UNIT 2:
MATHEMATICAL
ENVIRONMENT

BASICDSP

2.1 Introduction

This unit introduces some basic mathematical concepts and relates them to the notation used in the course.

When you have worked through this unit you should:

- appreciate that a mathematical formalism can be expressed in an algorithmic language like Visual Basic as a procedure or function.
- know how array indexing is formalised in the notes and programs
- know how complex numbers arise and how to perform basic arithmetic on complex numbers
- understand the special characteristics of the multiplication of complex numbers when expressed in polar form
- know how to represent polynomials as a coefficient series and as a set of roots

2.2 Relating algebra to algorithms

The mathematical notation used to describe signal processing involves scalars, complex numbers and vectors; it exploits operations of addition, subtraction, multiplication and division, as well as exponents, logarithms, indexing and summation. Each of these elements of mathematical notation have equivalents in the Visual Basic computer language implementation.

Consider the formal expression for the calculation of the roots of a quadratic (the values of x where the equation is zero):

$$ax^2 + bx + c = 0 = (x - x_1)(x - x_2)$$

$$where\ x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a},\ x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

To map such mathematical notation onto a computer program, we need to identify the inputs and constants, the outputs and intermediary values, the operations and procedure of calculation. In this example the inputs are the constants a b c, the outputs x1 x2, the operations + - * / sqrt, and the procedure one of computing the expressions for the roots and their assignment to the variables x1 and x2. With a procedural wrapper:

```
Sub FindQuadRoot(ByVal a As Double, ByVal b As Double, ByVal c As
Double, ByRef x1 As Double, ByRef x2 As Double)

    x1 = (-b + Math.sqrt(b*b - 4*a*c))/(2*a)
    x2 = (-b – Math.sqrt(b*b - 4*a*c))/(2*a)

EndSub
```

Where the notation ByRef indicates the values that are returned to the calling program.

Note that this version will fail for quadratics where $4ac > b^2$.

To manipulate signals as entities, we need to represent them as *vectors*, that is as an array of simple scalar values. We also need to be able to identify single elements (samples) of the vector by subscripting or *indexing*. Although in conventional mathematical notation we might write a vector symbol as $\underline{\mathbf{x}}$, its expansion as $(x_1, x_2, \ldots, x_N)$ and an element of the vector as x_k, in this course we choose:

vector:	x[]
expansion:	x[1], x[2], ..., x[N]
element:	x[k]

This allows a straight-forward mapping to the Visual Basic notation for arrays and array indexing.

Consider the conventional mathematical expression for convolution. This takes two vectors, and returns a third which is a kind of vector product:

$$\underline{z} = \underline{x} * \underline{y}$$

$$\text{where } z_j = \sum_{k=-\infty}^{k=+\infty} x_k y_{j-k}$$

In this course such a formula would be expressed in terms of a procedure that takes signals x[] and y[] and calculates each output sample z[j] using:

$$z[j] = \sum_{k=0}^{\infty} x[k+1]y[j-k]$$

Here we not only simplify the notation but incorporate standard assumptions such as that signals are indexed from time 1, and are zero at earlier times.

This formulation leads to a natural Visual Basic implementation:

```
Public Shared Function Convolve(ByRef x As Waveform, _
                                    ByRef y As Waveform) As Waveform
   Dim z As New Waveform(x.Count + y.Count - 1, x.Rate)
   Dim v As Double

   ' for each output sample
   For j As Integer = 1 To z.Count

       ' sum of product with reversed signal
       v = 0
       For k As Integer = 0 To h.Count() - 1
           v += x(k + 1) * y(j - k)
       Next
       z(j) = v
   Next
   Return z
End Function
```

The only difficult part with this translation is the selection of the upper limit for k in the summation loop, which should not be allowed to access samples before the start of the y[] signal. We also assume that x[] is at least as long as y[].

