A New Sequence in Signals and Linear Systems
Part I: ENEE 241

Adrian Papamarcou
Department of Electrical and Computer Engineering
University of Maryland, College Park

Draft 8, 01/24/07

© Adrian Papamarcou 2007

Contents

1 Numbers, Waveforms and Signals **1**

1.1 Real Numbers in Theory and Practice 2

1.1.1 Fixed-Point Representation of Real Numbers 2

1.1.2 Floating-Point Representation 3

1.1.3 Visualization of the Fixed and Floating Point Representations 4

1.1.4 Finite-Precision Representations of Real Numbers . . 5

1.1.5 The IEEE Standard for Floating-Point Arithmetic . . 6

1.2 Round-off Errors 8

1.2.1 Error Definitions 8

1.2.2 Round-Off 8

1.2.3 Round-Off Errors in Finite-Precision Computation . . 10

1.3 Review of Complex Numbers 13

1.3.1 Complex Numbers as Two-Dimensional Vectors . . . 13

1.3.2 The Imaginary Unit 15

1.3.3 Complex Multiplication and its Interpretation 15

1.3.4 The Complex Exponential 16

1.3.5 Inverses and Division 18

1.4 Sinusoids in Continuous Time 20

1.4.1 The Basic Cosine and Sine Functions 20

1.4.2 The General Real-Valued Sinusoid 21

1.4.3 Complex-Valued Sinusoids and Phasors 23

1.5 Sinusoids in Discrete Time 27

1.5.1 Discrete-Time Signals 27

1.5.2 Definition of the Discrete-Time Sinusoid 27

1.5.3 Equivalent Frequencies for Discrete-Time Sinusoids . . 29

1.5.4 Periodicity of Discrete-Time Sinusoids 32

1.6 Sampling of Continuous-Time Sinusoids 34

1.6.1 Sampling 34

1.6.2 Discrete-Time Sinusoids Obtained by Sampling 35
1.6.3 The Concept of Alias 36
1.6.4 The Nyquist Rate 39
Problems for Chapter 1 42

2 Matrices and Systems 49
2.1 Introduction to Matrix Algebra 50
2.1.1 Matrices and Vectors 50
2.1.2 Signals as Vectors 52
2.1.3 Linear Transformations 54
2.2 Matrix Multiplication 56
2.2.1 The Matrix of a Linear Transformation 56
2.2.2 The Matrix-Vector Product 57
2.2.3 The Matrix-Matrix Product 58
2.2.4 Associativity and Commutativity 59
2.3 More on Matrix Algebra 62
2.3.1 Column Selection and Permutation 62
2.3.2 Matrix Transpose 63
2.3.3 Row Selection 65
2.4 Matrix Inversion and Linear Independence 66
2.4.1 Inverse Systems and Signal Representations 66
2.4.2 Range of a Matrix 67
2.4.3 Geometrical Interpretation of the Range 68
2.5 Inverse of a Square Matrix 72
2.5.1 Nonsingular Matrices 72
2.5.2 Inverse of a Nonsingular Matrix 73
2.5.3 Inverses in Matrix Algebra 75
2.5.4 Nonsingularity of Triangular Matrices 76
2.5.5 The Inversion Problem for Matrices of Arbitrary Size 77
2.6 Gaussian Elimination 79
2.6.1 Statement of the Problem 79
2.6.2 Outline of Gaussian Elimination 79
2.6.3 Gaussian Elimination Illustrated 81
2.6.4 Row Operations in Forward Elimination 83
2.7 Factorization of Matrices in Gaussian Elimination 84
2.7.1 LU Factorization 84
2.7.2 The Matrix **L** 85
2.7.3 Applications of the LU Factorization 88
2.7.4 Row Operations in Backward Substitution 89
2.8 Pivoting in Gaussian Elimination 92

2.8.1 Pivots and Pivot Rows 92
2.8.2 Zero Pivots 92
2.8.3 Small Pivots 93
2.8.4 Row Pivoting 95
2.9 Further Topics in Gaussian Elimination 98
2.9.1 Computation of the Matrix Inverse 98
2.9.2 Ill-Conditioned Problems 100
2.10 Inner Products, Projections and Signal Approximation 102
2.10.1 Inner Products and Norms 102
2.10.2 Angles, Projections and Orthogonality 103
2.10.3 Formulation of the Signal Approximation Problem . . 105
2.10.4 Projection and the Signal Approximation Problem . . 106
2.10.5 Solution of the Signal Approximation Problem 108
2.11 Least-Squares Approximation in Practice 111
2.12 Orthogonality and Least-Squares Approximation 116
2.12.1 A Simpler Least-Squares Problem 116
2.12.2 Generating Orthogonal Reference Vectors 117
2.13 Complex-Valued Matrices and Vectors 122
2.13.1 Similarities to the Real-Valued Case 122
2.13.2 Inner Products of Complex-Valued Vectors 123
2.13.3 The Least-Squares Problem in the Complex Case . . . 126
Problems for Chapter 2 128

3 The Discrete Fourier Transform 144
3.1 Complex Sinusoids as Reference Vectors 145
3.1.1 Introduction 145
3.1.2 Fourier Sinusoids 146
3.1.3 Orthogonality of Fourier Sinusoids 148
3.2 The Discrete Fourier Transform and its Inverse 153
3.2.1 Definitions 153
3.2.2 Symmetry Properties of the DFT and IDFT Matrices 155
3.2.3 Signal Structure and the Discrete Fourier Transform . 158
3.3 Structural Properties of the DFT: Part I 163
3.3.1 The Spectrum of a Real-Valued Signal 163
3.3.2 Signal Permutations 166
3.4 Structural Properties of the DFT: Part II 170
3.4.1 Permutations of DFT and IDFT Matrices 170
3.4.2 Summary of Identities 171
3.4.3 Main Structural Properties of the DFT 172
3.5 Structural Properties of the DFT: Examples 176

3.5.1 Miscellaneous Signal Transformations 176
3.5.2 Exploring Symmetries 177
3.6 Multiplication and Circular Convolution 182
3.6.1 Duality of Multiplication and Circular Convolution . . 182
3.6.2 Computation of Circular Convolution 185
3.7 Periodic and Zero-Padded Extensions of a Signal 187
3.7.1 Definitions 187
3.7.2 Periodic Extension to an Integer Number of Periods . 188
3.7.3 Zero-Padding to a Multiple of the Signal Length . . . 191
3.8 Detection of Sinusoids Using the Discrete Fourier Transform . 192
3.8.1 Introduction 192
3.8.2 The DFT of a Complex Sinusoid 193
3.8.3 Detection of a Real-Valued Sinusoid 197
3.8.4 Final Remarks 199
Problems for Chapter 3 201

4 Introduction to Linear Filtering 215
4.1 The Discrete-Time Fourier Transform 216
4.1.1 From Vectors to Sequences 216
4.1.2 Development of the Discrete-Time Fourier Transform 217
4.2 Computation of the Discrete-Time Fourier Transform 220
4.2.1 Signals of Finite Duration 220
4.2.2 The DFT as a Sampled DTFT 223
4.2.3 Signals of Infinite Duration 225
4.3 Introduction to Linear Filters 229
4.3.1 Linearity, Time-Invariance and Causality 229
4.3.2 Sinusoidal Inputs and Frequency Response 231
4.3.3 Amplitude and Phase Response 233
4.4 Linear Filtering in the Frequency Domain 236
4.4.1 System Function and its Relationship to Frequency Response 236
4.4.2 Cascaded Filters 238
4.4.3 The General Finite Impulse Response Filter 239
4.4.4 Frequency Domain Approach to Computing Filter Response 241
4.4.5 Response to a Periodic Input 243
4.5 Frequency-Selective Filters 247
4.5.1 Main Filter Types 247
4.5.2 Ideal Filters 248
4.5.3 Amplitude Response of Practical Filters 249

4.5.4 Practical FIR Filters 251
4.5.5 Filter Transformations 254
4.6 Time Domain Computation of Filter Response 256
4.6.1 Impulse Response 256
4.6.2 Convolution and Linearity 258
4.6.3 Computation of the Convolution Sum 259
4.6.4 Convolution and Polynomial Multiplication 262
4.6.5 Impulse Response of Cascaded Filters 264
4.7 Convolution in Practice 266
4.7.1 Linear Convolution of Two Vectors 266
4.7.2 Linear Convolution versus Circular Convolution 267
4.7.3 Circular Buffers for Real-Time FIR Filters 270
4.7.4 Block Convolution 271
Problems for Chapter 4 274

Epilogue **284**
The Fourier Series 285
The Fourier Transform 287
Nyquist Sampling 290

Chapter 1

Numbers, Waveforms and Signals

1.1 Real Numbers in Theory and Practice

The real line **R** consists of a continuum of points, each point representing a different number. A real number can be

- *rational*, i.e., a ratio m/n of two integers, where $(n \neq 0)$; or
- *irrational*; examples of irrational numbers include π, $\sqrt{2}$, etc.

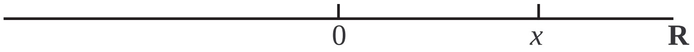


Figure 1.1: The real line.

1.1.1 Fixed-Point Representation of Real Numbers

This is the most natural representation of a real number x. The sign of x (also denoted by $\mathrm{sgn}(x)$) is followed by the *integer* part of x, the (fixed) fractional point, and the *fractional* part of x.

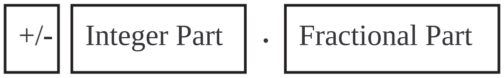


Figure 1.2: Fixed-point representation of a real number. Fractional part may consist of infinitely many digits.

The representation uses a *base* (or *radix*) B, which is a positive integer. All digits (in both the integer and fractional parts of x) are integers ranging from 0 to $B-1$, inclusive. For example,

- in the *decimal* representation of x, the base $B = 10$ is used, and the possible digits are $0, 1, \ldots, 9$;
- in the *binary* representation of x, the base $B = 2$ is used, and the possible digits are 0 and 1, only.

The integer part of x is always a finite string of digits. The fractional part is, in general, an infinite string; in other words, *non-terminating* expansions are the rule rather than the exception. For example $(B = 10)$:

$$\pi = 3.1415926\ldots \qquad \sqrt{2} = 1.4142136\ldots$$

The rational part of x has a *terminating* expansion if and only if x is an exact multiple of a negative power of B (i.e., it equals B^{-l}, where l is integer). Interestingly, terminating expansions can be expressed in two equivalent non-terminating forms. For example ($B = 10$):

$$13.75 = 13.7500000\ldots = 13.7499999\ldots$$

Clearly, decimal ($B = 10$) representations are most common. Binary forms ($B = 2$) are equally (if not more) useful in machine computations. Conversion to binary form involves expressing the integer and fractional parts of x in terms of nonnegative and negative powers of 2, respectively. For example,

$$\begin{aligned} 13 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (1101)_2 \\ 0.75 &= 1 \times 2^{-1} + 1 \times 2^{-2} = (0.11)_2 \end{aligned}$$

and thus

$$13.75 = (1101.11)_2 = (1101.1100000\ldots)_2 = (1101.1011111\ldots)_2$$

where both non-terminating expansions are shown. Note that for a positive integer l, 2^{-l} can be expressed as $5^l \times 10^{-l}$. Thus, if the binary representation of x has a terminating fractional part, so does the decimal one. The converse is not, however, true; e.g., the decimal fraction 0.6 does not have a terminating binary expansion.

1.1.2 Floating-Point Representation

A floating-point representation of x with respect to base B is usually specified as

$$\text{sgn}(x) \times (0.d_1 d_2 d_3 \ldots) \times B^E$$

where:

- the fraction $0.d_1 d_2 d_3 \ldots$ is known as the *mantissa* of x; and
- E is the *exponent* of the base B.

The integer factor B^E is also known as the *characteristic* of the floating-point representation. Note that the floating-point representation of x is not unique. For example, $x = 80$ can be written as

$$0.8 \times 10^2 = 0.08 \times 10^3 = 0.008 \times 10^4 = \ldots$$

The floating point representation can be made unique by *normalization*, i.e., requiring that d_1 be nonzero. Thus the normalized floating-point representation of $x = 80$ is $x = 0.8 \times 10^2$. Similarly,

$$\begin{aligned}\pi &= (0.31415926\ldots) \times 10^1 \\ 13.75 &= 0.1375 \times 10^2 = (0.110111)_2 \times 2^4 \\ 0.01375 &= 0.1375 \times 10^{-1}\end{aligned}$$

Note that in the binary ($B = 2$) case, normalization forces the first fractional digit d_1 to be equal to 1.

1.1.3 Visualization of the Fixed and Floating Point Representations

Assume a fixed base B. If, in the fixed-point representation of x, we keep the sign and integer part of x fixed and vary the fractional part, the value of x will range between two consecutive integers. In other words, the integer part of x corresponds to a unit-length interval on the real line, and such intervals form a partition of the real line as shown in Figure 1.3.

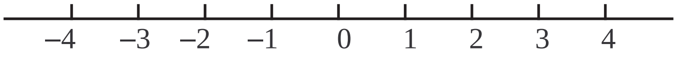


Figure 1.3: Partition of the real line into intervals of unit length.

In the normalized floating-point representation of x, keeping both the sign and exponent E of x fixed and varying the mantissa also results in x ranging over an interval of values. Since $d_1 \geq 1$, the range of $0.d_1d_2d_3\ldots$ is the interval $[1/B,\, 1]$, shown in bold in Figure 1.4.

0
1/2
1
B=2
0
1/10
1
B=10

Figure 1.4: Shown in bold are the ranges of values of the normalized mantissa of binary (left) and decimal (right) numbers.

Thus positive numbers with the same exponent E form the interval $[B^{E-1},\, B^E]$. This means that varying E results in a partition of the positive

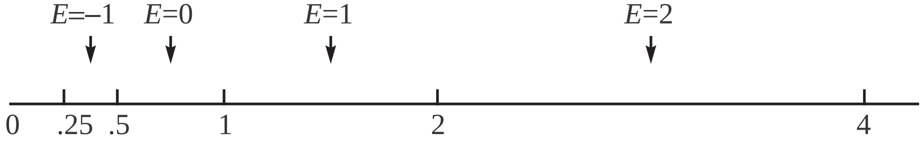


Figure 1.5: Partition of the real line according to the exponent E in the binary floating-point representation.

real line into intervals of *different* length—the ratio of interval lengths being a power of B. The binary case is illustrated in Figure 1.5.

Clearly, the same holds for the negative real line. Note that $x = 0$ is the only number that does not have a normalized representation (all the d_i's are zero).

1.1.4 Finite-Precision Representations of Real Numbers

Machine computations require finite storage and finite processing time. If K bits are allocated to storing a real variable, then that variable can take at most 2^K values. These values will be collectively termed as the *precise* numbers. In the course of a numerical computation, every real number is approximated by a precise number. This approximation is known as *roundoff* (more on that later).

Fixed-point computation involves, essentially, integer arithmetic. The 2^K precise numbers are uniformly spaced on an interval of length A. In a typical application, a precise number would be an exact multiple of $A \cdot 2^{-K}$ ranging in value from $-(A/2) + A \cdot 2^{-K}$ to $A/2$.

Although fixed-point computation is relatively simple and can be attractive for low-power applications, it can lead to large relative errors when small values are involved. *Floating-point* computation, on the other hand, is generally more accurate and preferable for most applications.

Precise numbers in floating-point computation are determined by assigning K_e bits to the exponent and K_m bits to the mantissa, where

$$1 + K_e + K_m = K$$

(One bit is required for encoding the sign of the number.) The 2^{K_e} possible exponents are consecutive integers; this places the positive precise numbers in K_e adjacent intervals of the form $[2^{E-1}, 2^E)$. Within each such interval, there are 2^{K_m} uniformly spaced precise numbers, each corresponding to a different value of the mantissa. Details of an actual implementation (e.g., in MATLAB) are described below.

1.1.5 The IEEE Standard for Floating-Point Arithmetic

MATLAB uses what is known as *double-precision* variables in its floating-point computations. The precise numbers and their encoding follow the IEEE 754 Standard. $K = 64$ bits are used for each precise number, with $K_e = 11$ bits allocated to the exponent and $K_m = 52$ bits allocated to the mantissa. The bits are arranged as shown in Figure 1.6:

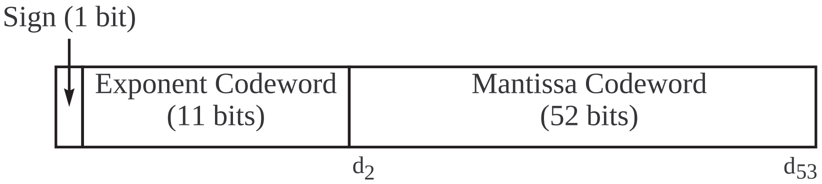


Figure 1.6: Binary encoding of a double-precision floating-point number.

The value of the sign bit is 0 for positive numbers, 1 for negative ones.

Read as a binary integer with the most significant digit on the left (as usual), the exponent codeword ranges in value from $C = 0$ to $C = 2^{11} - 1 = 2047$. The precise number a is then obtained as follows:

- If $1 \leq C \leq 2046$, then a has a *normalized* floating-point representation with exponent $E = C - 1022$ and mantissa given by $0.1d_2 \ldots d_{53}$, where $d_2, \ldots, d_{53}$ are read directly off the mantissa codeword in the order shown in Figure 1.6.

$$a = \mathrm{sgn}(a) \times (0.1d_2 \ldots d_{53}) \times 2^{C-1022}$$

- If $C = 0$, then a has a *denormalized* floating-point representation:

$$a = \mathrm{sgn}(a) \times (0.0d_2 \ldots d_{53}) \times 2^{-1021} = \mathrm{sgn}(a) \times (0.d_2 \ldots d_{53}) \times 2^{-1022}$$

  Note that $d_2 = \ldots = d_{53} = 0$ gives $a = 0$.

- If $C = 2047$, the value of a is either infinite (`Inf`) or indeterminate (`NaN`). An infinite value (positive or negative) is encoded using an all-zeros mantissa. Any other mantissa word denotes an indeterminate value (such values arise from expressions such as $0/0$, $\infty - \infty$, etc.).

The effective range of exponents in the normalized floating-point representation is $E = -1021$ to $E = 1024$. As discussed earlier, each exponent E corresponds to a range of 2^{52} precise numbers uniformly spaced in $[2^{E-1}, 2^{E})$ (on the positive real line). Denormalization also allows 2^{52} uniformly spaced precise numbers in the interval $[0, 2^{-1022})$.

MATLAB can display the binary string corresponding to any floating point number (after approximating it by the nearest precise number); the hexadecimal form is used (0000 through 1111 are encoded as `0` through `F`). The command

```
format hex
```

causes all subsequent numerical values to be shown in that form. Conversely, the command

```
hex2num('string')
```

where *string* consists of 16 hex digits (= 64 bits), will output the precise number corresponding to the input string. Thus:

- `hex2num('0000000000000001')` gives the smallest positive number, namely 2^{-1074}, which is approximately equal to 4.94×10^{-324};

- `hex2num('7FEFFFFFFFFFFFFF')` gives the largest positive number, namely $2^{1024} - 2^{971}$, which is approximately equal to 1.80×10^{308}.

1.2 Round-off Errors

1.2.1 Error Definitions

The *error* in approximating a number x by $\hat{x}$ is given by the difference

$$\hat{x} - x$$

The *relative error* is given by the ratio

$$\frac{\hat{x} - x}{x}$$

and is often quoted as a percentage. We will use the term "absolute" (error, relative error) to designate the absolute value of the quantities defined above.

Word of Caution: The term *absolute error* is sometimes used for the (signed) error $\hat{x} - x$ itself, to differentiate it from the relative error. We will not adopt this definition here.

1.2.2 Round-Off

Round-off is defined as the approximation of a real number by a finite-precision number. This approximation is necessary in most machine computations involving real numbers, due to the storage constraints (finite word lengths for all variables involved).

The most common round-off methods are *chopping* and *rounding.*

Chopping

Chopping is the approximation of a real number x by the closest precise number whose absolute value is less than, or equal to, that of x (i.e., the approximation is in the direction of the origin). Chopping is illustrated in Figure 1.7.

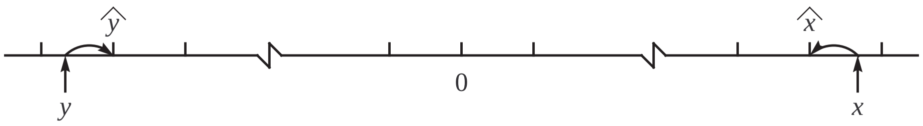


Figure 1.7: Chopping ($\hat{\ }$) of positive and negative numbers.

As an example, consider a *decimal fixed-point* machine with four-digit precision. This means that the precise numbers are all multiples of 10^{-4} that fall in a range $(-A/2, A/2]$, where A is a large number. Assuming that

x falls in the above range, chopping x is the same as keeping the first four fractional digits in the fixed-point decimal representation of x.

Thus, denoting chopping by "→", we have:

$$\begin{aligned} \pi &= 3.1415926\ldots &\rightarrow& \quad 3.1415 \\ 1-\sqrt{2} &= -0.4142136\ldots &\rightarrow& \quad -0.4142 \end{aligned}$$

The same procedure would apply to a *decimal floating-point* machine with (the same, for the sake of comparison) four-digit precision. In this case, four-digit precision refers to the decimal expansion of the *mantissa.* Thus, the same numbers would be chopped as follows:

$$\begin{aligned} \pi &= (0.31415926\ldots)\times 10^1 &\rightarrow& \quad 0.3141\times 10^1 \\ 1-\sqrt{2} &= (-0.4142136\ldots)\times 10^0 &\rightarrow& \quad -0.4142\times 10^0 \end{aligned}$$

The *significant digits* of a number begin with the leftmost nonzero digit in its fixed-point representation; that digit (known as the *most significant* digit) can be on either side of the fractional point. Thus, for example, the significant digits of π are:

$$31415926...$$

The significant digits are the same as the fractional digits in the *normalized* floating-point representation.

Note that in the case of π, the floating-point machine produced one less significant digit for π than the fixed-point machine. This is consistent with the fact that fixed-point computation results in errors $\hat{x}-x$ that are roughly of the same order of magnitude regardless of the order of magnitude of x; thus the relative error improves as the order of magnitude of x increases. Floating-point computation, on the other hand, maintains the same order of magnitude for the *relative* error throughout the range of values of x for which a normalized representation is possible; while the expected *actual* (i.e., not relative) error increases with $|x|$.

Rounding

Rounding is the approximation of x by the closest precise number. If x is midway between two precise numbers, then the number farther from the origin is chosen (i.e., the number with the larger absolute value). Rounding is illustrated in Figure 1.8.

In the fixed-point, four decimal-digit precision example discussed earlier, rounding (again denoted by "→") gives

$$\begin{aligned} \pi &= 3.1415926\ldots &\rightarrow& \quad 3.1416 \\ 1-\sqrt{2} &= -0.4142136\ldots &\rightarrow& \quad -0.4142; \end{aligned}$$

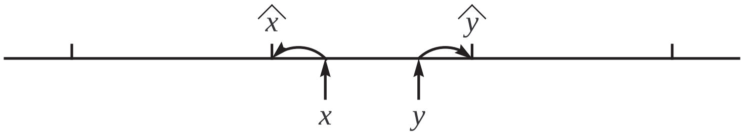


Figure 1.8: Rounding ($\hat{\ }$).

while in the floating-point case,

$$\begin{aligned} \pi &= (0.31415926\ldots) \times 10^1 \quad \rightarrow \quad 0.3142 \times 10^1 \\ 1 - \sqrt{2} &= (-0.4142136\ldots) \times 10^0 \quad \rightarrow \quad -0.4142 \times 10^0 \end{aligned}$$

Note that in the case of π, rounding gave a different answer from chopping. Rounding is preferable, since it *always* results in a lower absolute error value $|\hat{x} - x|$ than does chopping. The error resulting from chopping is also biased: it is always negative for positive numbers and positive for negative numbers. This can lead to very large cumulative errors in situations where one sign ($+$ or $-$) dominates, e.g., adding a large number of positive variables scaled by positive coefficients.

1.2.3 Round-Off Errors in Finite-Precision Computation

Fact. *Round-off errors are ubiquitous.*

Example 1.2.1. Suppose we enter

```
x = 0.6
```

in MATLAB. In decimal form, this is an exact fraction (6/10), which is far easier to specify numerically than, say, π. Yet 6/10 does not have a terminating binary expansion, i.e., it is not an exact multiple of 2^{-l} for some integer l. As a result, MATLAB will round 0.6 to 53 significant binary digits (equivalent to roughly 16 decimal digits). The value $x = 6/10$ falls in the range $[1/2,\ 1)$, where the spacing between precise numbers equals 2^{-53}. It follows that the round-off error in representing x will be less than 2^{-54} in absolute value, but it won't be zero. □

Fact. *The order of computation affects the accumulation of round-off errors. As a rule of thumb, summands should be ordered by increasing absolute value in floating-point computation.*

Example 1.2.2. Assume a decimal floating-point machine with four digit precision. At every stage of the computation (including input and output), every number x is rounded to the form

$$\mathrm{sgn}(x) \times (0.d_1 d_2 d_3 d_4) \times 10^E$$

where d_i are decimal digits such that $d_1 > 0$ (for normalization), and E is assumed unrestricted.

Suppose that we are asked to compute

$$x_0 + x_1 + \ldots + x_{10}$$

where $x_0 = 1$ and $x_1 = \ldots = x_{10} = 0.0001$. Note that the ratio of x_0 to each of the other numbers is 10^{-4}, which corresponds to four decimal digits. Also, each of the eleven numbers corresponds to a precise number, i.e., no round-off error is incurred in representing $x_0, \ldots, x_{10}$.

