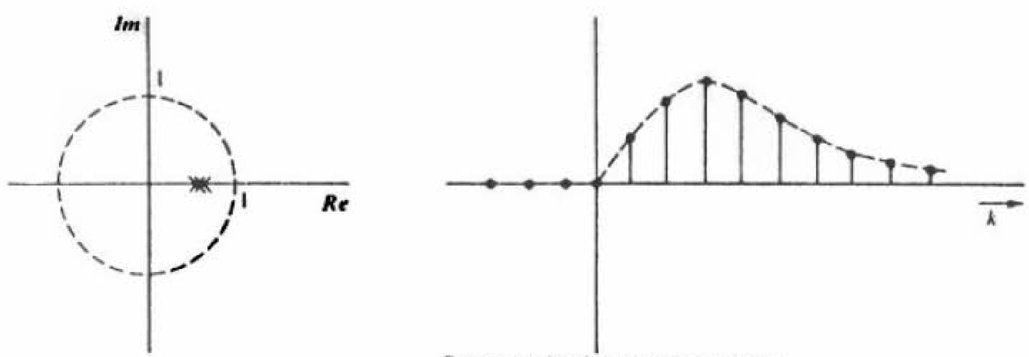
Damped sinusoidal sequence
$Ac^k \cos(\Omega k + \theta)$