2.2 Complex numbers

The fact that there are equations such as

$$x^2 + 3 = 0 \qquad x^2 - 10x + 40 = 0$$

which are not satisfied by any real value for x, leads to the introduction of complex numbers. A complex number is an ordered pair of real numbers, usually written in the convenient form x + *i*y, where x and y are real numbers. Complex numbers are subject to rules of arithmetic as defined below. We can of course refer to a complex number by a single algebraical variable, say z:

$$z = x + iy$$

The symbol *i* is called the *imaginary* unit (in engineering texts, it is sometimes referred to as *j*). The number *x* is called the *real part* of the complex number *z* and *y* is called the *imaginary* part of *z*. Thus the solutions for the equation $x^2 - 4x + 13 = 0$, may be written:

$$x^2 - 4x + 13 = 0 = (x - z_1)(x - z_2)$$
$$\text{where } z_1 = 2 + i3,\ z_2 = 2 - i3$$

Complex numbers can be represented as points on a plane where the horizontal or x-axis is called the real axis, and the vertical or y-axis is called the imaginary axis. Complex numbers are then points in a cartesian co-ordinate system on this plane, which is sometimes called the *complex plane*.

Two complex numbers

$$z_1 = x_1 + iy_1 \qquad z_2 = x_2 + iy_2$$

are defined to be *equal* if and only if their real parts are equal and their imaginary parts are equal, that is

$$z_1 = z_2 \text{ if and only if } x_1 = x_2 \text{ and } y_1 = y_2$$

Relational expressions between complex numbers, such as $z_1 < z_2$, have no meaning, although the magnitudes of complex numbers may be compared, see below.

Addition. The sum $z_1 + z_2$ is defined as the complex number obtained by adding the real and imaginary parts of z_1 and z_2, that is

$$z_1 + z_2 = (x_1 + x_2) + i(y_1 + y_2)$$

This addition is like the addition of vectors on the Cartesian plane.

Subtraction. The sum $z_1 - z_2$ is just the inverse of addition, that is

$$z_1 - z_2 = (x_1 - x_2) + i(y_1 - y_2)$$

Multiplication. The product z_1z_2 is defined as the complex number

$$z_1 z_2 = (x_1 + iy_1)(x_2 + iy_2) = (x_1x_2 - y_1 y_2) + i(x_1 y_2 + x_2 y_1)$$

which is obtained by applying the normal rules of arithmetic for real numbers, treating the symbol *i* as a number, and replacing $i^2 = ii$ by -1.

Division. This is defined as the inverse operation of multiplication; that is the quotient z = z_1/z_2 is the complex number z = x + *i*y which satisfies

$$z_1 = z\, z_2$$
$$(x_1 + iy_1) = (x + iy)(x_2 + iy_2)$$

For which a solution may be found by equating real and imaginary parts, assuming that x_2 and y_2 are not both zero:

$$x = \frac{x_1x_2 + y_1 y_2}{x_2^{\ 2} + y_2^{\ 2}} \qquad y = \frac{x_2\, y_1 - x_1\, y_2}{x_2^{\ 2} + y_2^{\ 2}}$$

Expressions. For any complex numbers z_1, z_2, z_3 we have:

$z_1 + z_2 = z_2 + z_1$ and $z_1z_2 = z_2z_1$ *Commutative laws*

$(z_1 + z_2) + z_3 = z_1 + (z_2 + z_3)$ and $(z_1z_2)z_3 = z_1(z_2z_3)$ *Associative laws*

$z_1(z_2 + z_3) = z_1z_2 + z_1z_3$ *Distributive law*

Conjugation. If z = x + *i*y is any complex number, then x - *i*y is called the *conjugate* of z, and is denoted by z*. The product of a complex number with its conjugate is a purely real number:

$$zz^* = (x + iy)(x - iy) = x^2 + y^2$$

Any complex number of the form x + *i*0 is just the real number x. Any complex number of the form 0 + *i*y is called a *pure imaginary number*.

Polar form of complex numbers. If we introduce polar co-ordinates r, θ in the complex plane by setting x = r cos θ, y = r sin θ, then the complex number z = x + *i*y may be written

$$z = r\cos\theta + ir\sin\theta = r(\cos\theta + i\sin\theta)$$

This is known as the polar form of the complex number z. The value r is called the *absolute value* or *modulus* of z, denoted by |z|. Thus

$$|z| = r = \sqrt{x^2 + y^2} = \sqrt{zz^*}$$

The directed angle measured from the positive x axis to the direction of the complex number from the origin of the complex plane is called the *argument* of z, denoted by arg z. Angles are always measured counterclockwise and in radians. Note:

$$arg\ z = \theta = \arcsin\frac{y}{r} = \arccos\frac{x}{r} = \arctan\frac{y}{x}$$