Clearly, the exact answer is 1.001, which is the precise four-digit floating-point number 0.1001×10^1. If we perform the summation in the natural order (i.e., that of the indices), then we have:

$$\begin{array}{rcrcl} x_0 + x_1 & = & 1.0001 & \rightarrow & 0.1000 \times 10^1 \\ 0.1000 \times 10^1 + x_2 & = & 1.0001 & \rightarrow & 0.1000 \times 10^1 \\ \vdots & \vdots & \vdots & & \\ 0.1000 \times 10^1 + x_{10} & = & 1.0001 & \rightarrow & 0.1000 \times 10^1 \end{array}$$

Thus at each stage, four-digit precision produces the same result as adding zero to x_0. The final result is clearly unsatisfactory (even though the relative error is only about 10^{-3}).

The exact answer can be obtained by summing x_1 through x_{10} first, and then adding x_0:

$$\begin{array}{rcl} x_1 + x_2 & = & 0.0002 = 0.2000 \times 10^{-3} \\ 0.2000 \times 10^{-3} + x_3 & = & 0.0003 = 0.3000 \times 10^{-3} \\ \vdots & \vdots & \vdots \\ 0.9000 \times 10^{-3} + x_{10} & = & 0.0010 = 0.1000 \times 10^{-2} \\ 0.1000 \times 10^{-2} + x_0 & = & 0.1001 \times 10^1 \end{array}$$

Note that no rounding was necessary at any point in the above computation.

□

Fact. *Computing the difference of two quantities that are close in value may result in low precision. If higher precision is sought, a different algorithm may be required.*

Example 1.2.3. Consider the computation of $1 - \cos x$ for $x = 0.02$ on a floating-point machine with three-digit precision. The exact answer is

$$1 - \cos(0.02) = 1 - (0.999800007\ldots) = (0.1999933...) \times 10^{-3}$$

If $\cos(0.2)$ is first computed correctly and then rounded to three significant digits, the resulting answer is

$$0.1000 \times 10^1 - 0.1000 \times 10^1 = 0$$

which is clearly unsatisfactory. A different algorithm is needed here.

The Taylor series expansion for $\cos x$ gives

$$\begin{aligned} \cos x &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots \\ 1 - \cos x &= \frac{x^2}{2!} - \frac{x^4}{4!} + \frac{x^6}{6!} - \ldots \end{aligned}$$

Evaluating the first three terms in the second expansion with three-digit precision, we obtain:

$$\begin{aligned} x^2/2 &= 0.200 \times 10^{-3} \\ x^4/24 &= 0.667 \times 10^{-8} \\ x^6/720 &= 0.889 \times 10^{-13} \end{aligned}$$

It is possible to estimate, using the remainder form of Taylor's theorem, the error involved in approximating $1 - \cos x$ by the first three terms in the expansion. That estimate turns out to be much less than the smallest of the three terms. Since the ratio between consecutive terms is much larger than 10^3, summing with 3-digit precision will result in an answer equal to the first term, namely 0.200×10^{-3}; while the error will be dominated by the second term. The relative error is thus roughly $(0.667 \times 10^{-8})/(0.200 \times 10^{-3}) = 3.33 \times 10^{-5}$, which is very satisfactory considering the low precision involved in the computation. □

1.3 Review of Complex Numbers

Complex numbers are especially important in signal analysis because they lead to an efficient representation of sinusoidal signals—arguably the most important class of signals.

1.3.1 Complex Numbers as Two-Dimensional Vectors

A complex number z is a two-dimensional vector, or equivalently, a point (x, y) on the Cartesian plane:

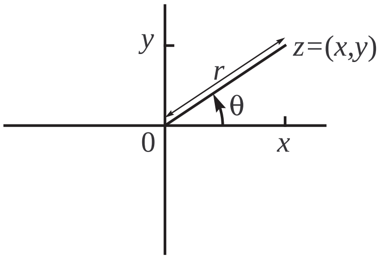


Figure 1.9: Cartesian and polar representation of a complex number z.

The *Cartesian* coordinate pair (x, y) is also equivalent to the *polar* coordinate pair (r, θ), where r is the (nonnegative) length of the vector corresponding to (x, y), and θ is the angle of the vector relative to positive real line. We have

$$\begin{aligned} x &= r\cos\theta \\ y &= r\sin\theta \\ r &= \sqrt{x^2+y^2} \end{aligned}$$

As implied by the arrow in Figure 1.9, the angle θ takes values in $[0, 2\pi)$. It is also acceptable to use angles outside that interval, by adding or subtracting integer multiples of 2π (this does not affect the values of the sine and cosine functions). In particular, angles in the range $(\pi,\ 2\pi)$ (corresponding to points below the x-axis) are often quoted as negative angles in $(-\pi, 0)$. A formula for θ in terms of the Cartesian coordinates x and y is

$$\theta = \arctan\left(\frac{y}{x}\right) + (0 \text{ or } \pi)$$

where π is added if and only if x is negative. Since the conventional range of the function $\arctan(\cdot)$ is the interval $[-\pi/2,\ \pi/2]$, the resulting value of θ is in the range $[-\pi/2,\ 3\pi/2)$.

We also have the following terminology and notation in conjunction with the complex number $z = (x, y) = (r, \theta)$:

- $x = \Re e\{z\}$, the *real part* of z;
- $y = \Im m\{z\}$, the *imaginary part* of z;
- $r = |z|$, the *modulus* or *magnitude* of z;
- $\theta = \angle z$, the *phase* or *angle* of z.

The *complex conjugate* of z is the complex number z^* defined by

$$\Re e\{z^*\} = \Re e\{z\} \qquad \text{and} \qquad \Im m\{z^*\} = -\Im m\{z\}$$

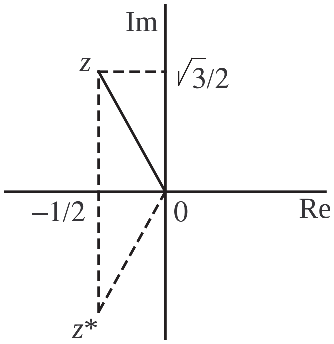
Equivalently,

$$|z^*| = |z| \qquad \text{and} \qquad \angle z^* = -\angle z$$

Example 1.3.1. Consider the complex number

$$z = \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$$

shown in the figure.



Example 1.3.1

We have $\Re e\{z\} = -1/2$, $\Im m\{z\} = \sqrt{3}/2$. The modulus of z equals

$$|z| = \sqrt{\left(\frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2} = 1$$

while the angle of z equals

$$\angle z = \arctan(-\sqrt{3}) + \pi = -\frac{\pi}{3} + \pi = \frac{2\pi}{3}$$

Also,

$$z^* = \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$$

with

$$|z^*| = 1 \qquad \text{and} \qquad \angle z^* = -\frac{2\pi}{3} \qquad \square$$

1.3.2 The Imaginary Unit

A complex number is also represented as

$$z = x + jy$$

where j denotes the *imaginary unit.* The '+' in the expression $z = x + jy$ represents the summation of the two vectors $(x, 0)$ and $(0, y)$, which clearly yields $(x, y) = z$.

It follows easily that if c is a real-valued constant, then

$$cz = cx + j(cy)$$

Also, if $z_1 = x_1 + jy_1$ and $z_2 = x_2 + jy_2$, then

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

Finally, the complex conjugate of $x + jy$ is given by

$$z^* = x - jy$$

Note that

$$\Re e\{z\} = \frac{z + z^*}{2} \qquad \text{and} \qquad \Im m\{z\} = \frac{z - z^*}{2j}$$

1.3.3 Complex Multiplication and its Interpretation

Unlike ordinary two-dimensional vectors, complex numbers can be multiplied together to yield another complex number (vector) on the Cartesian plane. This becomes possible by making the assignment

$$j^2 = -1$$

and applying the usual distributive and commutative laws of algebraic multiplication:

$$\begin{aligned}(a+jb)(c+jd) &= ac+jad+jbc+j^2bd \\ &= (ac-bd)+j(ad+bc)\end{aligned}$$

This result becomes more interesting if we express the two complex numbers in terms of polar coordinates. Let

$$z_1 = r_1(\cos\theta_1 + j\sin\theta_1) \qquad \text{and} \qquad z_2 = r_2(\cos\theta_2 + j\sin\theta_2)$$

Then

$$\begin{aligned}z_1z_2 &= r_1r_2(\cos\theta_1 + j\sin\theta_1)(\cos\theta_2 + j\sin\theta_2) \\ &= r_1r_2[(\cos\theta_1\cos\theta_2 - \sin\theta_1\sin\theta_2) + j(\cos\theta_1\sin\theta_2 + \sin\theta_1\cos\theta_2)] \\ &= r_1r_2[\cos(\theta_1+\theta_2) + j\sin(\theta_1+\theta_2)]\end{aligned}$$

where the last equality follows from standard trigonometric identities. Since a sine-cosine pair uniquely specifies an angle in $[0, 2\pi)$, we conclude that

$$|z_1z_2| = |z_1||z_2| \qquad \text{and} \qquad \angle z_1z_2 = \angle z_1 + \angle z_2$$

Thus, multiplication of two complex numbers viewed as vectors entails:

- *scaling* the length of either vector by the length of the other; and
- *rotation* of either vector through an angle equal to the angle of the other.

If both complex numbers lie on the unit circle (given by the equation $|z| = 1$), then multiplication is a pure rotation, as shown in Figure 1.10.

1.3.4 The Complex Exponential

The Taylor series expansions for $\sin\theta$ and $\cos\theta$ are given by

$$\begin{aligned}\cos\theta &= 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \cdots \\ \sin\theta &= \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \cdots\end{aligned}$$

A power series for $\cos\theta + j\sin\theta$ can be obtained by combining the two Taylor series:

$$\cos\theta + j\sin\theta = 1 + j\theta - \frac{\theta^2}{2!} - j\frac{\theta^3}{3!} + \frac{\theta^4}{4!} + j\frac{\theta^5}{5!} - \cdots$$

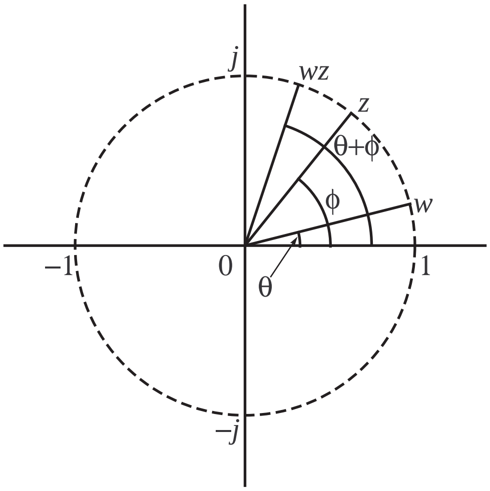


Figure 1.10: Multiplication of two complex numbers w and z on the unit circle.

Starting with j^1, consecutive (increasing) powers of j circle through the four values j, -1, $-j$ and $+1$. Thus the power series above can be expressed as

$$1 + j\theta + \frac{(j\theta)^2}{2!} + \frac{(j\theta)^3}{3!} + \frac{(j\theta)^4}{4!} + \frac{(j\theta)^5}{5!} + \cdots$$

which is recognizable as the Taylor series for e^x with the real-valued argument x replaced by the complex-valued argument $j\theta$. This leads to the expression

$$e^{j\theta} = \cos\theta + j\sin\theta$$

known as *Euler's formula.* Changing the sign of θ yields the complex conjugate:

$$e^{-j\theta} = \cos(-\theta) + j\sin(-\theta) = \cos\theta - j\sin\theta$$

By adding and subtracting the last two equations, we obtain

$$\cos\theta = \frac{e^{j\theta} + e^{-j\theta}}{2} \qquad \text{and} \qquad \sin\theta = \frac{e^{j\theta} - e^{-j\theta}}{2j}$$

respectively.

We have therefore derived an alternative expression for the polar form of a complex number $z = (r, \theta)$:

$$z = re^{j\theta} = |z|e^{j\angle z}$$

Note that if $z_1 = |z_1|e^{j\angle z_1}$ and $z_2 = |z_2|e^{j\angle z_2}$, then direct multiplication gives

$$\begin{aligned} z_1 z_2 &= |z_1||z_2|e^{j\angle z_1}e^{j\angle z_2} \\ &= |z_1||z_2|e^{j(\angle z_1 + \angle z_2)} \end{aligned}$$

as expected.

1.3.5 Inverses and Division

The (multiplicative) *inverse* z^{-1} of z is defined by the relationship

$$zz^{-1} = 1$$

The inverse exists as long as $z \neq 0$, and is easily obtained from the polar form of z:

$$\begin{aligned} z^{-1} &= |z|^{-1}\left(e^{j\angle z}\right)^{-1} \\ &= |z|^{-1}e^{-j\angle z} \end{aligned}$$

Thus the inverse of z is a scaled version of its complex conjugate:

$$z^{-1} = |z|^{-2}z^*$$

and in the special case where z lies on the unit circle, $z^{-1} = z^*$.

For the Cartesian form for the inverse of $z = x + jy$, we have

$$z^{-1} = |z|^{-2}z^* = \frac{x - jy}{x^2 + y^2}$$

As in the case of real numbers, the division of z_1 by z_2 can be defined in terms of a product and an inverse:

$$\frac{z_1}{z_2} = z_1 \cdot \frac{1}{z_2}$$

Example 1.3.2. If $z_1 = \sqrt{3} - j$ and $z_2 = 1 + j$, then

$$\begin{aligned} \frac{z_1}{z_2} &= \frac{\sqrt{3} - j}{1 + j} \\ &= \frac{(\sqrt{3} - j)(1 - j)}{1^2 + 1^2} \\ &= \frac{\sqrt{3} - 1}{2} - j\frac{\sqrt{3} + 1}{2} \end{aligned}$$

For the second expression on the right-hand side, we used the Cartesian form for the inverse of $z_2 = 1 + j$. Alternatively, one can multiply both numerator and denominator by $z_2^* = 1 - j$.

In polar form, $z_1 = 2e^{j(-\pi/6)}$ and $z_2 = \sqrt{2}e^{j(\pi/4)}$, thus

$$\frac{z_1}{z_2} = \frac{2}{\sqrt{2}}e^{j(-\pi/6-\pi/4)} = \sqrt{2}e^{j(-5\pi/12)} \qquad \square$$

1.4 Sinusoids in Continuous Time

1.4.1 The Basic Cosine and Sine Functions

Let θ be a real-valued angle, measured in radians (recall that π rad is the same as 180^0). The basic sinusoidal functions are $\cos\theta$ and $\sin\theta$, shown in Figure 1.11.

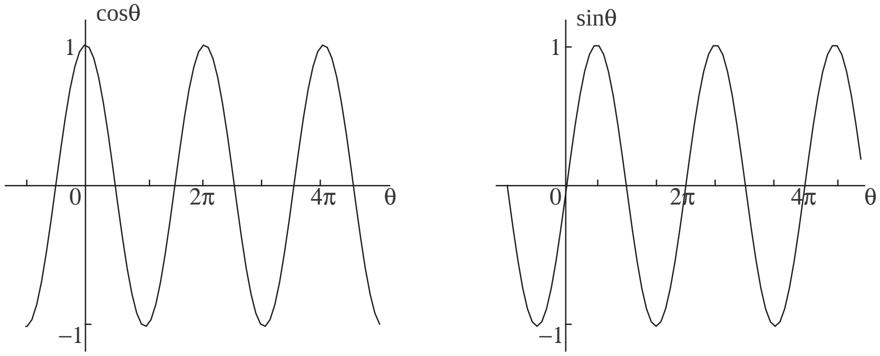


Figure 1.11: The cosine (left) and sine (right) functions.

Our first observation is that both $\cos\theta$ and $\sin\theta$ are *periodic* with period equal to 2π. This means that for any positive or negative integer k,

$$\begin{aligned}\cos(\theta + 2k\pi) &= \cos\theta \\ \sin(\theta + 2k\pi) &= \sin\theta\end{aligned}$$

Our second observation is that the cosine function is *symmetric*, or more precisely, *even-symmetric* about the origin, while the sine function is *anti-symmetric* (or *odd-symmetric*):

$$\cos(-\theta) = \cos\theta$$
$$\sin(-\theta) = -\sin\theta$$

Our third observation is that either function can be obtained from the other by means of a *shift* in the variable θ. The sine function is obtained from the cosine by a *delay* in θ equal to $\pi/2$ radians; equivalently, the cosine is obtained from the sine by an *advance* in θ. This follows also from the trigonometric identity

$$\sin\theta = \cos(\pi/2 - \theta)$$

and the fact that $\cos\theta = \cos(-\theta)$. We thus have

$$\begin{aligned}\sin\theta &= \cos(\theta - \pi/2)\\ \cos\theta &= \sin(\theta + \pi/2)\end{aligned}$$

1.4.2 The General Real-Valued Sinusoid

The shift properties discussed above allow us to express sinusoidal signals in terms of either the sine or the cosine function. We obtain the general form for the real-valued sinusoid by modifying the function $\cos\theta$ in three ways:

1. Vertical Scaling. Multiplication of $\cos\theta$ by a constant A yields an *amplitude* (peak value) equal to $|A|$. It is customary to use positive values for A. A negative A would result in a sign inversion, which is better described using a shift in θ:

$$-\cos\theta = \cos(\theta + \pi)$$

2. Horizontal Scaling. The horizontal scale can be changed by making θ a linear function of another variable; time t is often chosen for that purpose. We thus have, for $\Omega \geq 0$,

$$\theta = \theta(t) = \Omega t$$

and the resulting sinusoid is

$$x(t) = A\cos\Omega t$$

The scale parameter Ω is known as the *angular frequency*, or *radian frequency* of the signal $x(t)$. If t is measured in seconds, then Ω is measured in radians per second (rad/sec).

A full cycle (period) of $\cos\theta$ lasts 2π radians, or equivalently, $2\pi/\Omega$ seconds. It follows that the period T of $x(t)$ is

$$T = \frac{2\pi}{\Omega} \quad \text{(seconds)}$$

The *(cyclic) frequency* of $x(t)$ is the number of cycles per unit time, i.e.,

$$f = \frac{1}{T} = \frac{\Omega}{2\pi}$$

The unit of f is cycles/second, or Hertz (Hz). We thus have

$$1 \text{ Hz} = 2\pi \text{ rad/sec}$$

In Figure 1.12, $\Omega_2 > \Omega_1$, and thus $T_2 < T_1$.

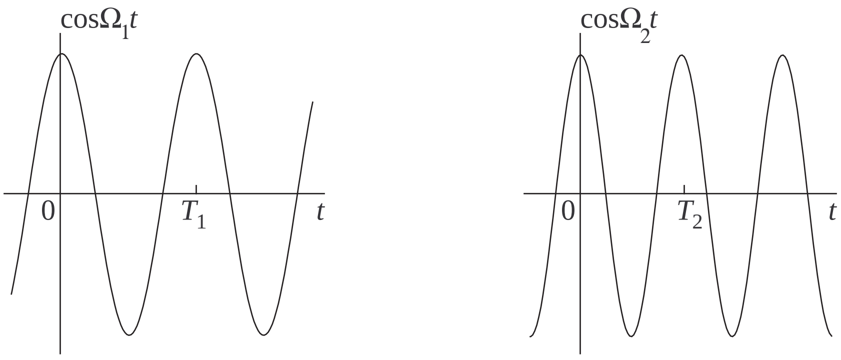


Figure 1.12: Sinusoids of two different frequencies.

3. Horizontal Shifting. The introduction of a *phase shift* angle ϕ in the argument of the cosine results in

$$x(t) = A\cos(\Omega t + \phi)$$

which is the most general form for the real-valued sinusoid. A positive value of ϕ represents a phase (angle) advance of $\phi/2\pi$ cycles, or a time advance of ϕ/Ω seconds; a negative value of ϕ represents a delay. Clearly, $x(t)$ can be also expressed in terms of the sine function, using an additional phase shift (see the discussion earlier):

$$\begin{aligned}\sin(\Omega t + \phi) &= \cos\left(\Omega t + \phi - \frac{\pi}{2}\right)\\ \cos(\Omega t + \phi) &= \sin\left(\Omega t + \phi + \frac{\pi}{2}\right)\end{aligned}$$

Example 1.4.1. Let us sketch the signal

$$x(t) = 3\cos\left(10\pi t + \frac{3\pi}{4}\right)$$

showing all features of interest such as the signal amplitude; the initial value of the signal ($t = 0$); and the positions of zeros, maxima and minima.

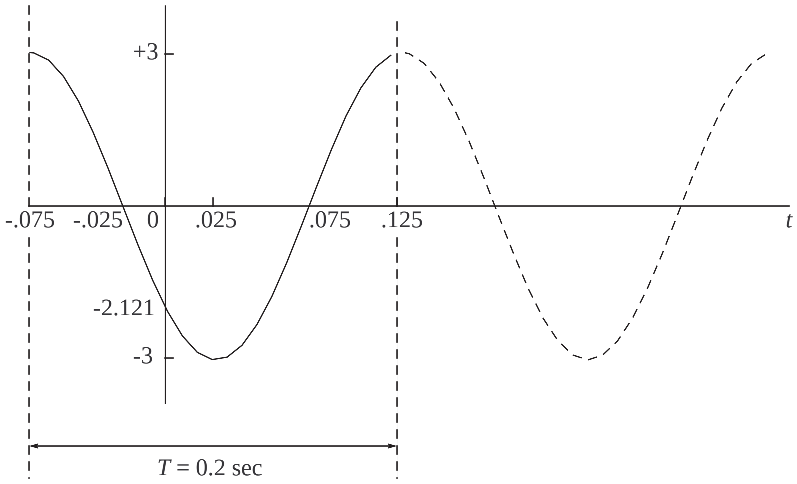
First, we draw one cycle of the cosine function (peak to valley to peak) without marking the time origin. The duration of that cycle is the period T of $x(t)$, which is obtained from Ω (the coefficient of t):

$$\Omega = 10\pi \text{ rad/sec} \Rightarrow f = 5 \text{ Hz} \Rightarrow T = 0.2 \text{ sec}$$

The cycle drawn starts on a peak, which would be the value of the signal at $t = 0$ if the phase shift were equal to 0. In this case, the phase shift equals $3\pi/4$, which is equivalent to 3/8 cycles. Thus the correct placement of the time origin ($t = 0$) is *to the right* of the leftmost peak, at a distance corresponding to 3/8 cycles, or 0.075 seconds. The value of the signal at $t = 0$ is

$$3\cos(3\pi/4) = -3\sqrt{2}/2 = -2.121$$

The positions of the zeros, maxima and minima of $x(t)$ are now marked relative to the origin. Note that these occur at quarter-cycle intervals, i.e., every 0.05 seconds. Since the phase shift at $t = 0$ is 3/8 cycles, the first such point on the positive t-axis will occur 1/8 cycles later (i.e., at $t = 0.025$ seconds), and will be a minimum.



Example 1.4.1

More cycles of the signal can then be added (e.g., the dotted line in the figure). □

1.4.3 Complex-Valued Sinusoids and Phasors

In our discussion of complex numbers, we established Euler's formula

$$e^{j\theta} = \cos\theta + j\sin\theta$$

and obtained the identities

$$\cos\theta = \frac{e^{j\theta} + e^{-j\theta}}{2} \qquad \text{and} \qquad \sin\theta = \frac{e^{j\theta} - e^{-j\theta}}{2j}$$

It is clear from the first of the two identities that $\cos\theta$ can be obtained from the two complex numbers $e^{j\theta}$ and $e^{-j\theta}$ (both on the unit circle) by projecting either number onto the real axis, or, equivalently, by averaging the two numbers. Analogous conclusions can be drawn about the sine function, using the imaginary axis and a slightly different linear combination.

The real-valued sinusoid

$$x(t) = A\cos(\Omega t + \phi)$$

can also be obtained from the *complex-valued sinusoid*

$$z(t) = Ae^{j(\Omega t+\phi)}$$

using

$$x(t) = \Re e\left\{Ae^{j(\Omega t+\phi)}\right\}$$

or, equivalently,

$$x(t) = \frac{Ae^{j(\Omega t+\phi)} + Ae^{-j(\Omega t+\phi)}}{2} = \frac{z(t) + z^*(t)}{2}$$

To interpret the above relationships, think of $z(t)$ and $z^*(t)$ as two points on the circle $|z| = A$, with initial angular positions (at $t = 0$) given by ϕ and $-\phi$. The two points rotate in opposite directions on the circle; their angular velocities are equal in magnitude. The real-valued sinusoid $x(t)$ is obtained by either projecting $z(t)$ onto the real axis, or taking the midpoint of the segment joining $z(t)$ and $z^*(t)$, as shown in Figure 1.13.

The foregoing discussion used the concept of *negative* angular frequency to describe an angle which is *decreasing* linearly in time. Negative frequencies are of no particular value in expressing *real* sinusoids, since

$$A\cos(-\Omega t + \phi) = A\cos(\Omega t - \phi) \ ,$$

i.e., changing the sign of Ω is no different from changing the sign of the phase angle ϕ. Using negative frequencies for *complex* sinusoids, on the other hand, allows us to linearly combine such complex signals to obtain a real-valued sinusoid. This concept will be further discussed in signal analysis.

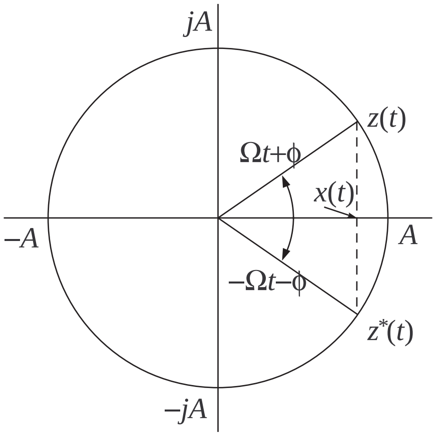


Figure 1.13: Rotating phasor.

A complex sinusoid such as $z(t)$ is also known as a *rotating phasor.* Its initial position at $t = 0$ is referred to as a *stationary phasor.* As expected,

$$Ae^{j(\Omega t+\phi)} = Ae^{j\phi} \cdot e^{j\Omega t}$$

i.e., the rotating phasor is obtained from the stationary phasor by rotation through an angle Ωt.