Exponentially expanding sinusoidal sequence
$A^k \cos(\Omega k + \theta)$

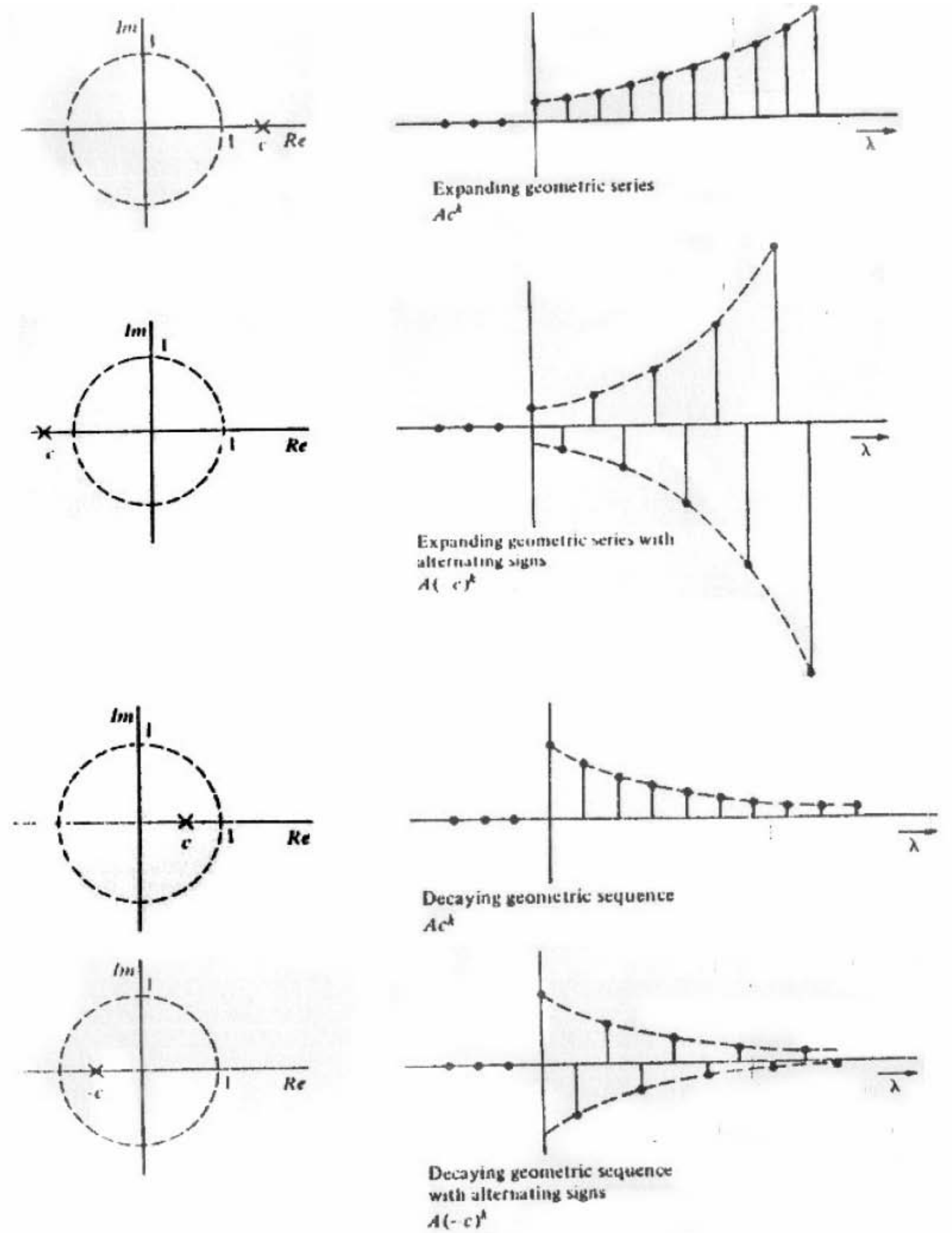


Ramp sequence
$Ak(1)^k = Ak$



Ramp-weighted geometric sequence
Akc^k

3



In order to find the poles of the transfer function T(Z) we make use of the equation which computes the roots of the polynomials with degree of two.

$$X_{1,2} = (-b \pm \sqrt{b^2 - 4ac}\ )/2a$$

This equation is obtained when it is solved for the poles of T(Z). Here, roots have real and imaginary parts. The poles will be symmetrical on the identity circle as the signal we want to generate is a periodic sinusoidal signal.

$$x_{1,2} = \frac{A \pm \sqrt{A^2 + 4B}}{2}$$

$$\mathrm{Re}(x) = A/2 = \cos\theta$$

$$\mathrm{Im}(x) = (\sqrt{A^2 + 4B})/2 = \sin\theta$$

A=2cosθ, B= -1, C=sinθ results are obtained from the equations.

Experiment 2B: Implementation on TMS320C5515

Before the Experiment

1. Remember the contents of the Experiment-2A and how we generate a sinusoidal signal using the second order IIR filter.
2. Review the Sample Project and the files within the project, how we open the sample project, how we debug and program the kit.
3. Conduct a research about how square waves and triangle waves are generated using the sinusoidal waves.

Experiment 2B.1

This part of the experiment aims to generate the sinusoidal signals by means of the second order IIR filters on the kit. Make the following necessary changes on the corresponding files:

***aic3204_test* function:**

```
Int16 aic3204_test( )
{
    SYS_EXBUSSEL = 0x6100;          // Enable I2C bus
    USBSTK5515_I2C_init( );         // Initialize I2C

    /* Codec tests */
    printf( " -> 1 KHz Tone on Headphone (J4).\n" );
    if ( aic3204_tone_headphone( ) )
        return 1;

    USBSTK5515_wait( 100 );  // Wait
  /*  printf( "<-> Audio Loopback from Stereo IN 1 (J3) --> to HP (J4)\n" );
    if ( aic3204_loop_stereo_in1( ) )
        return 1;
*/
    return 0;
}
```

***aic3204_tone_headphone* function:** Note that the calculations here are based on the sinusoidal signal with 100 samples.

```
Int16 output;
float y[] = {0.0,0.0,0.0};
float a = 2*cos(2*pi/100);

float b = -1.0;

y[1] = sin(2*pi/100);

for(;;){

    while((XmitR & I2S0_IR) == 0);
    //wait for transmit interrupt to be pending

    y[0]=a*y[1]+b*y[2];
    y[2]=y[1];
    y[1]=y[0];
    output=(y[0]*pow(2,15));

    I2S0_W0_MSW_W = output ; // 16 bit left channel transmit audio data
    I2S0_W1_MSW_W = output ; // 16 bit right channel transmit audio data

}

/* Disble I2S */
I2S0_CR = 0x00;
```

Make the necessary changes on the sample project and program the kit. Observe the generated sinusoidal signals on oscilloscope.

Experiment 2B.2

A melody can be generated by means of generating the sinusoidal signals with different frequencies in a harmony. Come up with an algorithm to generate a melody and do the necessary changes on the sample project. Program the kit and observe the generated signals on oscilloscope as well as listen the generated signal by means of a headphone. Pay attention not to keep the headphones too close to your ear so as to protect your ear in case of any unexpected noise.

Experiment 2B.3

You are expected to generate square waves and triangle waves using the sinusoidal signals on the DSP kit. With this aim:

- Outline your algorithm which you are going to use to generate square and triangle waves,
- Implement your algorithm within the sample project and program the kit,
- Observe the output of your program on the oscilloscope.

Report

1. Include your observations, the things you learn and analyzed during the experiments,
2. Explain your algorithms which you came up with for the sections 2B.2 and 2B.3.