By application of standard addition theorems of trigonometry we find the following relations for the polar form of the product of two complex numbers

$$|z_1 z_2| = |z_1||z_2|$$
$$arg(z_1 z_2) = arg\ z_1 + arg\ z_2$$

That is the magnitude of the product is the product of the input magnitudes, and the argument of the product is the sum of the input arguments. From these we can obtain the important result for the polar form of the powers of a complex number

$$z^n = r^n(\cos\theta + i\sin\theta)^n = r^n(\cos n\theta + i\sin n\theta)$$

This reduces to the so-called *De Moivre formula* for unity magnitude *z*:

$$(\cos\theta + i\sin\theta)^n = \cos n\theta + i\sin n\theta$$

From this we can see that the sequence of powers of a complex number of magnitude 1 are simply a sequence of counter-clockwise rotations by θ around the origin on the complex plane.

Complex exponential. The exponential of a real number x, written e^x or exp x, has the series expansion:

$$\exp x = e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ...$$

The exponential function for complex z = x + *i*y is denoted by e^z and defined in terms of the real functions e^x, cos y and sin y, as follows:

$$e^x = e^{x+iy} = e^x(\cos y + i\sin y)$$

From this we obtain the *Euler formula* for imaginary *z*:

$$e^{iy} = \cos y + i\sin y$$

This in turn leads to the following important identities:

$$|eiy| = \sqrt{\cos^2 y + \sin^2 y} = 1$$
$$e^{2\pi i} = 1$$
$$e^{\pi i} = e^{-\pi i} = -1$$
$$e^{\frac{\pi i}{2}} = i$$
$$e^{\frac{-\pi i}{2}} = -i$$

The complex numbers $e^{i\theta}$, $0<\theta<2\pi$ lie on the unit circle of the complex plane; and the values 1, *i*, -1, -*i* are the points where the unit circle crosses the axes.

2.3 Polynomials

The Z-transform of a signal, which we will meet later, is a way of expressing a time series of samples as a single mathematical object: a polynomial in the variable *z* where the coefficients of the polynomial are simply the sample values. Here we consider some basic operations on polynomials.

In its simplest form a polynomial may be expressed

$$a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$$

where the size of the polynomial *n* is called the *order* of the polynomial, and the constants a_0, a_1, etc are called the polynomial coefficients. Note that this expression may also be written

$$\sum_{j=0}^{n} a_j x^j$$

Operations such as addition, subtraction, multiplication and division may be performed on polynomials by elementary arithmetic and the collection of terms of equivalent powers of x. For multiplication, the product of a polynomial *p* of order *m* with a polynomial *q* of order *n* is a polynomial of order *m+n*. The division of a polynomial of order *m* by a polynomial of order *n* (where $n < m$), leaves a polynomial or order *m-n* and a remainder of degree *n-1*.

Polynomials may be reduced into a product of *factors*; a polynomial *p(x)* of order *n* having *n* factors:

$$p(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n = b_0 (x - b_1)(x - b_2)...(x - b_n)$$

For many purposes it is more convenient to work with the coefficients *b* than the polynomial coefficients *a*. This is because it easy to see that the values *b* are also the values of *x* for which the polynomial has the value 0. Thus the *b* coefficients are also called the *roots* of the polynomial p(x).

A polynomial of order *n* must have *n* roots, but these may not all be real. However if the coefficients of p(x) are real, any complex valued roots must occur in complex conjugate pairs. For polynomials of an order > 4, there exists no formula which directly calculates the values of the roots from the *a* coefficients, and so iterative numerical methods need to be used.

Finally, there is a close relationship between the form of polynomials and the form of *power series* approximations to functions. Here we just give some series expansions of some functions that may be of use later in the course:

$$(1 + x)^a = 1 + ax + \frac{a(a-1)}{2!} x^2 + \frac{a(a-1)(a-2)}{3!} x^3 + ... \text{ where } |x| < 1$$

$$(1 - x)^{-1} = 1 + x + x^2 + x^3 + ... \text{ where } |x| < 1$$

$$(1 - ax)^{-1} = 1 + ax + a^2 x^2 + a^3 x^3 + ... \text{ where } |x| < 1, |a| < 1.$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ ...$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ ...$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ ...$$

Exercise

2.1 Use the `Complex` class to implement a function that will solve arbitrary quadratic equations. In particular, write a program to accept the coefficients a, b and c, and which prints out the values of the two roots in complex notation.