Stationary phasors are particularly useful in depicting the relative phases of sinusoidal signals of the *same* frequency, such as voltages and currents in an AC (alternating current) circuit. One particular application involves sums of such signals:

Fact. *If* $x_1(t), \ldots, x_M(t)$ *are real sinusoids given by*

$$x_m(t) = A_m \cos(\Omega t + \phi_m)$$

then

$$x_1(t) + \cdots + x_M(t) = A\cos(\Omega t + \phi)$$

where

$$Ae^{j\phi} = \sum_{m=1}^{M} A_m e^{j\phi_m}$$

The fact that a sum of sinusoids of the same frequency is also sinusoidal is not at all obvious. To prove it, note first that

$$x_m(t) = \Re e\left\{A_m e^{(j\Omega t+\phi_m)}\right\}$$

so that

$$\sum_{m=1}^{M} x_m(t) = \sum_{m=1}^{M} \Re e \left\{ A_m e^{(j\Omega t + \phi_m)} \right\}$$

Since $\Re e\{z_1 + z_2\} = \Re e\{z_1\} + \Re e\{z_2\}$, it follows that

$$\begin{aligned}
\sum_{m=1}^{M} x_m(t) &= \Re e \left\{ \sum_{m=1}^{M} A_m e^{(j\Omega t + \phi_m)} \right\} \\
&= \Re e \left\{ \sum_{m=1}^{M} A_m e^{j\phi_m} e^{j\Omega t} \right\} \\
&= \Re e \left\{ \left(\sum_{m=1}^{M} A_m e^{j\phi_m} \right) \cdot e^{j\Omega t} \right\} \\
&= \Re e \left\{ A e^{j(\Omega t + \phi)} \right\} = A \cos(\Omega t + \phi)
\end{aligned}$$

where

$$A = \left| \sum_{m=1}^{M} A_m e^{j\phi_m} \right| \qquad \text{and} \qquad \phi = \angle \left(\sum_{m=1}^{M} A_m e^{j\phi_m} \right)$$

i.e.,

$$A e^{j\phi} = \sum_{m=1}^{M} A_m e^{j\phi_m} \qquad \Box$$

One way of interpreting this result is that the stationary phasor $Ae^{j\phi}$ corresponding to the sum signal $x(t)$ is just the (vector) sum of the stationary phasors of each of the components of $x_m(t)$. The same, of course, holds for the rotating phasors.

Example 1.4.2. To express

$$x(t) = 3\cos\left(10\pi t + \frac{3\pi}{4}\right) + 5\sin\left(10\pi t + \frac{\pi}{6}\right)$$

as a single sinusoid, we first write the second term as

$$5\cos\left(10\pi t + \frac{\pi}{6} - \frac{\pi}{2}\right) = 5\cos\left(10\pi t - \frac{\pi}{3}\right)$$

The stationary phasor for the sum signal $x(t)$ is given by

$$\begin{aligned}
3 \cdot e^{j(3\pi/4)} + 5 \cdot e^{-j(\pi/3)} &= 0.3787 - j2.209 \\
&= 2.241 \cdot e^{-j1.401}
\end{aligned}$$

Thus

$$x(t) = 2.241 \cdot \cos(10\pi t - 1.401) \qquad \Box$$

1.5 Sinusoids in Discrete Time

1.5.1 Discrete-Time Signals

A *discrete-time*, or *discrete-parameter*, signal is a sequence of real or complex numbers indexed by a discrete variable (or parameter) n, which takes values in a set of integers I. Examples of I include infinite sets such as $\mathbf{Z}$ (all integers), $\mathbf{N}$ (positive integers), etc.; as well as finite sets such as $\{1, \ldots, M\}$. In the latter case, the signal is also called a *vector*.

In what follows, we will assume that the discrete (time) parameter n takes values over all integers, i.e., $n \in \mathbf{Z}$. The notation for a discrete-time signal involves square brackets around n, to differentiate it from a continuous-time signal:

$$s[n],\ n \in \mathbf{Z} \qquad \text{distinct from} \qquad s(t),\ t \in \mathbf{R}$$

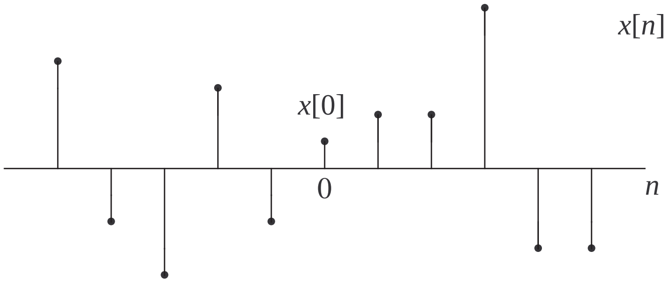


Figure 1.14: A discrete-time signal.

One important note here is that n has no physical dimensions, i.e, it is a pure integer number; this is true even when $s[n]$ represents samples of a continuous-parameter signal whose value varies in time, space, etc.

1.5.2 Definition of the Discrete-Time Sinusoid

A sinusoid in discrete time is defined as in continuous time, by replacing the continuous variable t by the discrete variable n:

$$\begin{aligned} x[n] &= A\cos(\omega n + \phi) \\ y[n] &= A\sin(\omega n + \phi) \\ z[n] &= Ae^{j(\omega n + \phi)} = x[n] + jy[n] \end{aligned}$$

Again, we assume $A > 0$. We will use the lower-case letter ω for angular frequency in discrete time because it is different from the parameter Ω used for continuous-time sinusoids. The difference is in the units: t is measured in seconds, hence Ω is measured in radians *per second*; while n has no units, hence ω is just an angle, measured in radians or radians *per sample*.

Example 1.5.1. If $\omega = 0$, then

$$x[n] = A\cos\phi, \qquad y[n] = A\sin\phi, \qquad \text{and} \qquad z[n] = Ae^{j\phi}$$

i.e., all three signals are constant in value. □

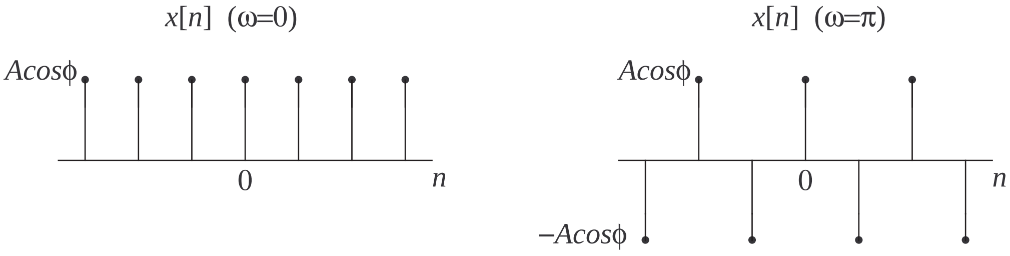
Example 1.5.2. If $\omega = \pi$, then

$$z[n] = Ae^{j(\pi n+\phi)} = A(e^{j\pi})^n \cdot e^{j\phi} = A(-1)^n \cdot e^{j\phi}$$

since $e^{j\pi} = -1$. Thus also

$$x[n] = A(-1)^n \cos\phi \qquad \text{and} \qquad y[n] = A(-1)^n \sin\phi$$

The signal $x[n]$ is shown in the figure. Note the oscillation between the two extreme values $+A\cos\phi$ and $-A\cos\phi$, which is the fastest possible for a discrete-time sinusoid. As we shall see soon, $\omega = \pi$ rad/sample is, effectively, the highest possible frequency for a discrete-time sinusoid. □



Example 1.5.2

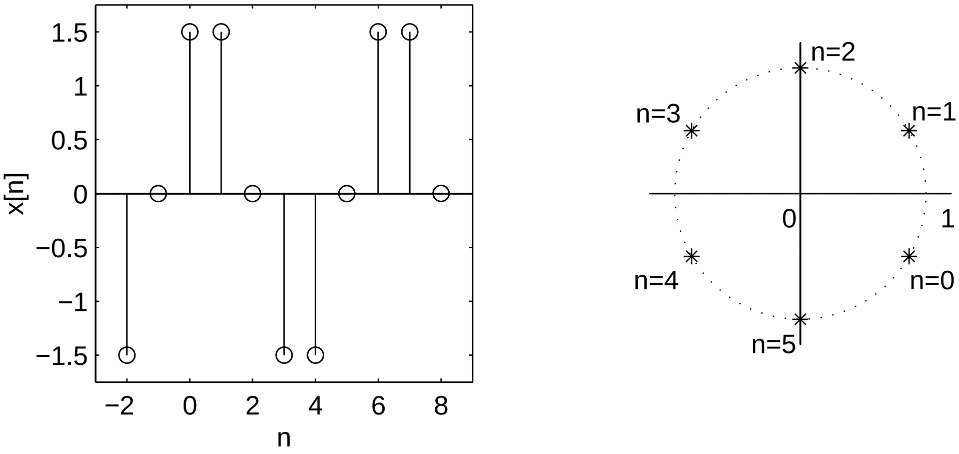
Example 1.5.3. The discrete-time sinusoid

$$x[n] = \sqrt{3}\cos\left(\frac{\pi n}{3} - \frac{\pi}{6}\right)$$

is shown in the figure (on the left). The graph was generated using the `stem` command in MATLAB:

```
n = -2:8;
x = sqrt(3) * cos((pi/3)*n -(pi/6));
stem(n,x)
```

Note that the signal is periodic—it repeats itself every 6 samples. The angle in the argument of the cosine is marked on the unit circle (shown on the right), for $n = 0, ..., 5$. □



Example 1.5.3

1.5.3 Equivalent Frequencies for Discrete-Time Sinusoids

As we mentioned earlier, the angular frequency parameter ω, which appears in the argument

$$\omega n + \phi$$

of a discrete-time sinusoid, is an angle. It actually acts as an angle *increment*: With each sample, the angle in the argument of the sinusoid is incremented by ω.

Since the sinusoids $\sin\theta$, $\cos\theta$ and $e^{j\theta}$ are all periodic in θ (with period 2π), it is conceivable that two different values of ω may result in *identical* sinusoidal waveforms. To see that this is indeed the case, consider

$$z[n] = Ae^{j(\omega n+\phi)}$$

We know that $e^{j\psi} = e^{j\theta}$ if and only if the angles θ and ψ correspond to the same point on the unit circle, i.e.,

$$\psi = \theta + 2k\pi$$

where k is an integer. If we replace ω by $\omega + 2k\pi$ in the equation for $z[n]$, we obtain a new signal $v[n]$ given by

$$\begin{aligned} v[n] &= Ae^{j((\omega+2k\pi)n+\phi)} \\ &= Ae^{j(\omega n+\phi+2kn\pi)} \\ &= Ae^{j(\omega n+\phi)} \\ &= z[n] \end{aligned}$$

where the angle argument was reduced by an integer ($= kn$) multiple of 2π without affecting the value of the sinusoid. Thus the signals $z[n]$ and $v[n]$ are identical.

Definition 1.5.1. The angular frequencies ω and ω' are *equivalent* for *complex-valued* sinusoids in discrete time if

$$\omega' = \omega + 2k\pi, \quad (k \in \mathbf{Z})$$ □

Equivalent frequencies result in identical complex sinusoids, provided the initial phase shifts ϕ are equal. Thus the effective range of frequencies for complex sinusoids can be taken as $(0, 2\pi]$ or $(-\pi, \pi]$ rad/sample, corresponding to one full rotation on the unit circle.

Example 1.5.4. The three complex sinusoids

$$\begin{aligned} z^{(1)}[n] &= \exp\left\{j\left(-\frac{\pi n}{6} + \phi\right)\right\} \\ z^{(2)}[n] &= \exp\left\{j\left(\frac{11\pi n}{6} + \phi\right)\right\} \\ z^{(3)}[n] &= \exp\left\{j\left(\frac{23\pi n}{6} + \phi\right)\right\} \end{aligned}$$

all represent the same signal (in discrete time). □

The notion of equivalent frequency also applies to the real-valued sinusoid

$$x[n] = A\cos(\omega n + \phi)$$

since $x[n] = \Re e\{z[n]\}$. Thus two frequencies which differ by an integral multiple of 2π will result in identical real-valued sinusoids provided the initial phase shifts ϕ are the same.

Moreover, for the real valued sinusoid above, it is fairly easy to show that using $\omega' = -\omega + 2k\pi$ instead of ω will result in an identical signal provided the sign of ϕ is also reversed. Let

$$u[n] = A\cos((-\omega + 2k\pi)n - \phi)$$

Then, recalling the identity

$$\cos(-\theta) = \cos(\theta)$$

we have

$$\begin{aligned} u[n] &= A\cos((-\omega + 2k\pi)n - \phi) \\ &= A\cos(\omega n + \phi - 2k\pi n) \\ &= A\cos(\omega n + \phi) \\ &= x[n] \end{aligned}$$

We also note that no sign reversal is needed for ϕ if $\omega = 0$ or $\omega = \pi$.

Definition 1.5.2. The angular frequencies ω and ω' are *equivalent* for *real-valued* sinusoids in discrete time if

$$\omega' = \pm\omega + 2k\pi, \quad (k \in \mathbf{Z})$$ □

This further reduces the effective range of frequency for real sinusoids to the interval $[0,\ \pi]$. This is illustrated in Figure 1.15.

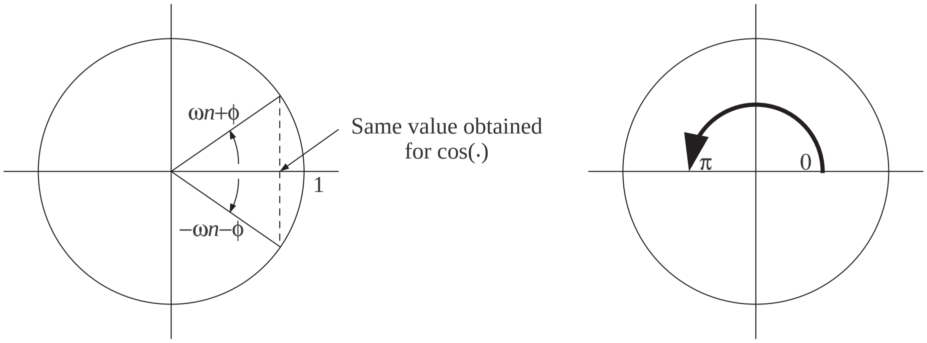


Figure 1.15: Two equivalent frequencies for real sinusoids (left); effective range of frequencies for real sinusoids (right).

Example 1.5.5. The three real sinusoids

$$\begin{aligned} x^{(1)}[n] &= \cos\left(-\frac{\pi n}{6}+\phi\right) \\ x^{(2)}[n] &= \cos\left(\frac{11\pi n}{6}+\phi\right) \\ x^{(3)}[n] &= \cos\left(\frac{23\pi n}{6}+\phi\right) \end{aligned}$$

all represent the same signal, which is also given by (note the sign reversal in the phase):

$$\cos\left(\frac{\pi n}{6}-\phi\right) = \cos\left(\frac{13\pi n}{6}-\phi\right) = \cos\left(\frac{25\pi n}{6}-\phi\right) \qquad \square$$

1.5.4 Periodicity of Discrete-Time Sinusoids

While continuous-time sinusoids are always periodic with period $T = 2\pi/\Omega$, periodicity is the exception (rather than the rule) for discrete-time sinusoids. First, a refresher on periodicity:

Definition 1.5.3. A discrete-time signal $s[n]$ is periodic if there exists an integer $N > 0$ such that for all $n \in \mathbf{Z}$,

$$s[n+N] = s[n]$$

The smallest such N is called the (fundamental) *period* of $s[n]$. □

For $z[n] = Ae^{j(\omega n+\phi)}$, the condition

$$(\forall n) \qquad z[n+N] = z[n]$$

holds for a given $N > 0$ if and only if

$$\omega(n+N)+\phi = \omega n + \phi + 2k\pi$$

which reduces to

$$\omega = \frac{k}{N}\cdot 2\pi$$

Thus $z[n]$ is periodic if and only if its frequency is a *rational* (i.e., ratio of two integers) *multiple* of 2π. The period (smallest such N) is found by cancelling out common factors between the numerator k and denominator N.

The same condition for periodicity can be obtained for the real-valued sinusoid $x[n] = A\cos(\omega n + \phi)$.

Example 1.5.6. The signal

$$x[n] = \cos(10n + 0.2)$$

is not periodic, while

$$s[n] = \cos\left(\frac{13\pi n}{15} + 0.4\right)$$

is. Since

$$\frac{13\pi}{15} = \frac{13}{30} \cdot 2\pi$$

(note that there are no common factors in the numerator and the denominator of the last fraction), the period of $s[n]$ is $N = 30$. □

1.6 Sampling of Continuous-Time Sinusoids

1.6.1 Sampling

Sampling is the process of recording the values taken by a continuous-time signal at discrete *sampling instants*. *Uniform* sampling is most commonly used: the sampling instants T_s are seconds apart in time, where T_s is the *sampling period*. The *sampling rate*, or *sampling frequency*, f_s is the reciprocal of the sampling period:

$$f_s = \frac{1}{T_s} \quad \text{(samples/sec)}$$

If the continuous time signal is $x(t)$, the discrete-signal resulting from sampling is

$$x[n] = x(nT_s)$$

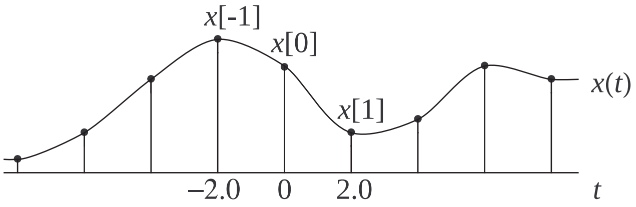


Figure 1.16: Sampling every two seconds (i.e., $T_s = 2.0$).

Sampling is one of two main components of analog-to-digital (A-to-D), the other component being quantization. The reverse process of digital-to-analog (D-to-A) conversion involves *interpolation*, namely reconstructing the continuous-time signal from the discrete (and quantized) samples. The sampling rate f_s determines, to a large extent, the quality of the interpolated analog signal. If the sampling rate is high enough, the samples will contain sufficient information about the sampled analog signal to allow accurate interpolation; otherwise, fine detail will be lost and the reconstructed signal will be a poor replica of the original. The minimum sampling rate f_s required for accurate interpolation is known as the *Nyquist rate*, which depends on the range of frequencies spanned by the original signal (in terms of its sinusoidal components). Although the theory behind the Nyquist rate is beyond the scope of this chapter, the necessity of sampling at the Nyquist rate (or higher) can be seen by considering simple properties of sampled sinusoidal signals.

1.6.2 Discrete-Time Sinusoids Obtained by Sampling

Sampling a continuous-time sinusoid produces a discrete-time sinusoid. Consider, for example, the real-valued sinusoid

$$x(t) = A\cos(\Omega t + \phi)$$

which has period $T = 2\pi/\Omega$ and cyclical frequency $f = \Omega/2\pi$. We have

$$\begin{aligned} x[n] &= A\cos(\Omega n T_s + \phi) \\ &= A\cos(\omega n + \phi) \end{aligned}$$

where

$$\omega = \Omega T_s = \frac{\Omega}{f_s}$$

Note that, as expected, ω has no physical dimensions, i.e., it is just an angle. It can also be expressed as

$$\omega = 2\pi \cdot \frac{T_s}{T} = 2\pi \cdot \frac{f}{f_s}$$

Thus the frequency of the discrete-time sinusoid obtained by sampling a continuous-time sinusoid is directly proportional to the sampling period T_s, or equivalently, inversely proportional to the sampling frequency f_s.

Suppose T_s is small in relation to T, or equivalently, f_s is high compared to f. Since many samples are taken in each period, the sample values will vary slowly, and thus frequency of the discrete-time signal will be low. As T_s increases (i.e., f_s decreases), the frequency ω of the discrete-time sinusoid also increases. Recall, however, that there is an upper limit, equal to π rad/sample, on the effective frequency of a real-valued discrete-time sinusoid. That value will be attained for

$$T_s = \frac{\pi}{\Omega} = \frac{T}{2},$$

(equivalently, for $f_s = 2f$). When T_s exceeds $T/2$, the effective frequency of the resulting samples will decrease with T_s. This is consistent with the fact that

$$\pi + \delta \qquad \text{and} \qquad \pi - \delta = -(\pi + \delta) + 2\pi$$

are equivalent frequencies. The effective frequency will become zero when $T_s = T$ (or $f_s = f$), and then it will start increasing again, etc.

Example 1.6.1. Consider the continuous-time sinusoid

$$x(t) = \cos(400\pi t + 0.8)$$

which has period $T = 5$ ms and frequency $f = 200$ Hz. Suppose the following three sampling rates are used: 1,000, 500, and 250 samples/sec.

In the first case, $T_s = 1$ ms and $\omega = (400\pi)/1000 = 2\pi/5$. The resulting signal is

$$x_1[n] = \cos\left(\frac{2\pi n}{5} + 0.8\right)$$

In the second case, $T_s = 2$ ms and $\omega = 4\pi/5$. The resulting signal is

$$x_2[n] = \cos\left(\frac{4\pi n}{5} + 0.8\right)$$

In the third case, $T_s = 4$ ms and $\omega = 8\pi/5$. The resulting signal is

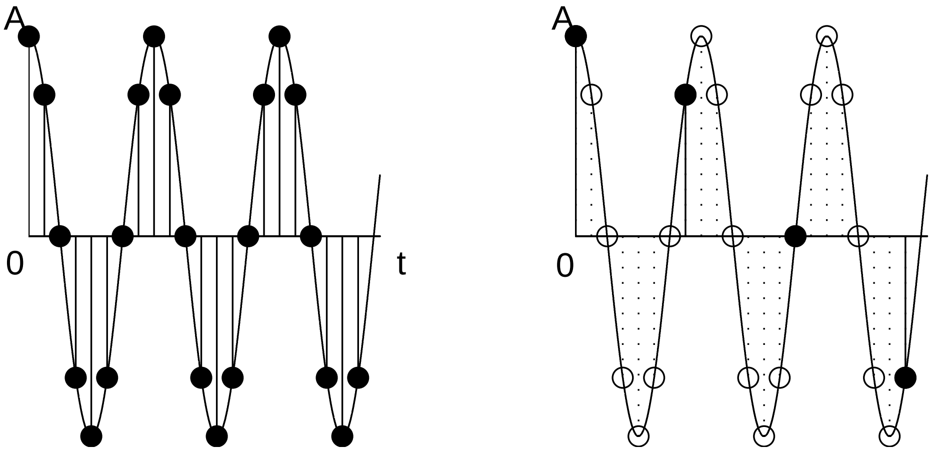
$$x_3[n] = \cos\left(\frac{8\pi n}{5} + 0.8\right) = \cos\left(\frac{2\pi n}{5} - 0.8\right)$$

i.e., it is a sinusoid of the same frequency as $x_1[n]$ (but with different phase shift). □

Example 1.6.2. The graphs show the continuous time sinusoid $A\cos(2\pi f t)$ sampled at $f_s = 8f$ (left) and $f_s = (8f)/7$ (right). The sample sequence on the right is formed by taking every seventh sample from the sequence on the left. Clearly, the two sequences are identical, and are given by the discrete-time sinusoid $A\cos(\pi n/4)$. □

1.6.3 The Concept of Alias

The notion of equivalent frequency was implicitly used in Example 1.6.1 to show that a given continuous-time sinusoid can be sampled at two *different* sampling rates to yield discrete sinusoids of the *same* frequency (differing possibly in the phase). If we were to plot the two sample sequences using an actual time axis, i.e., t instead of n (as in the graphs of Example 1.6.2), we would observe that one sequence of samples contains a lot more information about the analog signal. In the case of a sinusoidal signal, most of this information is redundant: as few as three samples may suffice to determine the unknown parameters Ω, ϕ and A (only two samples if the frequency Ω is known), and hence to interpolate the signal perfectly. On the other hand,



Example 1.6.2

for an arbitrary signal $x(t)$ that is neither sinusoidal nor (more generally) periodic, interpolation from a finite number of samples is impossible; for such a signal, a higher sampling rate is likely to result in better (more faithful) reconstruction.

Most analog (i.e., continuous-time) signals encountered in practice can be formed by *summing* together sinusoids of different frequencies and phase shifts. In most cases, the frequencies involved span an entire (continuous) interval or band, in which case *integration* is more appropriate than summation. The details of the sinusoidal representation of continuous-time signals (known as the Fourier transform) are not essential at this point. To appreciate a key issue which arises in sampling analog signals, it suffices to consider the simple class of signals which are sums of finitely many sinusoids.

Consider again the sinusoid

$$x(t) = A\cos(\Omega t + \phi)$$

sampled at a rate f_s which is *fixed* (unlike the situation discussed in the previous subsection). The resulting discrete-time sinusoid $x[n]$ has frequency $\omega = \Omega T_s = 2\pi(f/f_s)$.

Consider also a second sinusoid

$$x'(t) = A'\cos(\Omega' t + \phi')$$

sampled at the same rate, resulting in $x'[n]$ having frequency $\omega' = \Omega' T_s = 2\pi(f'/f_s)$.

The two frequencies ω and ω' are equivalent (allowing both $x[n]$ and $x'[n]$ to be expressed in terms of the same frequency) provided that

$$\omega' = \pm\omega + 2k\pi \ , \qquad k \text{ integer}$$

This is equivalent to

$$\frac{f'}{f_s} = \pm\frac{f}{f_s} + k$$

or, simply,

$$f' = \pm f + kf_s$$

for some integer k.

Frequencies f and f' (in Hz) satisfying the above relationship are known as *aliases* of each other with respect to the sampling rate f_s (in samples/second). Continuous-time sinusoids at frequencies related in that manner, will, when sampled at the appropriate rate, yield discrete-time samples having the same (effective) frequency.

Example 1.6.3. Consider again Example 1.6.1, where $f = 200$ Hz.

If $f_s = 1,000$ Hz, the aliases of f are at

$$f' = 200 + k(1000) \qquad \text{and} \qquad f' = -200 + k(1000) \text{ Hz}$$

If $f_s = 500$ Hz, the aliases of f are at

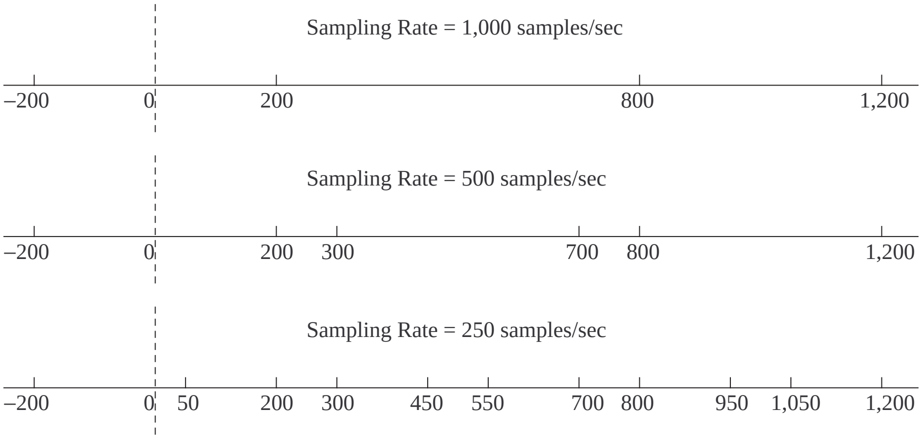
$$f' = 200 + k(500) \qquad \text{and} \qquad f' = -200 + k(500) \text{ Hz}$$

If $f_s = 250$ Hz, the aliases of f are at

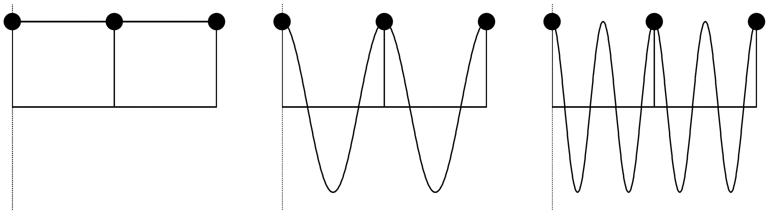
$$f' = 200 + k(250) \qquad \text{and} \qquad f' = -200 + k(250) \text{ Hz}$$

The aliases (in Hz) are plotted for each sampling rate. Clearly, only positive frequencies are of interest, and thus it suffices to consider $k \geq 0$ in both series. In the first two cases, the lowest (positive) alias is f itself. In the third case, the lowest alias is at $-200 + 250 = 50$ Hz. □

Example 1.6.4. The graphs illustrate how three continuous-time sinusoids with frequencies $f = 0$, $f = f_s$ and $f = 2f_s$ can yield the same discrete-time sequence with $\omega = 0$. □


Sampling Rate = 1,000 samples/sec
–200
0
200
800
1,200
Sampling Rate = 500 samples/sec
–200
0
200
300
700
800
1,200
Sampling Rate = 250 samples/sec
–200
0
50
200
300
450
550
700
800
950
1,050
1,200

Example 1.6.3



Example 1.6.4

1.6.4 The Nyquist Rate

The term *aliasing* is used in reference to sampling continuous-time signals which are expressible as sums of sinusoids of different frequencies (most signals fall in that category). Aliasing occurs in sampling a continuous-time signal $x(t)$ if two (or more) sinusoidal components of $x(t)$ are aliases of each other with respect to the sampling rate.

For example, consider

$$x(t) = \cos \Omega t + 4 \cos \Omega' t$$

where Ω and Ω' are aliases with respect to the sampling rate f_s. The resulting sequence of samples can be expressed as

$$x[n] = \cos \omega n + 4 \cos \omega' n$$

Since ω and ω' are equivalent frequencies (and no phase shifts are involved here), we can also write

$$x[n] = 5 \cos \omega n$$

Reconstructing $x(t)$ from the samples $x[n]$ without knowing the amplitudes of the two components $\cos\Omega t$ and $\cos\Omega' t$ is a hopeless task: the same sequence of samples could have been obtained from, e.g.,

$$u(t) = 2\cos\Omega t + 3\cos\Omega' t \qquad \text{or} \qquad v(t) = -\cos\Omega t + 6\cos\Omega' t$$

which are very different signals from $x(t)$. This shows convincingly that aliasing is undesirable in sampling signals composed of sinusoids. Thus absence of aliasing is a *necessary* condition for the faithful reconstruction of the sampled signal.

As we mentioned earlier, typical analog signals consist of a continuum of sinusoids spanning an entire interval, or band, of frequencies; these sinusoidal components are "summed" by an integral over frequency (rather than by a conventional sum). The effect of aliasing is the same whether we sum or integrate over frequency: components at frequencies that are aliases of each other cannot be resolved on the basis of the samples obtained, hence the amplitude and phase information needed to incorporate those components into the reconstructed signal will be unavailable.

If the sinusoidal components of $x(t)$ have frequencies which are limited to the frequency band $[0,\ f_B]$ Hz (where f_B is the signal *bandwidth*), aliasing is avoided if no frequency f in that band has an alias $f' = \pm f + kf_s$ (excluding itself) in the same band. With the aid of Figure 1.17, we will determine the *minimum* sampling rate f_s such that no aliasing occurs.

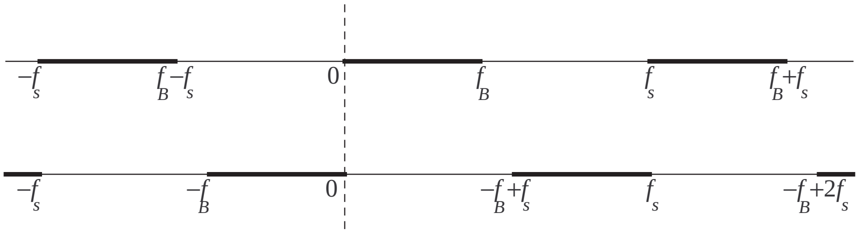


Figure 1.17: Illustration of the Nyquist condition: $-f_B + f_s > f_B$, i.e., $f_s > 2f_B$.

Consider the alias relationship $f' = f + kf_s$ first. Each k corresponds to an interval $[kf_s,\ f_B + kf_s]$ of aliases of frequencies in $[0,\ f_B]$ (top graph). No aliasing means that neither of the intervals $[-f_s,\ f_B - f_s]$ and $[f_s,\ f_B + f_s]$ (corresponding to $k = -1$ and $k = 1$, respectively) overlaps with $[0,\ f_B]$. This is ensured if and only if $f_s > f_B$.

Next, consider the alias relationship $f' = -f + kf_s$, which gives aliases of $[0,\ f_B]$ in the interval $[-f_B + kf_s,\ kf_s]$ (bottom graph). No aliasing means that the interval $[-f_B + f_s,\ f_s]$ does not overlap with $[0,\ f_B]$, which is ensured if and only if $f_s > 2f_B$.

Thus aliasing is avoided if $f_s > 2f_B$. In other words, the sampling rate must be greater than *twice the bandwidth* of the analog signal. This minimum rate is known as the *Nyquist rate.* Sampling at a rate higher than the Nyquist rate is a *necessary* condition for the faithful reconstruction of the signal $x(t)$ from its samples $x[n]$.

It turns out that the condition $f_s > 2f_B$ is also *sufficient* for the complete recovery of the analog signal from its samples. The proof of this fact requires more advanced tools in signal analysis, and will be outlined in the epilogue following Chapter 4.

Problems

Section 1.1

***P* 1.1.** What is the IEEE 754 double-precision encoding (in hexadecimal form) of the decimal numbers 0.125 and -15.75? You can use MATLAB (`format hex`) to verify your answers.

***P* 1.2.** Type `help round` and `help fix` in MATLAB and read the description of these two functions. If `x` is an arbitrary number and `m` is a positive integer, explain in your own words the effect of using the MATLAB commands

```
10^(-m)*round(x*10^m)
```

and

```
10^(-m)*fix(x*10^m)
```

***P* 1.3.** A terminating decimal expansion may correspond to an infinite (non-terminating) binary expansion. Such is the case with the decimal number 0.1, which is rounded in MATLAB—i.e., it is represented by a *different* precise binary number a. What is the value of the error $a - 0.1$? (Hint: using `format hex`, you can obtain the hexadecimal representation `3FB999999999999A` for a. You may find the geometric sum formula useful here.)

Section 1.2

***P* 1.4.** **(i)** Using your calculator, compute the value of

$$y = 1 - \frac{\sin x}{x}$$

for $x = 0.05$ radians, rounding to four significant digits *at the end*. Denote your answer by y_1.

(ii) Repeat the calculation of y, rounding to four significant digits *at each stage*. Denote your answer by y_2.

(iii) Compute the value of y using the expansion

$$\sin x = x - \frac{x^3}{6} + \frac{x^5}{120} - \cdots$$

rounding to four significant digits at each stage. Denote your answer by y_3.

***P* 1.5.** **(i)** Let $x = 0.02$. Using the full precision available on your calculator, compute

$$y = \frac{1+x}{\sqrt{1+2x}} - 1$$

Denote your answer by y_1.

(ii) Repeat the calculation of y, rounding to four significant digits at each stage. Denote your answer by y_2. Also compute the error relative to y_1.

(iii) Compute the value of y using the binomial expansion

$$(1+a)^r = 1 + ra + \frac{r(r-1)}{2!}a^2 + \frac{r(r-1)(r-2)}{3!}a^3 + \cdots$$

(valid for any r and $|a| < 1$). Include powers of x up to x^2 and use four-digit precision. Denote your answer by y_3. Again, compute the error relative to y_1.

***P* 1.6.** **(i)** Compute $\ln(1+c) + \ln(1-c)$ for $c = 0.002$ using the full precision available on your calculator (or MATLAB).

(ii) Repeat the computation with reduced precision, rounding each logarithm to six significant digits. What is the relative error (using the answer obtained in **(i)** above as the true value)?

(iii) Compute the same quantity using the Taylor series expansion

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots$$

(valid for $-1 < x \leq 1$) with six-digit precision. What is the relative error in this computation?

(iv) Use the approximate form

$$f''(x) \approx \frac{f(x+\Delta) - 2f(x) + f(x-\Delta)}{\Delta^2}$$

(where f'' denotes second derivative and Δ is small) to obtain a new approximation to $\ln(1+c) + \ln(1-c)$ using six-digit precision. What is the relative error in this computation?

***P* 1.7.** Suppose N is an integer. In infinite precision arithmetic, summing together N numbers, each equal to $1/N$, yields an answer exactly equal to 1. In finite precision arithmetic, that is not necessarily the case.

Try out the following MATLAB script (sequence of commands), after setting (i) $N = 100,000$ ($= 10^5$) and (ii) $N = 131,072$ ($= 2^{17}$). Compute the error relative to the exact answer ($= 1$) in each case. Explain any differences between the two cases.

```
y = 0;
for i = 1:N
y = y+(1/N);
end
y
```

Section 1.3

***P* 1.8.** Which of the following expressions represents the complex number $2 + 3j$ in MATLAB?

```
2 + 3*j
2 + 3*i
2 + j3
2 + 3j
[2+3j]
[2 + 3j]
[2 +3j]
```

***P* 1.9.** Simplify the complex fraction

$$\frac{(1 + j\sqrt{3})(1 - j)}{(\sqrt{3} + j)(1 + j)}$$

using (i) Cartesian forms; and (ii) polar forms, throughout your calculation.

***P* 1.10.** Let N be an arbitrary positive integer. Evaluate the product

$$\prod_{k=1}^{N-1} \left(\cos\left(\frac{k\pi}{N}\right) + j \sin\left(\frac{k\pi}{N}\right) \right) \ ,$$

expressing your answer in Cartesian form.

Section 1.4

***P* 1.11.** Consider the continuous-time sinusoid

$$x(t) = 5\cos(500\pi t + 0.25)$$

where t is in seconds.

(i) What is the first value of t greater than 0 such that $x(t) = 0$?

(ii) Consider the following MATLAB script which generates a discrete approximation to $x(t)$:

```
t = 0 : 0.0001 : 0.01 ;
x = 5*cos(500*pi*t + 0.25) ;
```

For which values of `n`, if any, is `x(n)` zero?

***P* 1.12.** The value of the continuous-time sinusoid $x(t) = A\cos(\Omega t + \phi)$ (where $A > 0$ and $0 \le \phi < 2\pi$) is between -2.0 and $+2.0$ for 70% of its period.

(i) What is the value of A?

(ii) If it takes 300 ms for the value of $x(t)$ to rise from -2.0 to $+2.0$, what is the value of Ω?

(iii) If $t = 40$ ms is the first positive time for which $x(t) = -2.0$ and $x'(t)$ (the first derivative) is negative, what is the value of ϕ?

***P* 1.13.** The input voltage $v(t)$ to a light-emitting diode circuit is given by $A\cos(\Omega t + \phi)$, where $A > 0$, $\Omega > 0$ and ϕ are unknown parameters. The circuit is designed in such a way that the diode turns on at the moment the input voltage exceeds $A/2$, and turns off when the voltage falls below $A/2$.

(i) What percentage of the time is the diode on?

(ii) Suppose the voltage $v(t)$ is applied to the diode at time $t = 0$. The diode turns on instantly, turns off at $t = 1.5$ ms, then turns on again at $t = 9.5$ ms. Based on this information, determine Ω and ϕ.

***P* 1.14.** Using stationary phasors, express each of the following sums in the form $A\cos(\Omega t + \phi)$.

(i)

$$\cos\left(7\pi t - \frac{\pi}{6}\right) - \sin\left(7\pi t - \frac{\pi}{6}\right) + 3\cos\left(7\pi t + \frac{\pi}{4}\right)$$

(ii)

$$2.26\cos(43t + 0.11) - 5.77\cos(43t + 2.08) + 0.49\cos(43t + 1.37)$$

Section 1.5

***P* 1.15.** **(i)** For exactly *one* value of ω in $[0,\pi]$, the discrete-time sinusoid

$$v[n] = A\cos(\omega n + \phi)$$

is periodic with period equal to $N = 4$ time units. What is that value of ω?
(ii) For that value of ω, suppose the first period of $v[n]$ is given by

$$v[0] = 1, \qquad v[1] = 1, \qquad v[2] = -1 \qquad \text{and} \qquad v[3] = -1$$

What are the values of $A > 0$ and ϕ?

***P* 1.16.** **(i)** Use the trigonometric identity

$$\cos(\alpha + \beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta$$

to show that

$$\cos(\omega(n+1) + \phi) + \cos(\omega(n-1) + \phi) = 2\cos(\omega n + \phi)\cos\omega$$

(ii) Suppose

$$x[1] = 1.7740, \quad x[2] = 3.1251 \quad \text{and} \quad x[3] = 0.4908$$

are three consecutive values of the discrete-time sinusoid $x[n] = A\cos(\omega n + \phi)$, where $A > 0$, $\omega \in [0,\pi]$ and $\phi \in [0, 2\pi]$. Use the equation derived in **(i)** to evaluate ω. Then use the ratio $x[2]/x[1]$ together with the given identity for $\cos(\alpha + \beta)$ to evaluate $\tan\phi$ and hence ϕ. Finally, determine A.

***P* 1.17.** **(i)** Use MATLAB to plot four periods of the discrete-time sinusoid

$$x_1[n] = \cos\left(\frac{7\pi n}{9} + \frac{\pi}{6}\right)$$

(ii) Show that the product

$$x_2[n] = x_1[n] \cdot \cos(\pi n)$$

is also a (real-valued) discrete-time sinusoid. Express it in the form $A\cos(\omega n + \phi)$, where $A > 0$, $\omega \in [0,\pi]$ and $\phi \in (0, 2\pi)$.

Section 1.6

***P* 1.18.** The continuous-time sinusoid

$$x(t) = \cos(2\pi f t + \phi)$$

is sampled every $T_s = 1/(12f)$ seconds to produce the discrete-time sinusoid

$$x[n] = x(nT_s)$$

(i) Write an expression for $x[n]$. What is the angular frequency ω of $x[n]$, in radians?

(ii) Consider the discrete-time signal

$$y[n] = x[n+1] + 2x[n] + x[n-1]$$

Using phasors, express $y[n]$ in the form

$$y[n] = A\cos(\omega n + \psi)$$

clearly showing the values of A and ψ.

***P* 1.19.** The continuous-time sinusoid

$$x(t) = \cos(150\pi t + \phi)$$

is sampled every $T_s = 3.0$ ms starting at $t = 0$. The resulting discrete-time sinusoid is

$$x[n] = x(nT_s)$$

(i) Express $x[n]$ in the form $x[n] = \cos(\omega n + \phi)$ i.e., determine the value of ω.

(ii) Is the discrete-time sinusoid $x[n]$ periodic? If so, what is its period?

(iii) Suppose that the sampling rate $f_s = 1/T_s$ is variable. For what values of f_s is $x[n]$ constant for all n? For what values of f_s does $x[n]$ alternate in value between $-\cos\phi$ and $\cos\phi$?

***P* 1.20.** Consider the continuous-time sinusoid

$$x(t) = 5\cos(200\pi t - 0.8) + 2\sin(200\pi t)$$

where t is in seconds.

(i) Express $x(t)$ in the form

$$x(t) = A\cos(200\pi t + \phi)$$

(ii) Suppose $x(t)$ is sampled at a rate of $f_s = 160$ samples/second. The discrete-time signal $x[n] = x(n/f_s)$ is expressed as

$$x[n] = A\cos(\omega n + \psi)$$

with ω in the interval $[0, \pi]$. What is the value of ω? What is the relationship between ψ and ϕ?

***P* 1.21.** For what frequencies f in the range 0 to 3.0 KHz does the sinusoid

$$x(t) = \cos(2\pi f t)$$

yield the signal

$$x[n] = \cos(0.4\pi n)$$

when sampled at a rate of $f_s = 1/T_s = 800$ samples/sec?

***P* 1.22.** The continuous-time sinusoid

$$x(t) = \cos(300\pi t)$$

is sampled every $T_s = 2.0$ ms, so that

$$x[n] = x(0.002n)$$

(i) For what other values of f in the range 0 Hz to 2.0 KHz does the sinusoid

$$v(t) = \cos(2\pi f t)$$

produce the same samples as $x(t)$ (i.e., $v[\cdot] = x[\cdot]$) when sampled at the same rate?

(ii) If we increase the sampling period T_s (or equivalently, drop the sampling rate), what is the least value of T_s greater than 2 ms for which $x(t)$ yields the same sequence of samples (as for $T_s = 2$ ms)?

Chapter 2

Matrices and Systems

2.1 Introduction to Matrix Algebra

The analysis of signals and linear systems relies on a variety of mathematical tools, different forms of which are suitable for signals and systems of differing complexity. The simplest type of signal we consider is a discrete-time signal of finite duration, which is the same as an n-dimensional vector consisting of real or complex-valued entries. In representing such signals and their transformations resulting from numerical processing, certain concepts and methods of linear algebra are indispensable. Understanding the role played by linear algebra in the analysis of such simple (finite-dimensional) signals also provides a solid foundation for studying more complex models where the signals evolve in discrete or continuous time and are infinite-dimensional.

2.1.1 Matrices and Vectors

A $m \times n$ matrix is an ordered array consisting of m rows and n columns of *elements*:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{21} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

A compact notation for the above matrix is

$$\mathbf{A} = [a_{ij}]_{m \times n} = [a_{ij}]$$

The last form is used whenever the *dimensions* m and n of $\mathbf{A}$ are known or can be easily inferred.

The elements (also called *entries*) of the matrix $\mathbf{A}$ are real or, more generally, complex numbers. We define

$$\begin{aligned} \mathbf{R}^{m \times n} &= \text{space of all } (m \times n)\text{-dimensional } \textit{real}\text{-valued matrices} \\ \mathbf{C}^{m \times n} &= \text{space of all } (m \times n)\text{-dimensional } \textit{complex}\text{-valued matrices} \end{aligned}$$

where $\mathbf{C}$ denotes the complex plane.

A *vector* is a special case of a matrix where one of the dimensions (m or n) equals unity. A *row vector* has the form

$$\begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix}$$

and lies in the space $\mathbf{R}^{1 \times n}$ or, more generally, $\mathbf{C}^{1 \times n}$.

A column vector has the form

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$$

and lies in the space $\mathbf{R}^{m\times 1}$ or, more generally, $\mathbf{C}^{m\times 1}$.

Row vectors and column vectors can be used interchangeably to represent an ordered set of (finitely many) numbers. Thus, in many ways, the spaces $\mathbf{R}^{1\times n}$ and $\mathbf{R}^{n\times 1}$ are no different from $\mathbf{R}^n$, the space of all n-tuples of real numbers. In fact, the redundant dimension (unity) in $\mathbf{R}^{1\times n}$ and $\mathbf{R}^{n\times 1}$ is often omitted when the orientation (row or column) of the vector can be easily deduced from the context. The same holds for $\mathbf{C}$ replacing $\mathbf{R}$.

The usual notation for a vector in $\mathbf{R}^n$ or $\mathbf{C}^n$ is a lower-case boldface letter, e.g.,

$$\mathbf{a} = (a_1, \ldots, a_n)$$

In operations involving matrices, the distinction between row and column vectors is important. *Unless otherwise stated, a lower-case boldface letter will denote a column vector*, i.e.,

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$$

To denote row vectors in expressions involving matrix operations, the transpose (T), or complex-conjugate transpose (H), operators will be used, i.e., a row vector will be denoted by $\mathbf{a}^T$ or $\mathbf{a}^H$.

In MATLAB, matrices are entered row-by-row, where rows are separated with a semicolon:

```
A = [-1 3 4; 0 1 5; 2 2 1]
```

Thus `[-1 3 4]` is a row vector, while `[-1; 0; 2]` is a column vector. The transpose operator T is entered as `.'` and the complex conjugate transpose H as `'`. Both produce the same result in the case of real-valued vectors:

```
[-1; 0; 2].'
```

and

```
[-1; 0; 2]'
```

are the same as

```
[-1 0 2]
```

The default orientation of a vector in MATLAB is horizontal, i.e., a *row*. This means that MATLAB functions returning vector-valued answers from scalar arguments will (usually) do so in row-vector format. An example of this is the notation `start:increment:end`, which generates a *row* vector. MATLAB will ignore the orientation of row and column vectors in places where it is unimportant, e.g., in arguments of certain functions.

An error message will be generated if the orientation of the vector is inconsistent with the matrix computation being performed. A notable exception is the addition of a scalar to a vector, which is interpreted in an element-wise fashion, i.e.,

```
1 + [-1 9 7]
```

results in

```
[0 10 8]
```

and

```
1 - [-1; 9; 7]
```

is the same as

```
[2; -8; -6]
```

2.1.2 Signals as Vectors

A discrete-time signal consisting of *finitely many* samples can be represented as a vector

$$\mathbf{x} = (x_1, \ldots, x_n)$$

where n is the total number of samples. Here, x_i is an abbreviated form of $x[i]$ (in the earlier notation). The choice of $i = 1$ for the initial time is mainly for consistency with standard indexing for matrices. The initial index $i = 0$ will be used at a later point.

From now on, and through the end of Section 2.12, all matrices and vectors will be assumed to be real-valued.

Definition 2.1.1. The *linear combination* of n-dimensional vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(r)}$ with coefficients $c_1, \ldots, c_r$ in $\mathbf{R}$ is defined as the n-dimensional vector

$$\mathbf{x} = c_1\mathbf{x}^{(1)} + \cdots + c_r\mathbf{x}^{(r)} \qquad \square$$

(Recall that multiplication of a vector by a scalar amounts to multiplying each entry of the vector by the scalar; while addition of two or more vectors amounts to adding their respective components.)

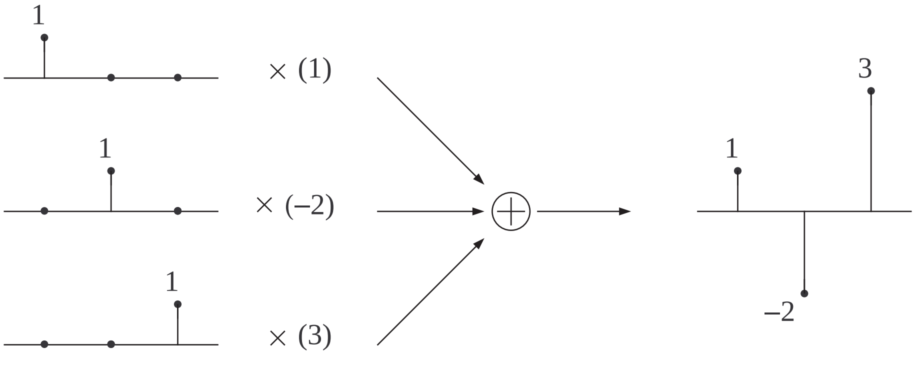
Any vector $\mathbf{x}$ in $\mathbf{R}^n$ can be expressed as a linear combination of the *standard unit vectors* $\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(n)}$, defined by

$$e_j^{(i)} = \begin{cases} 1, & i = j; \\ 0, & i \neq j. \end{cases}$$

Example 2.1.1. Consider the three-dimensional vector $\mathbf{x} = (1, -2, 3)$. Expressing it as a column-vector (the default format for matrix operations), we have

$$\mathbf{x} \;=\; \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix} = (1)\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + (-2)\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + (3)\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \;=\; \mathbf{e}^{(1)} - 2\mathbf{e}^{(2)} + 3\mathbf{e}^{(3)}$$

The linear combination of unit vectors shown above is illustrated graphically. Note that the i^{th} unit vector corresponds to a single pulse of unit height at time i. $\square$



Example 2.1.1

2.1.3 Linear Transformations

The concept of linearity is crucial for the analysis of signals and systems. The most important time-frequency transformations of signals, such as the Laplace, Fourier and z-transforms, are linear. Linear systems form a class which is by far the easiest to study, model and simulate. In fact, in studying the behavior of many nonlinear systems, it is common to use linear approximations and techniques from the theory of linear systems.

The simplest model for a linear system involves a finite-dimensional input signal $\mathbf{x} \in \mathbf{R}^n$ and a finite-dimensional output signal $\mathbf{y} \in \mathbf{R}^m$. The system A, and more precisely,

$$A : \mathbf{R}^n \mapsto \mathbf{R}^m \ ,$$

acts on the input $\mathbf{x}$ to produce the output signal $\mathbf{y}$ represented by

$$\mathbf{y} = A(\mathbf{x})$$

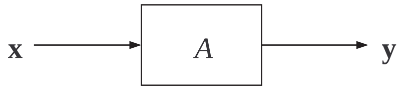


Figure 2.1: A system A with input $\mathbf{x}$ and output $\mathbf{y}$.

Definition 2.1.2. (*Linearity*) We say that the *transformation* (or system) A is linear if, for any two input vectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ and coefficients c_1 and c_2,

$$A(c_1\mathbf{x}^{(1)} + c_2\mathbf{x}^{(2)}) = c_1A(\mathbf{x}^{(1)}) + c_2A(\mathbf{x}^{(2)}) \qquad \square$$

To interpret this property in terms of an actual system, suppose at first that an input signal signal $\mathbf{x}^{(1)}$ is applied to such a system, resulting in a *response* (i.e., output) $\mathbf{y}^{(1)}$. The system is then "reset" and is run again with input $\mathbf{x}^{(2)}$, resulting in a response $\mathbf{y}^{(2)}$. Finally, the system is reset (once again) and run with input

$$\mathbf{x} = c_1\mathbf{x}^{(1)} + c_2\mathbf{x}^{(2)}$$

If the system is linear, the final response of the system will be

$$\mathbf{y} = c_1\mathbf{y}^{(1)} + c_2\mathbf{y}^{(2)}$$

This defining property of linear systems is also known as the *superposition property*, and can be extended to an arbitrary number of input signals:

$$A(c_1\mathbf{x}^{(1)} + \cdots + c_r\mathbf{x}^{(r)}) = c_1A(\mathbf{x}^{(1)}) + \cdots + c_rA(\mathbf{x}^{(r)})$$

As a consequence of the superposition property, knowing the response of a linear system to a number of different inputs allows us to determine and predict its response to any linear combination of those inputs.

Example 2.1.2. Consider the transformation $A : \mathbf{R}^n \mapsto \mathbf{R}^n$ which scales an n-dimensional vector by $\lambda \in \mathbf{R}$:

$$A(\mathbf{x}) = \lambda\mathbf{x}$$

Clearly,

$$\begin{array}{rcl} A(c_1\mathbf{x}^{(1)} + c_2\mathbf{x}^{(2)}) & = & \lambda(c_1\mathbf{x}^{(1)} + c_2\mathbf{x}^{(2)}) \\ & = & c_1(\lambda\mathbf{x}^{(1)}) + c_2(\lambda\mathbf{x}^{(2)}) \\ & = & c_1A(\mathbf{x}^{(1)}) + c_2A(\mathbf{x}^{(2)}) \end{array}$$

and thus the transformation is linear. Scaling is one of the simplest linear transformations of n-dimensional vectors; and in the case $n = 1$, it is the *only* linear transformation. □

Example 2.1.3. Consider the transformation $A : \mathbf{R}^3 \mapsto \mathbf{R}^3$ which cyclically shifts the entries of $\mathbf{x}$ to the right:

$$A(x_1, x_2, x_3) = (x_3, x_1, x_2)$$

We have

$$\begin{array}{rcl} A(c\cdot(u_1, u_2, u_3) + d\cdot(v_1, v_2, v_3)) & = & A(cu_1 + dv_1,\ cu_2 + dv_2,\ cu_3 + dv_3) \\ & = & (cu_3 + dv_3,\ cu_1 + dv_1,\ cu_2 + dv_2) \\ & = & c\cdot(u_3, u_1, u_2) + d\cdot(v_3, v_1, v_2) \\ & = & cA(u_1, u_2, u_3) + dA(v_1, v_2, v_3) \end{array}$$

Thus a cyclical shift is also a linear transformation. □

2.2 Matrix Multiplication

2.2.1 The Matrix of a Linear Transformation

The superposition property of a linear transformation $A : \mathbf{R}^n \mapsto \mathbf{R}^m$ was stated as

$$A(c_1\mathbf{x}^{(1)} + c_2\mathbf{x}^{(2)}) = c_1A(\mathbf{x}^{(1)}) + c_2A(\mathbf{x}^{(2)})$$

We also saw that any n-dimensional vector $\mathbf{x}$ can be expressed as a linear combination of the unit vectors $\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(n)}$:

$$\mathbf{x} = x_1\mathbf{e}^{(1)} + \cdots + x_n\mathbf{e}^{(n)}$$

Thus the result of applying the linear transformation A to *any* vector $\mathbf{x}$ is given by

$$A(\mathbf{x}) = x_1A(\mathbf{e}^{(1)}) + \cdots + x_nA(\mathbf{e}^{(n)})$$

We conclude that *knowing the effect of a linear transformation on each unit vector in $\mathbf{R}^n$ suffices to determine the effect of that transformation on any vector in $\mathbf{R}^n$.*

In systems terms, we can say that *the response of a linear system to an arbitrary input can be computed from the responses of the system to each of the unit vectors in the input signal space.* It follows that the m-dimensional vectors $A(\mathbf{e}^{(1)}), \ldots, A(\mathbf{e}^{(n)})$ completely specify the linear transformation A.

Definition 2.2.1. The matrix of a linear transformation $A : \mathbf{R}^n \mapsto \mathbf{R}^m$ is the $m \times n$ matrix $\mathbf{A}$ whose j^{th} column is given by $A(\mathbf{e}^{(j)})$. □

Example 2.2.1. Suppose the transformation $A : \mathbf{R}^3 \mapsto \mathbf{R}^2$ is such that

$$A(\mathbf{e}^{(1)}) = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad A(\mathbf{e}^{(2)}) = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad \text{and} \quad A(\mathbf{e}^{(3)}) = \begin{bmatrix} 4 \\ -1 \end{bmatrix}$$

Then the matrix of A is given by

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 4 \\ 1 & 0 & -1 \end{bmatrix}$$

Applying A to $\mathbf{x} = (x_1, x_2, x_3)$ produces

$$A(\mathbf{x}) = x_1\begin{bmatrix} 2 \\ 1 \end{bmatrix} + x_2\begin{bmatrix} -1 \\ 0 \end{bmatrix} + x_3\begin{bmatrix} 4 \\ -1 \end{bmatrix} = \begin{bmatrix} 2x_1 - x_2 + 4x_3 \\ x_1 - x_3 \end{bmatrix} \qquad \square$$

2.2.2 The Matrix-Vector Product

Definition 2.2.2. Let $\mathbf{A}$ be a $m \times n$ matrix representing a linear transformation $A : \mathbf{R}^n \mapsto \mathbf{R}^m$, and let $\mathbf{x}$ be a n-dimensional *column* vector. The product $\mathbf{Ax}$ is defined as the result of applying the transformation A to $\mathbf{x}$. In other words, $\mathbf{Ax}$ is the linear combination of the columns of $\mathbf{A}$ with coefficients given by the corresponding entries in $\mathbf{x}$. □

Example 2.2.1. (*Continued.*) If

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 4 \\ 1 & 0 & -1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

then

$$\mathbf{Ax} = \begin{bmatrix} 2x_1 - x_2 + 4x_3 \\ x_1 - x_3 \end{bmatrix}$$

□

A different way of obtaining the product $\mathbf{Ax}$ is by taking the inner (dot) product of each row of $\mathbf{A}$ with $\mathbf{x}$. To see why this is so, let

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Then, for $\mathbf{x} = (x_1, \dots, x_n)$, we have

$$\begin{aligned} \mathbf{Ax} &= x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} \\ &= \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix} \end{aligned}$$

Thus the i^{th} element of the resulting vector $\mathbf{y} = \mathbf{Ax}$ is given by

$$y_i = (\mathbf{Ax})_i = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = \sum_{j=1}^{n} a_{ij}x_j$$

i.e., it equals the inner product of the i^{th} row of $\mathbf{A}$ and $\mathbf{x}$.

2.2.3 The Matrix-Matrix Product

The matrix-vector product can be generalized to the product of two matrices, $\mathbf{A}$ and $\mathbf{B}$, by considering the linear transformations represented by $\mathbf{A}$ and $\mathbf{B}$.

Suppose $A : \mathbf{R}^p \mapsto \mathbf{R}^m$ and $B : \mathbf{R}^n \mapsto \mathbf{R}^p$ are two linear transformations, and consider the *composition* $A \circ B$ defined by

$$(A \circ B)(\mathbf{x}) = A(B(\mathbf{x}))$$

where $\mathbf{x}$ is n-dimensional. Thus B is applied to $\mathbf{x}$ first, to produce a p-dimensional vector $B(\mathbf{x})$; then A is applied to $B(\mathbf{x})$ to produce $(A \circ B)(\mathbf{x})$. In systems terms, we have two linear systems A and B connected in *series* (also *tandem* or *cascade*), as illustrated in Figure 2.2.

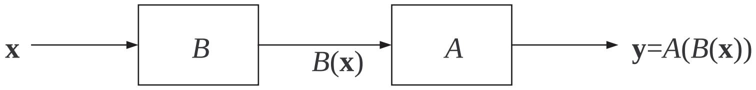


Figure 2.2: Two systems connected in series.

Definition 2.2.3. If $\mathbf{A}$ is the $m \times p$ matrix of the transformation $A : \mathbf{R}^p \mapsto \mathbf{R}^m$ and $\mathbf{B}$ is the $p \times n$ matrix of the transformation $B : \mathbf{R}^n \mapsto \mathbf{R}^p$, then the product $\mathbf{AB}$ is defined as the $m \times n$ matrix of the transformation $A \circ B : \mathbf{R}^n \mapsto \mathbf{R}^m$. □

To obtain $\mathbf{AB}$ in terms of the entries of $\mathbf{A}$ and $\mathbf{B}$, we examine the j^{th} column of $\mathbf{AB}$, which we denote here by $(\mathbf{AB})_{\cdot j}$.

We know that $(\mathbf{AB})_{\cdot j}$ is obtained by applying $A \circ B$ to the j^{th} unit vector $\mathbf{e}^{(j)}$ in the input space $\mathbf{R}^n$, i.e.,

$$(\mathbf{AB})_{\cdot j} = (A \circ B)(\mathbf{e}^{(j)}) = A(B(\mathbf{e}^{(j)}))$$

Since $B(\mathbf{e}^{(j)})$ is the j^{th} column of $\mathbf{B}$ (denoted by $(\mathbf{B})_{\cdot j}$), we have

$$(\mathbf{AB})_{\cdot j} = A((\mathbf{B})_{\cdot j}) = \mathbf{A}(\mathbf{B})_{\cdot j}$$

Thus the j^{th} column of the product $\mathbf{AB}$ is given by the product of $\mathbf{A}$ and the j^{th} column of $\mathbf{B}$:

$$(\mathbf{AB})_{\cdot j} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mp} \end{bmatrix} \begin{bmatrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{pj} \end{bmatrix}$$

It can be seen that the i^{th} entry of $(\mathbf{AB})_{\cdot j}$, which is the same as the $(i,j)^{\text{th}}$ entry of $\mathbf{AB}$, is given by the inner product of the i^{th} *row* of $\mathbf{A}$ and the j^{th} *column* of $\mathbf{B}$.

Fact. *If* $\mathbf{A}$ *is* $m \times p$ *and* $\mathbf{B}$ *is* $p \times n$, *then* $\mathbf{AB}$ *is the* $m \times n$ *matrix whose* $(i,j)^{\text{th}}$ *entry is given by*

$$(\mathbf{AB})_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ip}b_{pj} = \sum_{k=1}^{p} a_{ik}b_{kj} \qquad \square$$

Example 2.2.2. If

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 4 \\ 1 & 0 & -1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{bmatrix} 1 & -2 \\ 0 & 3 \\ 1 & 5 \end{bmatrix}$$

then

$$\mathbf{AB} = \begin{bmatrix} 6 & 13 \\ 0 & -7 \end{bmatrix} \qquad \square$$

Example 2.2.3. If

$$\mathbf{A} = \begin{bmatrix} 4 \\ 1 \\ -2 \end{bmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{bmatrix} 1 & -3 & 2 \end{bmatrix}$$

then

$$\mathbf{AB} = \begin{bmatrix} 4 & -12 & 8 \\ 1 & -3 & 2 \\ -2 & 6 & -4 \end{bmatrix}$$

Note that in this case, the $(i,j)^{\text{th}}$ element of $\mathbf{AB}$ is simply $a_i b_j$; this is because the rows of $\mathbf{A}$ and columns of $\mathbf{B}$ each consist of a single element. □

2.2.4 Associativity and Commutativity

Fact. *Matrix multiplication is associative.* □

If the dimensions of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are such that the products $\mathbf{AB}$ and $\mathbf{BC}$ can be formed, we can also form the product $\mathbf{ABC}$, which is the matrix of the cascaded transformation $A \circ B \circ C$ depicted in Figure 2.3.

Grouping C and B together, we can write the output $\mathbf{y}$ of the cascade as $\mathbf{A}(\mathbf{BC})\mathbf{x}$. Grouping B and A together, we have the equivalent expression $(\mathbf{AB})\mathbf{Cx}$. Thus

$$\mathbf{ABC} = \mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$$

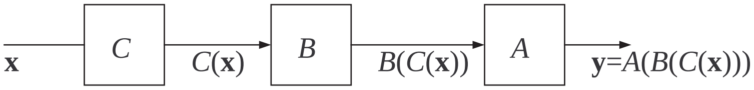


Figure 2.3: Three systems connected in series.

which is also known as the associative property of matrix multiplication. It can be extended to the product of L matrices, where L is arbitrary. The number of matrices in the product can be reduced to $L-1$ by multiplying out any two adjacent matrices without changing their position in the product; this procedure can then be repeated until the product is expressed as a single matrix.

Fact. *Matrix multiplication is* ***not*** *commutative.* □

We say that $\mathbf{A}$ and $\mathbf{B}$ *commute* if

$$\mathbf{AB} = \mathbf{BA}$$

Clearly, the only way the products $\mathbf{AB}$ and $\mathbf{BA}$ can both exist and have the same dimensions is if the matrices $\mathbf{A}$ and $\mathbf{B}$ are both square, i.e., $n \times n$. Still, this is not sufficient for commutativity, as Example 2.2.4 below shows.

Example 2.2.4. Consider the two-dimensional case ($n = 2$). Let $\mathbf{A}$ represent projection of (x_1, x_2) on the horizontal (x_1) axis. The linear transformation A applied to the two unit vectors $(1, 0)$ and $(0, 1)$ yields

$$A(1,0) = (1,0), \qquad A(0,1) = (0,0)$$

and therefore

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Let $\mathbf{B}$ represent a counterclockwise rotation of (x_1, x_2) by an angle θ. We can obtain the elements of $\mathbf{B}$ either from geometry, or by noting that rotation by angle θ amounts to multiplication of the complex number $x_1 + jx_2$ by $e^{j\theta} = \cos\theta + j\sin\theta$. The result of the multiplication is

$$x_1\cos\theta - x_2\sin\theta + j(x_1\sin\theta + x_2\cos\theta) = (x_1\cos\theta - x_2\sin\theta,\ x_1\sin\theta + x_2\cos\theta)$$

and thus

$$\mathbf{B} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Clearly,

$$\mathbf{AB} = \begin{bmatrix} \cos\theta & -\sin\theta \\ 0 & 0 \end{bmatrix}$$

while

$$\mathbf{BA} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \end{bmatrix}$$

Thus the two matrices do not commute except in the trivial case where $\sin\theta = 0$, i.e, $\theta = 0$ or $\theta = \pi$. This can be also seen using a geometrical argument. Briefly, the transformation $A \circ B$ (whose matrix is $\mathbf{AB}$) is a cascade of a rotation (first), followed by projection on the x_1-axis. The result of the transformation is therefore always on the x_1-axis, i.e., it has $x_2 = 0$. On the other hand, projection on the x_1-axis followed by rotation through an angle other than 0 or π will result in a vector with $x_2 \neq 0$. □

Example 2.2.5. If $\mathbf{A}$ and $\mathbf{B}$ are rotations by angles ϕ and θ, respectively, then

$$\mathbf{AB} = \mathbf{BA}$$

i.e., the two matrices commute. This is because the order of the two rotations is immaterial—both $\mathbf{AB}$ and $\mathbf{BA}$ represent a rotation by an angle $\theta + \phi$. □

2.3 More on Matrix Algebra

2.3.1 Column Selection and Permutation

We saw that if $\mathbf{A}$ is a $m \times n$ matrix and $\mathbf{e}^{(j)}$ is the j^{th} unit vector in $\mathbf{R}^{n\times 1}$, then

$$\mathbf{A}\mathbf{e}^{(j)} = (\mathbf{A})_{\cdot j}$$

where $(\mathbf{A})_{\cdot j}$ is the j^{th} column of the matrix $\mathbf{A}$. Thus multiplication of a matrix $\mathbf{A}$ by a (column) unit vector is the same as *column selection*, as illustrated in Example 2.3.1.

Example 2.3.1.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} b \\ e \\ h \end{bmatrix} \qquad \square$$

By appending one or more unit vectors to $\mathbf{e}^{(j)}$, it is possible to select two or more columns from the matrix $\mathbf{A}$. This is based on the following general fact.

Fact. *If*

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}^{(1)} & \mathbf{B}^{(2)} \end{bmatrix}$$

where $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$ are submatrices of $\mathbf{B}$, each having the same number of rows as $\mathbf{B}$, then

$$\mathbf{AB} = \begin{bmatrix} \mathbf{AB}^{(1)} & \mathbf{AB}^{(2)} \end{bmatrix}$$

Similarly, if

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{(1)} \\ \mathbf{A}^{(2)} \end{bmatrix}$$

where $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ are submatrices of $\mathbf{A}$, each having the same number of columns as $\mathbf{A}$, then

$$\mathbf{AB} = \begin{bmatrix} \mathbf{A}^{(1)}\mathbf{B} \\ \mathbf{A}^{(2)}\mathbf{B} \end{bmatrix}$$

Also,

$$\mathbf{AB} = \begin{bmatrix} \mathbf{A}^{(1)}\mathbf{B}^{(1)} & \mathbf{A}^{(1)}\mathbf{B}^{(2)} \\ \mathbf{A}^{(2)}\mathbf{B}^{(1)} & \mathbf{A}^{(2)}\mathbf{B}^{(2)} \end{bmatrix} \qquad \square$$

The formulas above are easily obtained by recalling that the $(i, j)^{\text{th}}$ element of the product $\mathbf{AB}$ is the inner product of the i^{th} row of $\mathbf{A}$ and the j^{th} column of $\mathbf{B}$. They can be extended to horizontal partitions of $\mathbf{A}$ and vertical partitions of $\mathbf{B}$ involving an arbitrary number of submatrices.

Choosing $\mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots$ as unit vectors in $\mathbf{R}^{n\times 1}$, we obtain a matrix consisting of columns of $\mathbf{A}$. This is illustrated in Example 2.3.2 below.

Example 2.3.2.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} b & b \\ e & e \\ h & h \end{bmatrix}$$

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} b & c & a \\ e & f & d \\ h & i & g \end{bmatrix}$$

□

Note that the second product in Example 2.3.2 resulted in a matrix with the same columns as those of $\mathbf{A}$, but permuted. This is because

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

consists of all the unit (column) vectors in arbitrary order. This type of matrix is known as a *permutation* matrix.

Definition 2.3.1. A $n \times n$ matrix $\mathbf{P}$ is a permutation matrix if all rows and columns of $\mathbf{P}$ are unit vectors in $\mathbf{R}^n$.

Two equivalent conditions are:

- the rows of $\mathbf{P}$ are the unit vectors in $\mathbf{R}^{1\times n}$ (in any order); and
- the columns of $\mathbf{P}$ are the unit vectors in $\mathbf{R}^{n\times 1}$ (in any order). □

The product $\mathbf{AP}$ is a matrix whose columns are the same as those of $\mathbf{A}$, ordered in the same way as the unit vectors are in the columns of $\mathbf{P}$. Note that $\mathbf{A}$ can be any $m \times n$ matrix, i.e., it *need not be square.*

2.3.2 Matrix Transpose

If $\mathbf{A}$ is a $m \times n$ matrix, then $\mathbf{A}^T$ is the $n \times m$ matrix whose rows are the columns of $\mathbf{A}$ (equivalently, whose columns are the rows of $\mathbf{A}$). $\mathbf{A}^T$ is known as the *transpose* of $\mathbf{A}$, and T is known as the transpose operator.

Definition 2.3.2. If $\mathbf{A} = [a_{ij}]$, then $\mathbf{A}^T$ is defined by

$$(\mathbf{A}^T)_{ij} = (\mathbf{A})_{ji} = a_{ji}$$

for every pair (i, j). □

Example 2.3.3.

$$\begin{aligned}
\begin{bmatrix} a & b & c \end{bmatrix}^T &= \begin{bmatrix} a \\ b \\ c \end{bmatrix} \\
\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^T &= \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix} \\
\begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix}^T &= \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix}
\end{aligned}$$

□

Note that the third transposition in Example 2.3.3 above resulted in the same (square) matrix. This was a consequence of symmetry about the leading diagonal.

Definition 2.3.3. A $n \times n$ matrix $\mathbf{A}$ is *symmetric* if $\mathbf{A} = \mathbf{A}^T$. Equivalent conditions are:

- for every (i, j), $a_{ij} = a_{ji}$; and
- for every i, the i^{th} row of $\mathbf{A}$ equals (the transpose of) its i^{th} column. □

Fact.

$$(\mathbf{A}^T)^T = \mathbf{A}$$

□

Fact. *Assuming the product* $\mathbf{AB}$ *exists,*

$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$$

To prove this fact, note that the $(i, j)^{\text{th}}$ element of $(\mathbf{AB})^T$ is the $(j, i)^{\text{th}}$ element of $\mathbf{AB}$. This equals the inner product of the j^{th} row of $\mathbf{A}$ and the i^{th} column of $\mathbf{B}$; which is the same as the inner product of the the j^{th} column of $\mathbf{A}^T$ and the i^{th} row of $\mathbf{B}^T$. This is just the $(i, j)^{\text{th}}$ element of $\mathbf{B}^T\mathbf{A}^T$. □

2.3.3 Row Selection

As usual, let $\mathbf{A}$ be a $m \times n$ matrix. If $\mathbf{e}^{(i)}$ is the i^{th} unit vector in $\mathbf{R}^{m\times 1}$, then $(\mathbf{e}^{(i)})^T$ is also the i^{th} unit vector in $\mathbf{R}^{1\times m}$. The product

$$(\mathbf{e}^{(i)})^T\mathbf{A}$$

is a row vector whose transpose equals

$$((\mathbf{e}^{(i)})^T\mathbf{A})^T = \mathbf{A}^T\mathbf{e}^{(i)}$$

namely the i^{th} column of the $n\times m$ matrix $\mathbf{A}^T$. As we know, this is the same as the i^{th} row of $\mathbf{A}$. We therefore see that left multiplication of a matrix $\mathbf{A}$ by a *row* unit vector results in selecting a *row* from $\mathbf{A}$.

Example 2.3.4.

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} d & e & f \end{bmatrix} \qquad \square$$

Similarly, left-multiplying $\mathbf{A}$ by a $m \times m$ permutation matrix results in a permutation of the *rows* or $\mathbf{A}$. That is, $\mathbf{PA}$ is a matrix whose rows are the same as those of $\mathbf{A}$, ordered in the same way as the unit vectors are ordered in the rows of $\mathbf{P}$. (Note that this ordering is *not* necessarily the same as that of the unit vectors in the *columns* of $\mathbf{P}$.)

Example 2.3.5.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} d & e & f \\ a & b & c \\ g & h & i \end{bmatrix} \qquad \square$$

Fact. *If* $\mathbf{P}$ *is a permutation matrix, then so is* $\mathbf{P}^T$. *Furthermore,*

$$\mathbf{P}^T\mathbf{P} = \mathbf{I}$$

where

$$I_{ij} = \begin{cases} 1, & i = j; \\ 0, & i \neq j. \end{cases}$$

The first statement follows directly from the definition of a permutation matrix. To prove the second statement, note that the $(i,j)^{\text{th}}$ element of $\mathbf{P}^T\mathbf{P}$ equals the inner product of the i^{th} and j^{th} columns of $\mathbf{P}$. Since the columns of $\mathbf{P}$ are distinct unit vectors, it follows that the inner product equals unity if $i = j$, zero otherwise. □

2.4 Matrix Inversion and Linear Independence

2.4.1 Inverse Systems and Signal Representations

We used the equation

$$\mathbf{y} = A(\mathbf{x}) = \mathbf{A}\mathbf{x}$$

to describe the response $\mathbf{y}$ of a linear system A to an input signal $\mathbf{x}$. In general, the matrix $\mathbf{A}$ is of size $m \times n$, which means that the input vector $\mathbf{x}$ is n-dimensional and the output vector $\mathbf{y}$ is m-dimensional.

In almost all cases, the output signal $\mathbf{y}$ contains useful information about the input signal $\mathbf{x}$, which may otherwise be "hidden" (i.e., not directly accessible). The natural question to ask is whether this information is complete; in other words, whether it possible to recover $\mathbf{x}$ from $\mathbf{y}$. The transformation $\mathbf{y} \to \mathbf{x}$, if properly defined, would be the *inverse* system A^{-1}.

A careful approach to this (inversion) problem must first address the following two issues:

- *Existence* of an inverse: Given any $\mathbf{y}$ in the output space ($\mathbf{R}^m$), can we find a vector $\mathbf{x}$ in the input space ($\mathbf{R}^n$) such that $\mathbf{y} = A(\mathbf{x})$? If not, for which vectors $\mathbf{y}$ in $\mathbf{R}^m$ is this possible?

- *Uniqueness* of the inverse: If, for a particular vector $\mathbf{y}$ in $\mathbf{R}^m$, there exists $\mathbf{x}$ in $\mathbf{R}^n$ such that $\mathbf{y} = A(\mathbf{x})$, is $\mathbf{x}$ unique? Or do there exist other vectors $\mathbf{x}' \neq \mathbf{x}$ which also satisfy $\mathbf{y} = A(\mathbf{x}')$?

As we will soon see, the answers to questions posed above (for a given system A) determine whether the inverse system A^{-1} can be defined in a meaningful way. How that system can be obtained practically from A is a separate problem which we will discuss later.

Inversion of linear transformations is also important in signal analysis. Consider a $m \times n$ matrix $\mathbf{V}$ consisting of columns $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}$:

$$\mathbf{V} = \left[\begin{array}{ccc} \mathbf{v}^{(1)} & \ldots & \mathbf{v}^{(n)} \end{array}\right]$$

If the column vectors are taken as reference signals (e.g., discrete-time sinusoids in Fourier analysis), a natural question to ask is whether an arbitrary signal $\mathbf{s}$ in $\mathbf{R}^m$ can be expressed as a linear combination of these reference signals, i.e., whether there exists a vector $\mathbf{c}$ of coefficients $c_1, \ldots, c_n$ such that

$$\mathbf{s} = \sum_{r=1}^{n} c_r \mathbf{v}^{(r)}$$

or equivalently,

$$\mathbf{s} = \mathbf{V}\mathbf{c}$$

This problem of *signal representation* is analogous to the system inversion problem posed earlier, with a change of notation from $\mathbf{y} = \mathbf{A}\mathbf{x}$ to $\mathbf{s} = \mathbf{V}\mathbf{c}$. In particular, the existence and uniqueness questions formulated earlier are also relevant in signal representation.

In some applications, the set of reference signals $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}$ may not be sufficiently large to provide a representation for every signal $\mathbf{s}$ in $\mathbf{R}^m$, i.e., the answer to the existence question is negative. In such cases, we are often interested in the best *approximation* to a signal $\mathbf{s}$ in terms of a linear combination of the columns of $\mathbf{V}$. In other words, we are interested in minimizing some function of the error vector

$$\mathbf{V}\mathbf{c} - \mathbf{s}$$

by choice of $\mathbf{c}$. We will later study the solution to this problem in the case where the function chosen for that purpose is the sum of squares of the error vector entries.

2.4.2 Range of a Matrix

The *range*, or *column space*, of a $m \times n$ matrix $\mathbf{A}$ is defined as the set of all linear combinations of its columns, and is denoted by $\mathcal{R}(\mathbf{A})$. Thus

$$\mathcal{R}(\mathbf{A}) = \{\mathbf{A}\mathbf{c} : \mathbf{c} \in \mathbf{R}^{n\times 1}\}$$

$\mathcal{R}(\mathbf{A})$ is also the range of the transformation A, which maps $\mathbf{R}^n$ into $\mathbf{R}^m$. Clearly,

$$\mathcal{R}(\mathbf{A}) \subset \mathbf{R}^{m\times 1}$$

The concept of range allows us to reformulate the existence and uniqueness questions of the previous section as follows:

- *Existence* of an inverse: What is the range $\mathcal{R}(\mathbf{A})$? Does it coincide with $\mathbf{R}^{m\times 1}$, or is it a proper subset of $\mathbf{R}^{m\times 1}$?

- *Uniqueness* of an inverse: For $\mathbf{y} \in \mathcal{R}(\mathbf{A})$, does the set of vectors $\mathbf{x} \in \mathbf{R}^{n\times 1}$ satisfying $\mathbf{y} = \mathbf{A}\mathbf{x}$ consist of a single element? Or does it contain two or more vectors?

In what follows, we will examine the properties of the range of a *square* (i.e., $n \times n$) matrix $\mathbf{A}$, and argue that the answers to the existence and uniqueness questions posed above are either *both* affirmative or *both* negative. Our discussion will involve the concept of *linear independence* of a set of vectors, and will also allow us to draw certain conclusions about matrices of arbitrary size $m \times n$.

2.4.3 Geometrical Interpretation of the Range

$\mathcal{R}(\mathbf{A})$ is a *linear subspace* of $\mathbf{R}^{m\times 1}$. Here, "linear" means that all linear combinations of vectors in $\mathcal{R}(\mathbf{A})$ also lie in $\mathcal{R}(\mathbf{A})$. To see why this has to be so, take two vectors in $\mathcal{R}(\mathbf{A})$, say $\mathbf{Ac}^{(1)}$ and $\mathbf{Ac}^{(2)}$, and form their linear combination $\lambda\mathbf{Ac}^{(1)} + \mu\mathbf{Ac}^{(2)}$. This can be written as $\mathbf{A}(\lambda\mathbf{c}^{(1)} + \mu\mathbf{c}^{(2)})$, and therefore also lies in $\mathcal{R}(\mathbf{A})$.

The largest Euclidean space $\mathbf{R}^{m\times 1}$ which we comfortably visualize is the three-dimensional one, i.e., $\mathbf{R}^{3\times 1}$. It has four types of subspaces, categorized by their dimension d:

- $\underline{d=0}$: Only one such subspace exists, consisting of a single point, namely the origin $\mathbf{0} = [0\ 0\ 0]^T$.

- $\underline{d=1}$: Straight lines through the origin.

- $\underline{d=2}$: Planes through the origin.

- $\underline{d=3}$: Only one such subspace exists, namely $\mathbf{R}^{3\times 1}$ itself.

The range of a $3 \times n$ matrix will fall into one of the above categories, i.e., it will have dimension 0, 1, 2 or 3. To see how algebraic relationships between columns affect the dimension of the range, let us focus on the special case of a square (3×3) matrix $\mathbf{A}$ given by

$$\mathbf{A} = \left[\begin{array}{ccc} \mathbf{a}^{(1)} & \mathbf{a}^{(2)} & \mathbf{a}^{(3)} \end{array}\right]$$

The range of the first column $\mathbf{a}^{(1)}$ consists of all vectors of the form $c_1\mathbf{a}^{(1)}$, where $c_1 \in \mathbf{R}$. As c_1 varies, we obtain a full line through the origin *unless* $\mathbf{a}^{(1)} = \mathbf{0}$. Thus with only one exception, the range of a single column vector is one-dimensional, as illustrated in Figure 2.4.

Clearly, the same statements can be made about the range of the second column $\mathbf{a}^{(2)}$. If we now consider the range of the matrix $[\mathbf{a}^{(1)}\ \mathbf{a}^{(2)}]$, we observe the following: *provided* $\mathcal{R}(\mathbf{a}^{(2)})$ *and* $\mathcal{R}(\mathbf{a}^{(2)})$ *are two distinct lines through the origin*, linear combinations of $\mathbf{a}^{(1)}$ and $\mathbf{a}^{(2)}$ will produce a linear subspace of higher dimension, i.e., a plane. In fact, $\mathcal{R}([\mathbf{a}^{(1)}\ \mathbf{a}^{(2)}])$ is the

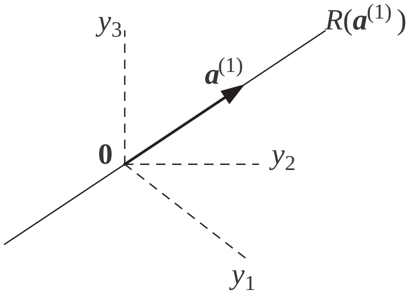


Figure 2.4: The range of a single nonzero vector $\mathbf{a}^{(1)}$ is a straight line through the origin.

unique plane which contains the lines $\mathcal{R}(\mathbf{a}^{(2)})$ and $\mathcal{R}(\mathbf{a}^{(2)})$, as illustrated in Figure 2.5 (on the left).

It is important to note that any vector $\mathbf{y}$ on the plane $\mathcal{R}([\mathbf{a}^{(1)}\ \mathbf{a}^{(2)}])$ has a unique representation as a linear combination of $\mathbf{a}^{(1)}$ and $\mathbf{a}^{(2)}$. This is illustrated in Figure 2.5 (on the right), where a line parallel to $\mathbf{a}^{(2)}$ is drawn through $\mathbf{y}$ to intersect the line containing $\mathbf{a}^{(1)}$. This results in two vectors, $c_1\mathbf{a}^{(1)}$ and $c_2\mathbf{a}^{(2)}$, and the unique representation

$$\mathbf{y} = c_1\mathbf{a}^{(1)} + c_2\mathbf{a}^{(2)}$$

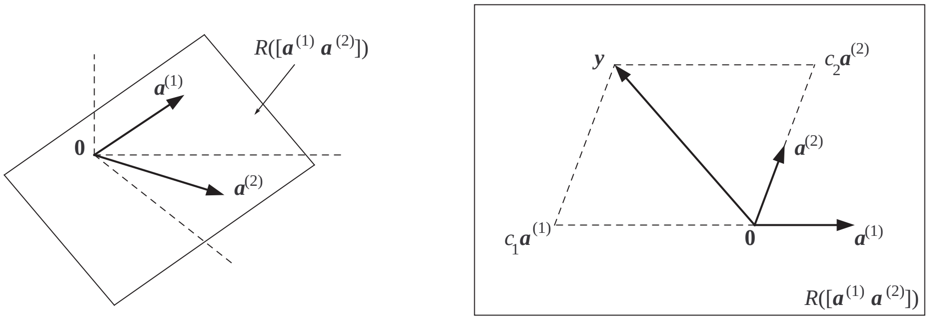

$R([\boldsymbol{a}^{(1)}\ \boldsymbol{a}^{(2)}])$
$\boldsymbol{a}^{(1)}$
$\mathbf{0}$
$\boldsymbol{a}^{(2)}$
$\boldsymbol{y}$
$c_2\boldsymbol{a}^{(2)}$
$\boldsymbol{a}^{(2)}$
$c_1\boldsymbol{a}^{(1)}$
$\mathbf{0}$
$\boldsymbol{a}^{(1)}$
$R([\boldsymbol{a}^{(1)}\ \boldsymbol{a}^{(2)}])$

Figure 2.5: The range of $[\mathbf{a}^{(1)}\ \mathbf{a}^{(2)}]$ is a plane through the origin (left). Any point on that plane has a unique representation as a linear combination of $\mathbf{a}^{(1)}$ and $\mathbf{a}^{(2)}$ (right).

As pointed out earlier, $\mathcal{R}([\mathbf{a}^{(1)}\ \mathbf{a}^{(2)}])$ is two-dimensional if and only if $\mathcal{R}(\mathbf{a}^{(1)})$ and $\mathcal{R}(\mathbf{a}^{(2)})$ are two distinct lines through the origin. This is the same as saying that neither $\mathbf{a}^{(1)}$ nor $\mathbf{a}^{(2)}$ is an all-zeros vector; and, in addition, the two vectors are not multiples of each other. These conditions can be concisely stated as follows: *The only linear combination*

$$c_1\mathbf{a}^{(1)} + c_2\mathbf{a}^{(2)}$$

which equals the all-zeros vector $\mathbf{0} = [0\ 0\ 0]^T$ *is the one with* $c_1 = c_2 = 0$.

Finally, let us consider the third column $\mathbf{a}^{(3)}$. Arguing in the same way as before, we see that, *provided* $\mathcal{R}([\mathbf{a}^{(1)}\ \mathbf{a}^{(2)}])$ *is a (two-dimensional) plane that does not contain* $\mathbf{a}^{(3)}$, linear combinations of vectors in $\mathcal{R}([\mathbf{a}^{(1)}\ \mathbf{a}^{(2)}])$ with $\mathbf{a}^{(3)}$ will produce a linear subspace of higher dimension, namely $\mathbf{R}^{3\times 1}$ itself. Furthermore, any vector in $\mathbf{R}^{3\times 1}$ will have a unique representation of the form

$$\mathbf{y} = c_1\mathbf{a}^{(1)} + c_2\mathbf{a}^{(2)} + c_3\mathbf{a}^{(3)}$$

This is illustrated in Figure 2.6 (on the right), where a line parallel to $\mathbf{a}^{(3)}$ is drawn through $\mathbf{y}$ to intersect the plane $\mathcal{R}([\mathbf{a}^{(1)}\ \mathbf{a}^{(2)}])$. We thus determine c_3, and, proceeding as before (i.e., as in the two-dimensional case), we also determine c_1 and c_2.

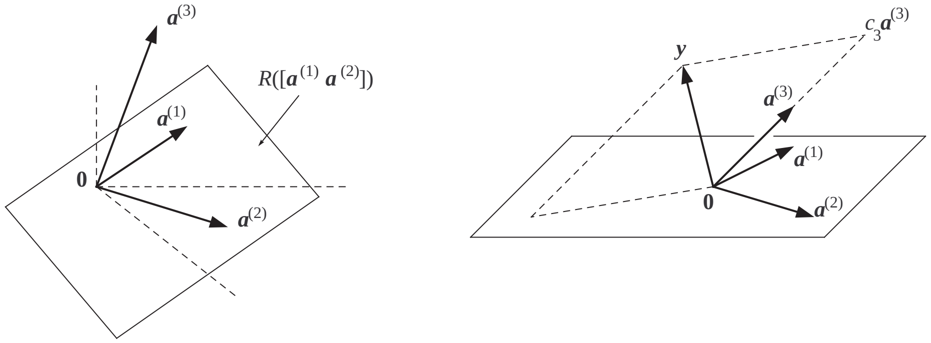


Figure 2.6: The range of the matrix $[\mathbf{a}^{(1)}\ \mathbf{a}^{(2)}\ \mathbf{a}^{(3)}]$ is the entire three-dimensional space $\mathbf{R}^{3\times 1}$ (left). Any point in$\mathbf{R}^{3\times 1}$ has a unique representation as a linear combination of $\mathbf{a}^{(1)}$, $\mathbf{a}^{(2)}$ and $\mathbf{a}^{(3)}$ (right).

In conclusion, $\mathcal{R}(\mathbf{A}) = \mathbf{R}^{3\times 1}$ if and only if *none of the columns of* $\mathbf{A}$ *can be expressed as a linear combination of the others*; this condition also prohibits any of the columns from being an all-zeros vector or a multiple of another column. An equivalent way of stating this is as follows: *The only linear combination*

$$c_1\mathbf{a}^{(1)} + c_2\mathbf{a}^{(2)} + c_3\mathbf{a}^{(3)}$$

which equals the all-zeros vector $\mathbf{0}$ *is the one where* $c_1 = c_2 = c_3 = 0$*.*

We have thus arrived at one of the most important concepts in linear algebra, that of *linear independence*. We restate it for an arbitrary set of m-dimensional vectors.

Definition 2.4.1. The vectors $\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(n)}$ in $\mathbf{R}^{m\times 1}$ are *linearly independent* if the only linear combination

$$c_1\mathbf{a}^{(1)} + \cdots + c_n\mathbf{a}^{(n)}$$

which equals the all-zeros vector $\mathbf{0}$ is the one where $c_1 = \ldots = c_n = 0$. □

An equivalent definition of linear independence in terms of the $m \times n$ matrix

$$\mathbf{A} = \left[\begin{array}{ccc} \mathbf{a}^{(1)} & \ldots & \mathbf{a}^{(n)} \end{array}\right]$$

is the following: The columns of $\mathbf{A}$ are linearly independent if the only solution to the equation

$$\mathbf{A}\mathbf{c} = \mathbf{0}$$

is the all-zeros vector $\mathbf{c} = \mathbf{0}$ (in $\mathbf{R}^{n\times 1}$).

2.5 Inverse of a Square Matrix

2.5.1 Nonsingular Matrices

In the previous section, we showed that linear independence of the columns of a 3×3 matrix $\mathbf{A}$ is a necessary and sufficient condition for the range (or column space) $\mathcal{R}(\mathbf{A})$ of $\mathbf{A}$ to have dimension $d = 3$, i.e., coincide with $\mathbf{R}^{3\times 1}$. Since $\mathcal{R}(\mathbf{A})$ is the set of vectors $\mathbf{y} \in \mathbf{R}^{3\times 1}$ for which the equation

$$\mathbf{Ax} = \mathbf{y}$$

has a solution (given by $\mathbf{x}$), it follows that linear independence is a necessary and sufficient condition for the above equation to have a solution for *every* $\mathbf{y}$ in $\mathbf{R}^{3\times 1}$.

By extension, the same argument can be applied to an arbitrary $n \times n$ matrix. In other words, linear independence of the columns of $\mathbf{A}$ is a necessary and sufficient condition for $\mathcal{R}(\mathbf{A})$ to coincide with $\mathbf{R}^{n\times 1}$, and for the above equation to have a solution for every $\mathbf{y}$ in $\mathbf{R}^{n\times 1}$. Thus linear independence provides an answer to the question of *existence* of an inverse.

The uniqueness of the inverse can be also addressed using the concept of linear independence. Using the argument made for the case $n = 3$, we conclude that if $\mathcal{R}(\mathbf{A}) = \mathbf{R}^{n\times 1}$, then every vector in $\mathbf{R}^{n\times 1}$ has a unique representation as a linear combination of columns of $\mathbf{A}$, i.e., the above equation has a unique solution. If, on the other hand, $\mathcal{R}(\mathbf{A})$ has dimension $d < n$, meaning that the columns of $\mathbf{A}$ are linearly *dependent*, then at least one column of $\mathbf{A}$ lies on the subspace generated by the remaining columns and can thus be expressed as a linear combination of those columns. Any point in that subspace will have *infinitely many* representations. As an example, take the case $n = 3$ and suppose that

$$c_1\mathbf{a}^{(1)} + c_2\mathbf{a}^{(2)} + c_3\mathbf{a}^{(3)} = \mathbf{0}$$

where $\mathbf{c}$ is such that (say) $c_3 \neq 0$. Then

$$\mathbf{a}^{(3)} = -\frac{\lambda c_1}{c_3}\mathbf{a}^{(1)} - \frac{\lambda c_2}{c_3}\mathbf{a}^{(2)} + (1-\lambda)\mathbf{a}^{(3)}$$

for *every* value of $\lambda \in \mathbf{R}$, showing that $\mathbf{a}^{(3)}$ (and by extension, every linear combination of $\mathbf{a}^{(1)}$, $\mathbf{a}^{(2)}$ and $\mathbf{a}^{(3)}$) has infinitely many representations.

In conclusion, linear independence of the columns of a $n \times n$ matrix is a necessary and sufficient condition for the existence of a solution to the equation

$$\mathbf{Ax} = \mathbf{y}$$

for every $\mathbf{y} \in \mathbf{R}^{n\times 1}$, as well as for the uniqueness of that solution.

Definition 2.5.1. A $n \times n$ matrix is *nonsingular* if its columns are linearly independent. It is *singular* if its columns are linearly dependent. □

Example 2.5.1. The 3×3 matrix

$$\mathbf{A} = \begin{bmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{bmatrix}$$

is nonsingular. Note that the first column does not equal the all-zeros vector. Also, the second column cannot be obtained as from the first one by scaling (no scaling of zero can produce a nonzero entry in the second row). Similarly, the third column cannot be obtained as a linear combination of the first and second columns (no linear combination of zeros can produce a nonzero entry in the third row). □

Example 2.5.2. The 3×3 matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 3 \\ -1 & 0 & -1 \\ 0 & 1 & 2 \end{bmatrix}$$

is singular, since

$$\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} + (2) \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix}$$

(A test for nonsingularity will be developed later in conjunction with Gaussian elimination.) □

2.5.2 Inverse of a Nonsingular Matrix

From the foregoing discussion, it follows that a nonsingular matrix $\mathbf{A}$ defines a linear transformation $A : \mathbf{R}^n \mapsto \mathbf{R}^n$ which is a one-to-one correspondence. In particular, an inverse transformation $A^{-1} : \mathbf{R}^n \mapsto \mathbf{R}^n$ also exists, and

$$\mathbf{y} = A(\mathbf{x}) \quad \Leftrightarrow \quad \mathbf{x} = A^{-1}(\mathbf{y})$$

The transformation A^{-1} is also *linear*. To see this, let $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)}$ be any two vectors in $\mathbf{R}^n$, and let

$$\mathbf{x}^{(1)} = A^{-1}(\mathbf{y}^{(1)}) \qquad \text{and} \qquad \mathbf{x}^{(2)} = A^{-1}(\mathbf{y}^{(2)})$$

Since the forward transformation A is linear, we have for any c_1 and c_2,

$$\begin{aligned} A(c_1\mathbf{x}^{(1)} + c_2\mathbf{x}^{(2)}) &= c_1A(\mathbf{x}^{(1)}) + c_2A(\mathbf{x}^{(2)}) \\ &= c_1\mathbf{y}^{(1)} + c_2\mathbf{y}^{(2)} \end{aligned}$$

Therefore

$$\begin{aligned} A^{-1}(c_1\mathbf{y}^{(1)} + c_2\mathbf{y}^{(2)}) &= c_1\mathbf{x}^{(1)} + c_2\mathbf{x}^{(2)} \\ &= c_1A^{-1}(\mathbf{y}^{(1)}) + c_2A^{-1}(\mathbf{y}^{(2)}) \end{aligned}$$

which proves that A^{-1} is a linear transformation. As such, it has a matrix, denoted by $\mathbf{A}^{-1}$.

Definition 2.5.2. If $\mathbf{A}$ is a nonsingular matrix corresponding to a linear transformation A, then $\mathbf{A}^{-1}$ is the matrix of the inverse transformation A^{-1}. □

From the above definition, we have

$$(\mathbf{A}^{-1})^{-1} = \mathbf{A}$$

and we can also evaluate the product $\mathbf{A}\mathbf{A}^{-1}$. Indeed, $\mathbf{A}\mathbf{A}^{-1}$ is the matrix of the cascade $A \circ A^{-1}$. Since

$$A^{-1}(A(\mathbf{x})) = \mathbf{x}$$

for every $\mathbf{x} \in \mathbf{R}^n$, it follows that

$$A^{-1} \circ A = I$$

namely the *identity* transformation. The corresponding matrix is the $n \times n$ *identity matrix*

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

which clearly satisfies

$$\mathbf{I}\mathbf{x} = \mathbf{x}$$

for every $\mathbf{x} \in \mathbf{R}^{n\times 1}$. We conclude that

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

Similarly, $A^{-1} \circ A$ is also an identity system, and therefore

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$$

Either of the last two equations can be used to check whether two given matrices are inverses of each other.

2.5.3 Inverses in Matrix Algebra

Fact. *If* $\mathbf{A}$ *and* $\mathbf{B}$ *are both nonsingular, then so is the product* $\mathbf{AB}$*, and*

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$

This fact can be demonstrated schematically (see Figure 2.7), by noting that the $\mathbf{AB}$ is the matrix of the cascade $A \circ B$. This cascade can be inverted by applying the transformation A^{-1} to the output, followed by B^{-1}.

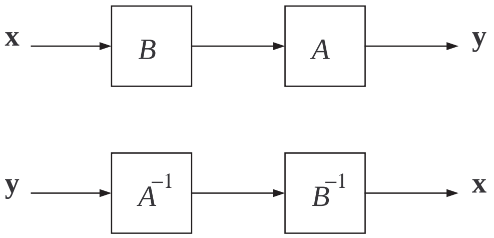


Figure 2.7: Illustration of the identity $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$.

Thus

$$(A \circ B)^{-1} = B^{-1} \circ A^{-1}$$

and consequently

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$

An alternative proof is based on the identities developed in the previous section. We have

$$\mathbf{A}^{-1}(\mathbf{AB}) = (\mathbf{A}^{-1}\mathbf{A})\mathbf{B} = \mathbf{B}$$

where the first equality is due to the associativity of the matrix product. It follows that

$$\mathbf{B}^{-1}\mathbf{A}^{-1}(\mathbf{AB}) = \mathbf{B}^{-1}\mathbf{B} = \mathbf{I}$$

and thus $\mathbf{AB}$ and $\mathbf{B}^{-1}\mathbf{A}^{-1}$ are inverses of each other. □

Fact. *If* $\mathbf{A}$ *is nonsingular, then so is* $\mathbf{A}^T$*, and*

$$(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$$

To prove this identity, recall that

$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$$

Taking $\mathbf{B} = \mathbf{A}^{-1}$, we obtain

$$\mathbf{I}^T = (\mathbf{A}^{-1})^T \mathbf{A}^T$$

But $\mathbf{I}$ is symmetric: $\mathbf{I}^T = \mathbf{I}$. Therefore

$$(\mathbf{A}^{-1})^T \mathbf{A}^T = \mathbf{I}$$

which means that

$$(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$$

□

2.5.4 Nonsingularity of Triangular Matrices

Definition 2.5.3. A *lower triangular* matrix $\mathbf{A}$ is a square matrix defined by

$$(\forall j > i) \quad a_{ij} = 0$$

i.e., all elements above the main diagonal (also known as *superdiagonal* elements) are zero. Similarly, an *upper triangular* matrix $\mathbf{A}$ is a square matrix whose elements below the main diagonal (the *subdiagonal* elements) are zero. □

Triangular matrices have numerous applications in linear algebra, and are particularly useful in solving equations of the form $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is square. The question of invertibility (i.e, nonsingularity) of triangular matrices is important in all such applications. As it turns out, nonsingularity can be readily determined by inspection of the matrix.

Recall how, in Example 2.5.1, we argued that the upper triangular matrix

$$\begin{bmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{bmatrix}$$

is nonsingular. A similar argument could be given for the lower triangular matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & c & 1 \end{bmatrix}$$

but there is no need to do so: a matrix is nonsingular if and only if its transpose is. In what follows, we make a general observation about triangular matrices of arbitrary size.

Fact. *A triangular matrix* $\boldsymbol{A}$ *is nonsingular if and only if all elements on its main diagonal are nonzero, i.e.,* $a_{ii} \neq 0$ *for all* i.

To prove the "if" statement, suppose that all (main) diagonal elements are nonzero. Arguing as in Example 2.5.1, we see immediately that the first column does not equal the all-zeros vector (since $a_{11} \neq 0$). Examining each of the following columns in turn, we see that the j^{th} column cannot be expressed as a linear combination of the first $j-1$ columns since its nonzero diagonal entry a_{jj} would have to be obtained by a linear combination of zeros—which is clearly impossible. Thus the columns of $\mathbf{A}$ are linearly independent, and $\mathbf{A}$ is nonsingular.

To prove the "only if" statement, we argue by contradiction. Suppose that there is at least one zero on the diagonal, and that the leftmost such zero is in the j^{th} column, i.e.,

$$a_{jj} = 0 \qquad \text{and} \qquad (\forall i < j) \quad a_{ii} \neq 0$$

This means that the first j columns of $\mathbf{A}$ will be of the form

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1,j-1} & a_{1j} \\ 0 & a_{22} & a_{23} & \ldots & a_{2,j-1} & a_{2j} \\ 0 & 0 & a_{33} & \ldots & a_{3,j-1} & a_{3j} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & a_{j-1,j-1} & a_{j-1,j} \\ 0 & 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & 0 \end{bmatrix}$$

Since diagonal elements a_{11} through $a_{j-1,j-1}$ are all nonzero, the first $j-1$ columns will be linearly independent (see the "if" argument above). The same will be true of the column segments obtained by discarding the zeros beneath the $(j-1)^{\text{th}}$ row. Since the range of $j-1$ linearly independent vectors in $\mathbf{R}^{j-1}$ is the entire space $\mathbf{R}^{j-1}$, we can obtain the first $j-1$ entries of the j^{th} column by linearly combining the corresponding segments of the first $j-1$ columns. The entire j^{th} column can therefore be obtained by a linear combination of the first $j-1$ columns, and the matrix $\mathbf{A}$ is singular. □

2.5.5 The Inversion Problem for Matrices of Arbitrary Size

The concept of linear independence allows us to investigate the existence and uniqueness of the solution of the equation

$$\mathbf{Ax} = \mathbf{y}$$

where $\mathbf{y} \in \mathbf{R}^{m\times 1}$ is known, $\mathbf{x} \in \mathbf{R}^{n\times 1}$ is unknown and the matrix $\mathbf{A}$ is of size $m \times n$, with $n \neq m$.

In short, depending on whether $m > n$ or $m < n$, and on whether the columns of $\mathbf{A}$ are linearly dependent or independent, a solution may or may not exist for a particular $\mathbf{y} \in \mathbf{R}^{m\times 1}$. If a solution exists *for all* $\mathbf{y} \in \mathbf{R}^{m\times 1}$, then that solution *cannot* be unique. This is in sharp contrast to the case $m = n$, where existence of a solution for all $\mathbf{y} \in \mathbf{R}^{m\times 1}$ is *equivalent* to uniqueness of that solution (and also equivalent to linear independence of the columns of $\mathbf{A}$). It also tells us that defining an inverse matrix $\mathbf{A}^{-1}$ in the case $n \neq m$ can be very tricky—in fact, it is not done.

In more specific terms, the equation $\mathbf{A}\mathbf{x} = \mathbf{y}$ seeks to express the m-dimensional column vector $\mathbf{y}$ as a linear combination of the n columns of $\mathbf{A}$. A solution exists if and only if $\mathbf{y} \in \mathcal{R}(\mathbf{A})$; and it is unique if and only if every $\mathbf{y} \in \mathcal{R}(\mathbf{A})$ is given by a unique linear combination of columns of $\mathbf{A}$, i.e., if and only if the columns of $\mathbf{A}$ are linearly independent. Note that since $\mathbf{A}$ has n columns, the dimension d of $\mathcal{R}(\mathbf{A})$ must satisfy $d \leq n$. And since $\mathcal{R}(\mathbf{A})$ is a subset of $\mathbf{R}^{m\times 1}$, we must also have that $d \leq m$. Thus

$$d \leq \min(m, n).$$

The overdetermined case ($n < m$). Here $d < m$, hence $\mathcal{R}(\mathbf{A})$ is a lower-dimensional subspace of $\mathbf{R}^{m\times 1}$. Thus there are (infinitely many) $\mathbf{y}$'s for which the equation has no solution. For those $\mathbf{y}$'s for which a solution exists (i.e, the $\mathbf{y}$'s in $\mathcal{R}(\mathbf{A})$), the solution is unique if and only if the columns of $\mathbf{A}$ are linearly independent, i.e., $d = n$.

The underdetermined case ($n > m$). The number of columns of $\mathbf{A}$ exceeds m, the dimension of $\mathbf{R}^{m\times 1}$. We know that *at most* m of these columns can be linearly independent, i.e., $d \leq m$. There are two possibilities:

- If $d = m$, then $\mathcal{R}(\mathbf{A}) = \mathbf{R}^{m\times 1}$ and a solution exists for every $\mathbf{y} \in \mathbf{R}^{m\times 1}$. That solution is *not* unique since the $n-m$ redundant columns can be also used in the linear combination $\mathbf{A}\mathbf{x}$ which gives $\mathbf{y}$.

- If $d < m$, then $\mathcal{R}(\mathbf{A})$ is a lower-dimensional subset of $\mathbf{R}^{m\times 1}$. Then a solution cannot exist for every $\mathbf{y}$; if it does exist, it cannot (again) be unique.

The overdetermined case will be examined later on from the viewpoint of signal approximation, i.e, solving for $\mathbf{x}$ which minimizes a function of the error vector $\mathbf{A}\mathbf{x} - \mathbf{y}$.

2.6 Gaussian Elimination

2.6.1 Statement of the Problem

We now focus on solving the equation

$$\mathbf{Ax} = \mathbf{b}$$

where $\mathbf{A}$ is a $n \times n$ matrix, and $\mathbf{x}$ and $\mathbf{b}$ are both n-dimensional column vectors. This (matrix-vector) equation can be expanded into n simultaneous linear equations in the variables $x_1, \dots, x_n$:

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 \\ \vdots & \vdots & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n & = & b_n \end{array}$$

We know that a unique solution $\mathbf{x}$ exists for every $\mathbf{b}$ if and only if $\mathbf{A}$ is nonsingular. The procedure (algorithm) known as *Gaussian elimination* produces the unique solution $\mathbf{x}$ in the case where $\mathbf{A}$ is nonsingular, and can be also used to detect singularity of $\mathbf{A}$.

2.6.2 Outline of Gaussian Elimination

The algorithm is based on the fact that if

$$\mathbf{Ax} = \mathbf{b}$$

and

$$\mathbf{MAv} = \mathbf{Mb}$$

for two nonsingular matrices $\mathbf{A}$ and $\mathbf{M}$, then $\mathbf{x} = \mathbf{v}$. Indeed,

$$\mathbf{v} = (\mathbf{MA})^{-1}\mathbf{Mb} = \mathbf{A}^{-1}\mathbf{M}^{-1}\mathbf{Mb} = \mathbf{A}^{-1}\mathbf{b} = \mathbf{x}$$

where all the inverses in the above equalities are valid by the assumption of nonsingularity. Thus the equations $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{MAx} = \mathbf{Mb}$ have the same solution.

In Gaussian elimination, left-multiplication of both sides of the equation by a nonsingular matrix is performed repeatedly until the matrix on the left is reduced to the $n \times n$ identity:

$$\begin{aligned}
\mathbf{A}\mathbf{x} &= \mathbf{b} \\
\mathbf{M}^{(1)}\mathbf{A}\mathbf{x} &= \mathbf{M}^{(1)}\mathbf{b} \\
\mathbf{M}^{(2)}\mathbf{M}^{(1)}\mathbf{A}\mathbf{x} &= \mathbf{M}^{(2)}\mathbf{M}^{(1)}\mathbf{b} \\
\vdots \quad &\vdots \quad \vdots \\
\mathbf{M}^{(l)}\cdots\mathbf{M}^{(2)}\mathbf{M}^{(1)}\mathbf{A}\mathbf{x} &= \mathbf{M}^{(l)}\cdots\mathbf{M}^{(2)}\mathbf{M}^{(1)}\mathbf{b}
\end{aligned}$$

where

$$\mathbf{M}^{(l)}\cdots\mathbf{M}^{(2)}\mathbf{M}^{(1)}\mathbf{A} = \mathbf{I} \qquad \text{or equivalently,} \qquad \mathbf{M}^{(l)}\cdots\mathbf{M}^{(2)}\mathbf{M}^{(1)} = \mathbf{A}^{-1}$$

The solution is then given by the right-hand side vector:

$$\mathbf{x} = \mathbf{M}^{(l)}\cdots\mathbf{M}^{(2)}\mathbf{M}^{(1)}\mathbf{b}$$

Recall that left multiplication of a matrix by a row vector (i.e., a vector-matrix product) produces a linear combination of the matrix *rows*, with coefficients given by the corresponding entries in the row vector. By considering each row of the matrix $\mathbf{M}$ separately, we see that the product $\mathbf{MA}$ (which has the same dimensions as $\mathbf{A}$) consists of rows that are linear combinations of the rows of $\mathbf{A}$. Thus Gaussian elimination proceeds by a sequence of steps, each step involving certain *row operations* which are “encoded” as a nonsingular matrix $\mathbf{M}^{(j)}$. We should add that, since the same matrix $\mathbf{M}^{(j)}$ also multiplies the right-hand side vector, we effectively take linear combinations of entire *equations* (not just matrix rows).

Gaussian elimination is completed in two phases:

- *forward elimination* (also known as *forward substitution*), followed by
- *backward substitution* (also known as *backward elimination*).

The forward elimination phase is completed in $n-1$ steps. During the j^{th} step, scaled versions of the j^{th} equation are subtracted from the equations below it so as to eliminate the variable x_j from all these equations. The matrix $\mathbf{M}^{(j)}$ is a (nonsingular) *lower triangular* matrix whose form we will soon examine. As we shall see later, in certain situations it may be desirable (or even necessary) to change the order of equations indexed j through n prior to executing the j^{th} step. Such interchanges can be formally dealt with by inserting permutation matrices into the string $\mathbf{M}^{(l)}\cdots\mathbf{M}^{(1)}$; an easier, and more practical, way of representing such permutations will also be discussed.

The backward substitution phase is completed in n steps. The j^{th} step solves for variable x_{n-j+1} using the previously computed values of $x_n, x_{n-1}, \ldots, x_{n-j+2}$. The matrices $\mathbf{M}^{(j)}$ are (nonsingular) *upper triangular* matrices. Different equivalent forms for these matrices will be examined in Section 2.7. The solution of $\mathbf{Ax} = \mathbf{b}$ is thus obtained in a total of $l = 2n - 1$ steps.

2.6.3 Gaussian Elimination Illustrated

Consider the 3×3 equation given by

$$\begin{array}{rcr} 2x_1 + x_2 - x_3 & = & 6 \\ 4x_1 \qquad - x_3 & = & 6 \\ -8x_1 + 2x_2 + 3x_3 & = & -10 \end{array}$$

Since row operations (i.e., multiplication by matrices $\mathbf{M}^{(j)}$) are performed on both sides of the equations, it is convenient to append the vector $\mathbf{b}$ to the matrix $\mathbf{A}$:

$$\mathbf{S} = \left[\begin{array}{cc} \mathbf{A} & \mathbf{b} \end{array} \right]$$

We will use the symbol $\mathbf{S}$ *to denote the initial matrix* $[\mathbf{A} \quad \mathbf{b}]$ *as well as its updates; thus* s_{ij} *will refer to the* $(i,j)^{\text{th}}$ *element in the current matrix* $\mathbf{S}$.

We display $\mathbf{S}$ in a table, with an additional empty column on the left labeled "m" (for "multiplier").

m	x_1	x_2	x_3	b
	2	1	−1	6
	4	0	−1	6
	−8	2	3	−10

We eliminate x_1 from the second row by adding a multiple of the first row (to the second row). By inspection of the first column, the value of the multiplier equals $-(4/2) = -2$. Similarly, x_1 can be eliminated from the third row by adding a multiple of the first row (to the third row), the value of the multiplier being $-(-8)/2 = 4$. These two values are entered in the m column, in their respective rows. The coefficient of x_1 in the first row is also underlined; this coefficient was used to obtain the multiplier for each subsequent row.

m	x_1	x_2	x_3	b
	$\underline{2}$	1	−1	6
−2	4	0	−1	6
4	−8	2	3	−10

Upon adding the appropriate multiples of the first row to the second and third, we obtain

m	x_1	x_2	x_3	b
	2	1	−1	6
	0	−2	1	−6
	0	6	−1	14

To eliminate x_2 from the third row, we add a multiple of the second row (to the third row), the value of the multiplier being $-(6/(-2)) = 3$. As before, we enter the multiplier in the m column, in the third row. The coefficient of x_2 in the second equation is also underlined.

m	x_1	x_2	x_3	b
	2	1	−1	6
	0	$-\underline{2}$	1	−6
3	0	6	−1	14

Upon adding the multiple of the second row to the third one, we obtain

m	x_1	x_2	x_3	b
	2	1	−1	6
	0	−2	1	−6
	0	0	2	−4

At this point we have completed the forward elimination. The resulting equations are

$$\begin{aligned} 2x_1 + x_2 - x_3 &= 6 \\ -2x_2 + x_3 &= -6 \\ 2x_3 &= -4 \end{aligned}$$

In backward substitution, we solve for x_3 (third equation):

$$x_3 = (-4)/2 = -2 \ ;$$

then use the result to solve for x_2 (second equation):

$$x_2 = (-6 - x_3)/(-2) = 2 \ ;$$

and finish by solving for x_1 (first equation):

$$x_1 = (6 - x_2 + x_3)/2 = 1 \ .$$

Thus $\mathbf{x} = [1 \quad 2 \quad -2]^T$.

2.6.4 Row Operations in Forward Elimination

Let $\mathbf{S}$ be the current update of $[\mathbf{A} \quad \mathbf{b}]$, and denote by $(\mathbf{S})_{i\cdot}$ the i^{th} row of $\mathbf{S}$. As we saw in the previous subsection, the j^{th} step in the forward elimination entails replacing each row $(\mathbf{S})_{i\cdot}$ below $(\mathbf{S})_{j\cdot}$ by

$$(\mathbf{S})_{i\cdot} + m_{ij}(\mathbf{S})_{j\cdot}$$

where

$$m_{ij} = -\frac{s_{ij}}{s_{jj}}$$

Rows $(\mathbf{S})_{1\cdot}, \ldots, (\mathbf{S})_{j\cdot}$ are not modified.

These row operations can be carried out using matrix multiplication, namely by updating $\mathbf{S}$ to the product $\mathbf{M}^{(j)}\mathbf{S}$, where

$$\mathbf{M}^{(j)} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & m_{j+1,j} & 1 & & \\ & & & & \vdots & & \ddots & \\ & & & & m_{nj} & & & 1 \end{bmatrix}$$

(only nonzero elements are shown above). Note that the lower triangular matrix $\mathbf{M}^{(j)}$ differs from the identity only in the subdiagonal segment of the j^{th} column. That column segment (given by $(m_{j+1,j}, \ldots, m_{n,j})$) is read off the m column in the Gaussian elimination table.

Thus, in the example of the previous subsection, we had

$$\mathbf{M}^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{M}^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 3 & 1 \end{bmatrix} \quad .$$

2.7 Factorization of Matrices in Gaussian Elimination

2.7.1 *LU* Factorization

Consider again the 3×3 example of Subsection 2.6.3:

$$\begin{aligned} 2x_1 + x_2 - x_3 &= 6 \\ 4x_1 - x_3 &= 6 \\ -8x_1 + 2x_2 + 3x_3 &= -10 \end{aligned}$$

We saw that the row operations performed during the forward elimination phase amounted to left-multiplying the matrix

$$\mathbf{S} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} = \begin{bmatrix} 2 & 1 & -1 & 6 \\ 4 & 0 & -1 & 6 \\ -8 & 2 & 3 & -10 \end{bmatrix}$$

by the lower-triangular matrices

$$\mathbf{M}^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{M}^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 3 & 1 \end{bmatrix}$$

in succession. The update of $\mathbf{S}$ at the end of the forward elimination phase was

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -1 & 6 \\ 4 & 0 & -1 & 6 \\ -8 & 2 & 3 & -10 \end{bmatrix} = \begin{bmatrix} 2 & 1 & -1 & 6 \\ 0 & -2 & 1 & -6 \\ 0 & 0 & 2 & -4 \end{bmatrix}$$

and thus

$$\mathbf{M}^{(2)}\mathbf{M}^{(1)} \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{U} & \mathbf{y} \end{bmatrix}$$

where the matrix

$$\mathbf{U} = \mathbf{M}^{(2)}\mathbf{M}^{(1)}\mathbf{A} = \begin{bmatrix} 2 & 1 & -1 \\ 0 & -2 & 1 \\ 0 & 0 & 2 \end{bmatrix}$$

was *upper triangular.* We then solved

$$\mathbf{U}\mathbf{x} = \mathbf{y}$$

by backward substitution, a procedure which, as we shall soon see, can be also implemented by a series of matrix products.

Recall that triangular matrices are nonsingular if and only if they have no zeros on the main diagonal. Thus both $\mathbf{M}^{(1)}$ and $\mathbf{M}^{(2)}$ are nonsingular in this case, and since $\mathbf{M}^{(2)}\mathbf{M}^{(1)}\mathbf{A} = \mathbf{U}$, we have that

$$\mathbf{A} = (\mathbf{M}^{(1)})^{-1}(\mathbf{M}^{(2)})^{-1}\mathbf{U}$$

In general, the inverse of a (nonsingular) lower triangular matrix is also lower triangular; and the product of two such matrices is also lower triangular. Thus we can write

$$\mathbf{A} = \mathbf{LU}$$

where

$$\mathbf{L} = (\mathbf{M}^{(1)})^{-1}(\mathbf{M}^{(2)})^{-1}$$

is lower triangular.

The foregoing discussion is clearly applicable to any dimension n, i.e., forward elimination on a nonsingular $n \times n$ matrix $\mathbf{A}$ yields a factorization of the form $\mathbf{A} = \mathbf{LU}$, where

$$\mathbf{L} = (\mathbf{M}^{(1)})^{-1} \cdots (\mathbf{M}^{(n-1)})^{-1}$$

2.7.2 The Matrix L

It turns out $\mathbf{L}$ is particularly easy to compute due to the following interesting fact.

Fact. *Let $\mathbf{C}^{(j)}$ be a $n \times n$ matrix whose entries are zero with the possible exception of the subdiagonal segment of the j^{th} column. If $k \geq j$, then*

$$\mathbf{C}^{(j)}\mathbf{C}^{(k)} = \mathbf{0}$$

(Here $\mathbf{0}$ is the all-zeros matrix in $\mathbf{R}^{n \times n}$.)

To prove this fact, note the special form of $\mathbf{C}^{(j)}$ (elements not shown are zero):

$$\mathbf{C}^{(j)} \;=\; \begin{bmatrix}
0 & & & & & & & \\
 & 0 & & & & & & \\
 & & \ddots & & & & & \\
 & & & 0 & & & & \\
 & & & & 0 & & & \\
 & & & & m_{j+1,j} & 0 & & \\
 & & & & \vdots & & \ddots & \\
 & & & & m_{nj} & & & 0
\end{bmatrix}$$

Right multiplication of $\mathbf{C}^{(j)}$ by $\mathbf{C}^{(k)}$ will yield all-zero vectors in all columns of $\mathbf{C}^{(j)}\mathbf{C}^{(k)}$ except (possibly) the k^{th} column. That column is a linear combination of the columns of $\mathbf{C}^{(j)}$ with coefficients given by corresponding entries in the k^{th} column of $\mathbf{C}^{(k)}$. Since the nonzero entries in the k^{th} column of $\mathbf{C}^{(k)}$ appear below the main diagonal, only columns indexed $k+1$ through n of $\mathbf{C}^{(j)}$ are included in the linear combination. But those columns are all zero, since $k+1 > j$ by hypothesis. This completes the proof. □

Now, the lower triangular matrix

$$\mathbf{M}^{(j)} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & m_{j+1,j} & 1 & & \\ & & & & \vdots & & \ddots & \\ & & & & m_{nj} & & & 1 \end{bmatrix}$$

can be expressed as

$$\mathbf{M}^{(j)} = \mathbf{I} + \mathbf{C}^{(j)}$$

Using the previous fact, we have

$$(\mathbf{I} + \mathbf{C}^{(j)})(\mathbf{I} - \mathbf{C}^{(j)}) = \mathbf{I} - \mathbf{C}^{(j)} + \mathbf{C}^{(j)} - \mathbf{C}^{(j)}\mathbf{C}^{(j)} = \mathbf{I}$$

and therefore

$$(\mathbf{M}^{(j)})^{-1} = \mathbf{I} - \mathbf{C}^{(j)} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & -m_{j+1,j} & 1 & & \\ & & & & \vdots & & \ddots & \\ & & & & -m_{nj} & & & 1 \end{bmatrix}$$

Using the same fact, we have that

$$\begin{aligned} (\mathbf{M}^{(1)})^{-1}(\mathbf{M}^{(2)})^{-1} &= (\mathbf{I} - \mathbf{C}^{(1)})(\mathbf{I} - \mathbf{C}^{(2)}) \\ &= \mathbf{I} - \mathbf{C}^{(1)} - \mathbf{C}^{(2)} + \mathbf{C}^{(1)}\mathbf{C}^{(2)} \\ &= \mathbf{I} - \mathbf{C}^{(1)} - \mathbf{C}^{(2)} \end{aligned}$$

and thus the product $(\mathbf{M}^{(1)})^{-1}(\mathbf{M}^{(2)})^{-1}$ is obtained by "overlaying" the two matrices, i.e., adding their subdiagonal parts while keeping the same (unit) diagonal. Right multiplication by $(\mathbf{M}^{(3)})^{-1}, \ldots, (\mathbf{M}^{(n-1)})^{-1}$ in succession yields the final result

$$\mathbf{L} = (\mathbf{M}^{(1)})^{-1}(\mathbf{M}^{(2)})^{-1} \cdots (\mathbf{M}^{(n-1)})^{-1} = \mathbf{I} - \mathbf{C}^{(1)} - \mathbf{C}^{(2)} - \cdots - \mathbf{C}^{(n-1)}$$

or

$$\mathbf{L} = \begin{bmatrix} 1 & & & & \\ -m_{21} & 1 & & & \\ -m_{31} & -m_{32} & & & \\ \vdots & \vdots & \ddots & & \\ -m_{n-1,1} & -m_{n-1,2} & \ldots & 1 & \\ -m_{n1} & -m_{n2} & \ldots & -m_{n,n-1} & 1 \end{bmatrix}$$

(all superdiagonal elements are zero).

In other words, the subdiagonal entries of $\mathbf{L}$ are obtained from the m column of the Gaussian elimination table by a simple sign inversion.

Example 2.7.1. Continuing the example of Subsection 2.7.1, we have

$$\mathbf{M}^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{M}^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 3 & 1 \end{bmatrix}$$

Therefore

$$(\mathbf{M}^{(1)})^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} \qquad \text{and} \qquad (\mathbf{M}^{(2)})^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix}$$

Using the overlay property, we obtain

$$(\mathbf{M}^{(1)})^{-1}(\mathbf{M}^{(2)})^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -4 & -3 & 1 \end{bmatrix}$$

Thus the LU factorization of $\mathbf{A}$ is given by

$$\begin{bmatrix} 2 & 1 & -1 \\ 4 & 0 & -1 \\ -8 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -4 & -3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -1 \\ 0 & -2 & 1 \\ 0 & 0 & 2 \end{bmatrix} \qquad \square$$

Note: The overlay property *cannot* be used to compute the product

$$\mathbf{L}^{-1} = \mathbf{M}^{(n-1)} \cdots \mathbf{M}^{(2)}\mathbf{M}^{(1)}$$

since the order of the nonzero column segments in the $\mathbf{M}^{(j)}$'s is reversed. Thus in Example 2.7.1,

$$\mathbf{L}^{-1} = \mathbf{M}^{(2)}\mathbf{M}^{(1)} \neq \mathbf{M}^{(1)}\mathbf{M}^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix}$$

2.7.3 Applications of the LU Factorization

In many practical situations, the equation

$$\mathbf{Ax} = \mathbf{b}$$

is solved for the same matrix $\mathbf{A}$ (which could represent a known system or a set of standard reference signals), but for *many* different vectors $\mathbf{b}$. The LU factorization of $\mathbf{A}$ allows us to store the results of the forward elimination in a way that minimizes the computational effort involved in solving the equation for a new value of $\mathbf{b}$. Indeed, forward elimination on $\mathbf{A}$ can be performed ahead of time (i.e., off-line). It is interesting to note that the information contained in the pair $(\mathbf{L}, \mathbf{U})$ (ignoring the known zero and unit elements) amounts to n^2 real numbers, which is exactly the same amount as in the original matrix $\mathbf{A}$.

For a given $\mathbf{b}$, the equation

$$\mathbf{Ax} = \mathbf{b}$$

is equivalent to

$$\mathbf{LUx} = \mathbf{b}$$

which has solution

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} = \mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{b}$$

We saw that forward elimination transforms $\mathbf{A}$ to $\mathbf{L}^{-1}\mathbf{A} = \mathbf{U}$; and that backward substitution transforms $\mathbf{U}$ to $\mathbf{U}^{-1}\mathbf{U} = \mathbf{I}$. Focusing on the updates of the right-hand side vector $\mathbf{b}$, we note that forward elimination transforms $\mathbf{b}$ to

$$\mathbf{y} = \mathbf{L}^{-1}\mathbf{b}$$

and backward substitution transforms $\mathbf{y}$ to

$$\mathbf{x} = \mathbf{U}^{-1}\mathbf{y}$$

Thus in effect, forward elimination solves the equation

$$\mathbf{Ly} = \mathbf{b}$$

while backward substitution solves

$$\mathbf{Ux} = \mathbf{y}$$

The two systems above can be solved using the same principle, namely substitution of known values to equations below (in the former case) or above (in the latter case). Thus the terms *substitution* and *elimination* can be used interchangeably in both the forward and the backward phase of Gaussian elimination. The similarity between the two procedures becomes more prominent when the factors **L** and **U** are known ahead of time, in which case there is no need to *derive* **L** by working on **A**.

2.7.4 Row Operations in Backward Substitution

The analogy between the lower triangular system $\mathbf{Ly} = \mathbf{b}$ and the upper triangular system $\mathbf{Ux} = \mathbf{y}$ suggests that row operations in backward substitution may also be implemented by a series of matrix multiplications similar to those used in forward elimination. This is indeed true, and is illustrated in Example 2.7.2.

Example 2.7.2. The forward elimination phase in Subsection 2.6.3 resulted in

m	x_1	x_2	x_3	b
	2	1	-1	6
	0	-2	1	-6
	0	0	2	-4

Dividing each row by its (main) diagonal element, we obtain

m	x_1	x_2	x_3	b
	1	1/2	$-1/2$	3
	0	1	$-1/2$	3
	0	0	1	-2

We now enter multipliers 1/2 and 1/2 for the first and second rows respectively. These are sign-inverted entries from the third column (corresponding to x_3).

m	x_1	x_2	x_3	b
1/2	1	1/2	$-1/2$	3
1/2	0	1	$-1/2$	3
	0	0	1	-2

Scaling the third row by the multiplier shown and adding it to the corresponding row results in eliminating x_3 from both the first and second equation:

m	x_1	x_2	x_3	b
$-1/2$	1	$1/2$	0	2
	0	1	0	2
	0	0	1	-2

Finally, we use the multiplier $-1/2$ (shown above) to scale the second row prior to adding it to the first one, which results in elimination of x_2 from the first equation:

m	x_1	x_2	x_3	b
	1	0	0	1
	0	1	0	2
	0	0	1	-2

We have thus reduced $\mathbf{U}$ to the identity $\mathbf{I}$, and obtained the solution $\mathbf{x}$ in the last column.

In terms of matrix multiplications, we have

$$\mathbf{I} = \mathbf{M}^{(5)}\mathbf{M}^{(4)}\mathbf{M}^{(3)}\mathbf{U}$$

where

$$\mathbf{M}^{(3)} = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & -1/2 & 0 \\ 0 & 0 & 1/2 \end{bmatrix}$$

$$\mathbf{M}^{(4)} = \begin{bmatrix} 1 & 0 & 1/2 \\ 0 & 1 & 1/2 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{M}^{(5)} = \begin{bmatrix} 1 & -1/2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \square$$

We make the following additional observations on Example 2.7.2.

- $\mathbf{M}^{(3)}$ is actually diagonal, with inverse

$$\mathbf{D} = (\mathbf{M}^{(3)})^{-1} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

The result of left-multiplying $\mathbf{U}$ by $\mathbf{M}^{(3)} = \mathbf{D}^{-1}$ is

$$\mathbf{V} = \mathbf{D}^{-1}\mathbf{U} = \begin{bmatrix} 1 & 1/2 & -1/2 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1 \end{bmatrix}$$

which, like $\mathbf{L}$, has diagonal entries *normalized* to unity. We thus have an equivalent *normalized* factorization

$$\mathbf{A} = \mathbf{LDV}$$

- If the diagonal entries are not normalized prior to backward substitution, then the multipliers will be of the form $-s_{ij}/s_{jj}$, as was the case in forward elimination. At the end, $\mathbf{U}$ will be reduced to a diagonal (not necessarily identity) matrix.

2.8 Pivoting in Gaussian Elimination

2.8.1 Pivots and Pivot Rows

The j^{th} step in the forward elimination process aims at eliminating x_j from all equations below the j^{th} one. This is accomplished by subtracting multiples of the j^{th} row of the updated matrix

$$\mathbf{S} = \mathbf{M}^{(j-1)} \cdots \mathbf{M}^{(2)}\mathbf{M}^{(1)} \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$$

from rows indexed $j+1$ through n. Thus for $i \geq j+1$ (only), row $(\mathbf{S})_{i\cdot}$ is replaced by

$$(\mathbf{S})_{i\cdot} + m_{ij}(\mathbf{S})_{j\cdot}$$

This is equivalent to left-multiplying $\mathbf{S}$ by

$$\mathbf{M}^{(j)} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & m_{j+1,j} & 1 & & \\ & & & & \vdots & & \ddots & \\ & & & & m_{nj} & & & 1 \end{bmatrix}$$

The multipliers are given by

$$m_{ij} = -\frac{s_{ij}}{s_{jj}}$$

The denominator s_{jj} in the above expression is known as the *pivotal element*, or simply *pivot*, for x_j. The j^{th} row, multiples of which are subtracted from the remaining $n-j$ rows (below it), is known as the *pivot row* for x_j. Clearly, the pivot row contains s_{jj} in its j^{th} position (column).

2.8.2 Zero Pivots

If a zero pivot s_{jj} is encountered in the forward elimination process, then the j^{th} row cannot be used for eliminating x_j from the rows below it—the multipliers m_{ij} will equal infinity for all i such that $s_{ij} \neq 0$.

Assuming that such $s_{ij} \neq 0$ exists (for $i > j$), it makes sense to interchange the i^{th} and j^{th} rows and use the nonzero value—formerly s_{ij} and now s_{jj}—as pivot. This is a simple concept which will be illustrated later.

An interesting question is what happens when s_{jj}, as well as *all* elements below it, are zero. To answer this question, suppose that s_{jj} is the *first* zero pivot encountered. We will then have

$$\mathbf{M}^{(j-1)} \cdots \mathbf{M}^{(2)} \mathbf{M}^{(1)} \mathbf{A} = \begin{bmatrix} s_{11} & s_{12} & s_{13} & \dots & s_{1,j-1} & s_{1j} & \dots \\ 0 & s_{22} & s_{23} & \dots & s_{2,j-1} & s_{2j} & \dots \\ 0 & 0 & s_{33} & \dots & s_{3,j-1} & s_{3j} & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & s_{j-1,j-1} & s_{j-1,j} & \dots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots \end{bmatrix}$$

where the diagonal entries $s_{11}, \dots, s_{j-1,j-1}$ are previously used pivots and are therefore nonzero. As we argued earlier in Subsection 2.5.4, the j^{th} column can be obtained as a linear combination of columns to its left, and thus the matrix is *singular*. Since

$$\mathbf{M}^{(j-1)} \cdots \mathbf{M}^{(2)} \mathbf{M}^{(1)}$$

is singular, $\mathbf{A}$ must be singular also. (This can be argued by contradiction: if $\mathbf{A}$ were nonsingular, then

$$\mathbf{M}^{(j-1)} \cdots \mathbf{M}^{(2)} \mathbf{M}^{(1)} \mathbf{A}$$

would be the product of two nonsingular matrices, and would thus be nonsingular.)

Singularity of $\mathbf{A}$ will therefore result in a zero pivot during the forward phase of the Gaussian elimination. Recall that singularity implies that

$$\mathbf{Ax} = \mathbf{b}$$

will have a solution only for certain vectors $\mathbf{b}$, namely those in the range of $\mathbf{A}$ (which has dimension smaller than $n-1$). If the solution exists, it will not be unique. The course of action when singularity in $\mathbf{A}$ is detected depends very much on the application in question.

2.8.3 Small Pivots

We saw that row interchanges are necessary when zero pivots are encountered (assuming $\mathbf{A}$ is nonsingular). Row interchanges are also highly desirable when small pivots are likely to cause problems in finite-precision calculations.

Consider the simple 2×2 system given by

$$\begin{aligned} \epsilon x_1 + a_{12}x_2 &= b_1 \\ a_{21}x_1 + a_{22}x_2 &= b_2 \end{aligned}$$

where a_{12}, a_{21}, a_{22}, b_1 and b_2 are of the same order of magnitude, and are all much larger than ϵ. The exact solution is easy to obtain in closed form; if we ignore small contributions due to ϵ, we have the approximations

$$x_1 \simeq \frac{a_{12}b_2 - a_{22}b_1}{a_{12}a_{21}} \qquad \text{and} \qquad x_2 \simeq \frac{b_1}{a_{12}}$$

Neither x_1 nor x_2 is particularly large if the constants involved (except ϵ) are of the same order of magnitude.

Clearly, the same solution is obtained regardless of the pivoting order. But if precision is an issue, then the order shown above can lead to large errors in the variable x_1, which is obtained from the computed value of x_2 (by back-substitution):

$$x_1 = \frac{b_1 - a_{12}x_2}{\epsilon}$$

We know that the value of x_1 is not particularly large; therefore the numerator cannot be that much larger than the denominator. Yet both b_1 and $a_{12}x_2$ are (absolutely) much larger than ϵ. This means that the difference $b_1 - a_{12}x_2$ is very small compared to either term, and is likely to be poorly approximated if the precision of the computation is not high enough. The following numerical example further illustrates these difficulties.

Example 2.8.1. Consider the system

$$\begin{aligned} 0.003x_1 + 6x_2 &= 4.001 \\ x_1 + x_2 &= 1 \end{aligned}$$

which has exact solution $x_1 = 1/3$, $x_2 = 2/3$. Let us solve the system using four-digit precision throughout.

Treating the equations in their original order, we obtain a multiplier

$$m_{21} = -(1.000)/(0.003) = -333.3$$

and thus the second equation becomes

$$(1 - (333.3)(6))x_2 = 1 - (333.3)(4.001)$$

The resulting value of x_2 is

$$x_2 = 0.6668$$

Although the relative error in x_2 is only 0.02%, the effect of this error is much more severe on x_1: using that value for x_2 in the first equation, we have

$$x_1 = \frac{4.001 - (6)(0.6668)}{0.003}$$

Rounding the product (6)(0.6668) to four digits results in 4.001, and thus the answer is $x_1 = 0.000$. This is clearly unsatisfactory.

Interchanging the equations and repeating the calculation, we obtain $x_2 = 0.6667$ and $x_1 = 0.3333$, i.e., the correct values (rounded to four significant digits). □

2.8.4 Row Pivoting

Row pivoting is a simple strategy for avoiding small pivots, which can be summarized as follows.

1. At the beginning of the j^{th} step, let I_j be the set of indices i corresponding to rows that have not yet been used as pivot rows.

2. Compare the values of $|s_{ij}|$ for all i in I_j. If $i = i(j)$ yields the largest such value, take $i(j)$ as the pivot row for the j^{th} step.

3. Remove index $i(j)$ from the set I_j.

4. For each i in I_j, compute $m_{ij} = -s_{ij}/s_{i(j),j}$.

5. For each i in I_j, add the appropriate multiple of row $i(j)$ to row i.

6. Increment j by 1.

In implementing row pivoting, we use an additional column ("p" for "pivot order") to indicate the first, second, third, etc., pivot row.

Example 2.8.2. Consider the system

$$\begin{aligned} -x_2 + 3x_3 &= 6 \\ 3x_1 + 2x_2 + x_3 &= 15 \\ 2x_1 + 4x_2 - 5x_3 &= 1 \end{aligned}$$

Clearly, pivoting is required for the first step since $a_{11} = 0$. We will follow the row pivoting algorithm given above. We initialize the table:

p	m	x_1	x_2	x_3	b
		0	-1	3	6
		3	2	1	15
		2	4	-5	1

To eliminate x_1, we look for the (absolutely) largest value in the first (x_1) column, which equals 3. The corresponding row is chosen as the first pivot row, shown as "1" in the pivot column. Multipliers are also computed for the remaining rows:

p	m	x_1	x_2	x_3	b
	0	0	-1	3	6
1		$\underline{3}$	2	1	15
	$-2/3$	2	4	-5	1

After eliminating x_1 from rows 1 (trivially) and 2, we compare the second (x_2) column entries for those rows. The pivot is $8/3$, and row 3 is the pivot row. A "2" is marked for that row in the pivot column, and the multiplier is computed for row 1:

p	m	x_1	x_2	x_3	b
	$3/8$	0	-1	3	6
1		3	2	1	15
2		0	$\underline{8/3}$	$-17/3$	-9

After eliminating x_2 from row 1, we mark a "3" for that row in the pivot column (even though it is not used for pivoting subsequently):

p	m	x_1	x_2	x_3	b
3		0	0	$7/8$	$21/8$
1		3	2	1	15
2		0	$8/3$	$-17/3$	-9

Backward substitution can be carried out by handling the equations in pivoting order (i.e., $1, 2, 3$, from the p column):

$$\begin{aligned} 3x_1 + 2x_2 + x_3 &= 15 \\ (8/3)x_2 - (17/3)x_3 &= -9 \\ (7/8)x_3 &= 21/8 \end{aligned}$$

Solving in the usual fashion, we obtain $x_3 = 3$, $x_2 = 3$ and $x_1 = 2$. □

Row pivoting results in a so-called *permuted LU* factorization of the matrix $\mathbf{A}$:

$$\mathbf{LU} = \mathbf{PA}$$

where $\mathbf{L}$ and $\mathbf{U}$ are lower and upper triangular, respectively, and $\mathbf{P}$ is a permutation matrix which rearranges the rows of A in the order in which they were used for pivoting. Thus in Example 2.8.2,

$$\mathbf{P}\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

where the vector on the right is just the pivot order column at the end of the forward elimination. Thus

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \qquad \text{and} \qquad \mathbf{PA} = \begin{bmatrix} 2 & 4 & -5 \\ 3 & 2 & 1 \\ 0 & -1 & 3 \end{bmatrix}$$

The matrices $\mathbf{L}$ and $\mathbf{U}$ are obtained by applying the same permutation to the multiplier vectors and $\mathbf{S}$ (at the end of forward elimination). Thus in the same example,

$$\mathbf{L} = \mathbf{I} - \mathbf{P}\begin{bmatrix} 0 & 3/8 & 0 \\ 0 & 0 & 0 \\ -2/3 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2/3 & 1 & 0 \\ 0 & -3/8 & 1 \end{bmatrix}$$

and

$$\mathbf{U} = \mathbf{P}\begin{bmatrix} 0 & 0 & 7/8 \\ 3 & 2 & 1 \\ 0 & 8/3 & -17/3 \end{bmatrix} = \begin{bmatrix} 3 & 2 & 1 \\ 0 & 8/3 & -17/3 \\ 0 & 0 & 7/8 \end{bmatrix}$$

If the permuted LU factorization of $\mathbf{A}$ is known, then the equation $\mathbf{Ax} = \mathbf{b}$ can be solved using the forward elimination and backward substitution equations

$$\mathbf{Ly} = \mathbf{Pb} \qquad \text{and} \qquad \mathbf{Ux} = \mathbf{y}\ .$$

2.9 Further Topics in Gaussian Elimination

2.9.1 Computation of the Matrix Inverse

In most computations, the inverse $\mathbf{A}^{-1}$ of a nonsingular $n \times n$ matrix $\mathbf{A}$ appears in a product such as $\mathbf{A}^{-1}\mathbf{b}$ or, more generally, $\mathbf{A}^{-1}\mathbf{B}$. As long as $\mathbf{A}$ is available, it is best to compute that product via Gaussian elimination, i.e., by solving

$$\mathbf{Ax} = \mathbf{b}$$

or, if $\mathbf{B}$ has more than one column,

$$\mathbf{AX} = \mathbf{B}$$

(here $\mathbf{X}$ has the same number of columns as $\mathbf{B}$). This approach is faster (at least by a factor of two), and produces a much smaller average error in the resulting entries of $\mathbf{Ax}-\mathbf{b}$, than computing the inverse $\mathbf{A}^{-1}$ followed by the product $\mathbf{A}^{-1}\mathbf{b}$.

If $\mathbf{Ax} = \mathbf{b}$ needs to be solved for many different vectors $\mathbf{b}$, having $\mathbf{A}^{-1}$ computed and stored off-line offers no particular advantage over the standard LU factorization of $\mathbf{A}$. This is because the product $\mathbf{A}^{-1}\mathbf{b}$ requires the same number ($\approx 2n^2$) of floating-point operations as the two triangular systems $\mathbf{Ly} = \mathbf{b}$ and $\mathbf{Ux} = \mathbf{y}$ combined.

If we are asked to compute $\mathbf{A}^{-1}$ explicitly, we can do so by solving the n^2 simultaneous equations in

$$\mathbf{AX} = \mathbf{I}$$

where $\mathbf{X} = [\mathbf{x}^{(1)} \ \ldots \ \mathbf{x}^{(n)}]$. This is equivalent to solving n systems of n simultaneous equations each:

$$\mathbf{Ax}^{(1)} = \mathbf{e}^{(1)} \ , \qquad \mathbf{Ax}^{(2)} = \mathbf{e}^{(2)} \ , \qquad \ldots \ , \qquad \mathbf{Ax}^{(n)} = \mathbf{e}^{(n)}$$

We illustrate this procedure, also known as *Gauss-Jordan elimination*, by an example.

Example 2.9.1. To invert

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 2 \\ -2 & 1 & 1 \\ -1 & 2 & 1 \end{bmatrix}$$

we construct a table with the elements of $\mathbf{S} = [\mathbf{A} \quad \mathbf{I}]$:

m	x_1	x_2	x_3	b_1	b_2	b_3
	$\underline{1}$	-1	2	1	0	0
2	-2	1	1	0	1	0
1	-1	2	1	0	0	1

Eliminating x_1 from rows 2 and 3 (using the pivot and multipliers shown above), we obtain

m	x_1	x_2	x_3	b_1	b_2	b_3
	1	-1	2	1	0	0
	0	$-\underline{1}$	5	2	1	0
1	0	1	3	1	0	1

Eliminating x_2 from row 3 (using the pivot and multiplier shown above), we obtain

m	x_1	x_2	x_3	b_1	b_2	b_3
	1	-1	2	1	0	0
	0	-1	5	2	1	0
	0	0	8	3	1	1

i.e., $\mathbf{S} = [\mathbf{U} \quad \mathbf{L}^{-1}]$ at this point. Scaling the rows of $\mathbf{U}$ by the diagonal elements, we obtain

m	x_1	x_2	x_3	b_1	b_2	b_3
-2	1	-1	2	1	0	0
5	0	1	-5	-2	-1	0
	0	0	$\underline{1}$	3/8	1/8	1/8

Substituting x_3 back into rows 2 and 1 (using the multipliers shown above), we obtain

m	x_1	x_2	x_3	b_1	b_2	b_3
1	1	-1	0	2/8	$-2/8$	$-2/8$
	0	$\underline{1}$	0	$-1/8$	$-3/8$	5/8
	0	0	1	3/8	1/8	1/8

Substituting x_2 back into row 1 (using the multiplier shown above), we obtain

m	x_1	x_2	x_3	b_1	b_2	b_3
	1	0	0	1/8	$-5/8$	3/8
	0	1	0	$-1/8$	$-3/8$	5/8
	0	0	1	3/8	1/8	1/8

i.e., $\mathbf{S} = [\mathbf{I} \quad \mathbf{A}^{-1}]$ at the end. Thus

$$\mathbf{A}^{-1} = \begin{bmatrix} 1/8 & -5/8 & 3/8 \\ -1/8 & -3/8 & 5/8 \\ 3/8 & 1/8 & 1/8 \end{bmatrix} \qquad \square$$

In case where zero pivots are encountered, pivoting is necessary in the above procedure. The final table is of the form

$$[\mathbf{P}^{-1} \quad \mathbf{P}^{-1}\mathbf{A}^{-1}]$$

where $\mathbf{P}$ is the permutation matrix in the (permuted) LU factorization of $\mathbf{A}$:

$$\mathbf{LU} = \mathbf{PA}$$

Note that $\mathbf{P}^{-1} = \mathbf{P}^T$ by the fact given in Subsection 2.3.2.

2.9.2 Ill-Conditioned Problems

In Section 2.8, we saw how small pivots can cause large errors in finite-precision computation, and how simple techniques such as row pivoting can essentially eliminate these errors. There are, nevertheless, problems where errors due to finite precision are unavoidable, whether pivoting is used or not. Such problems are referred to as *ill-conditioned*, primarily in regard to the matrix $\mathbf{A}$.

We can think of an ill-conditioned matrix $\mathbf{A}$ as one that is "nearly singular" in terms of the precision used in solving $\mathbf{Ax} = \mathbf{b}$. This concept can quantified in terms of the so-called *condition number* of $\mathbf{A}$, which can be computed in MATLAB using the function `COND`. A well-conditioned matrix is one whose condition number is small (the minimum possible value is 1). An ill-conditioned matrix is one whose condition number is of the same order of magnitude as 10^K, where K is the number of significant decimal digits used in the computation—the same definition can be given in terms of binary digits by changing the base to 2.

Ill-conditioned problems are characterized by large variations in the solution vector $\mathbf{x}$ caused by small perturbations of the parameter values (entries of $\mathbf{A}$ and $\mathbf{b}$). They are thus *extremely* sensitive to rounding. The following example demonstrates this sensitivity.

Example 2.9.2. Consider the system

$$\begin{aligned} 1.001x_1 + x_2 &= 2.001 \\ x_1 + 1.001x_2 &= 2.001 \end{aligned}$$

which has exact solution

$$x_1 = x_2 = 1.$$

Note how the two columns of $\mathbf{A}$ are nearly equal, and thus $\mathbf{A}$ is nearly singular. The condition number of $\mathbf{A}$ equals 2.001×10^3. Based on the foregoing discussion, large errors are possible when rounding to *three* significant digits. In fact, we will show that this is the case even with four-digit precision.

Suppose we change the right-hand vector to

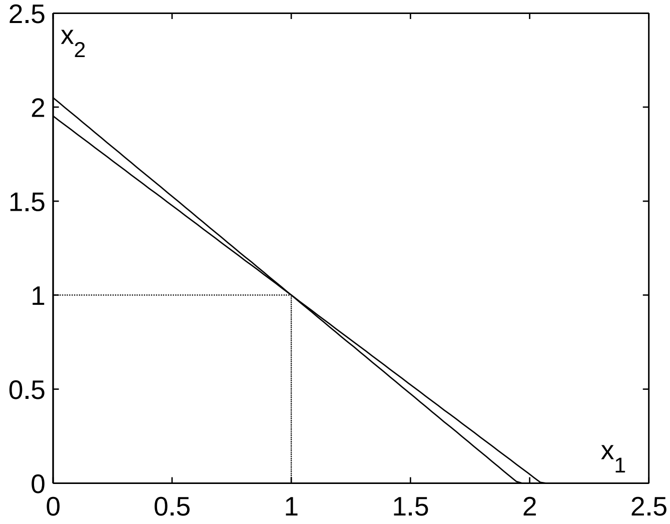
$$b_1 = 2.0006 \ , \qquad b_2 = 2.0014$$

Both these values are rounded to 2.001 using four-digit precision, with an error of 0.02%. Ignoring any further round-off errors in the solution of the system (i.e., solving it exactly), we have the exact solution

$$x_1 = 1.600 \ , \qquad x_2 = 0.3995$$

Note the 60% error in each entry relative to the earlier solution.

We give a graphical illustration of the difficulties encountered in this problem. The two equations represent two nearly parallel straight lines on the (x_1, x_2) plane. A small parallel displacement of either line (as a result of changing the value of an intercept b_i) will cause a large movement of the solution point. □



Example 2.9.2

Ill-conditioned problems force us to reconsider the physical parameters and models involved. For example, if the signal representation problem $\mathbf{s} = \mathbf{V}\mathbf{c}$ is ill-conditioned, we may want to choose a different set of reference signals in the matrix $\mathbf{V}$. In a systems context, where, for example, $\mathbf{y} = \mathbf{A}\mathbf{x}$ is an observation signal for a hidden state vector $\mathbf{x}$, an ill-conditioned matrix $\mathbf{A}$ might force us to seek a different measuring device or observation method.

2.10 Inner Products, Projections and Signal Approximation

2.10.1 Inner Products and Norms

The *inner product* of two vectors $\mathbf{a}$ and $\mathbf{b}$ in $\mathbf{R}^{m\times 1}$ is defined by

$$\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{m} a_i b_i$$

Clearly, the inner product is *symmetric* in its two arguments and can be expressed as the product of a row vector and a column vector:

$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T\mathbf{b} = \mathbf{b}^T\mathbf{a} = \langle \mathbf{b}, \mathbf{a} \rangle$$

Also, the inner product is *linear* in one of its arguments (i.e., vectors) when the other argument is kept constant:

$$\langle c_1\mathbf{a}^{(1)} + c_2\mathbf{a}^{(2)},\, \mathbf{b} \rangle = \sum_{i=1}^{m} \left(c_1 a_i^{(1)} + c_2 a_i^{(2)} \right) b_i = c_1\langle \mathbf{a}^{(1)}, \mathbf{b} \rangle + c_2\langle \mathbf{a}^{(2)}, \mathbf{b} \rangle$$

The *norm* of a vector $\mathbf{a}$ in $\mathbf{R}^{m\times 1}$ is defined as the square root of the inner product of $\mathbf{a}$ with itself:

$$\|\mathbf{a}\| = \langle \mathbf{a}, \mathbf{a} \rangle^{1/2} = \left(\sum_{i=1}^{m} a_i^2 \right)^{1/2}$$

or equivalently,

$$\|\mathbf{a}\|^2 = \langle \mathbf{a}, \mathbf{a} \rangle = \sum_{i=1}^{m} a_i^2$$

By the Pythagorean theorem, $\|\mathbf{a}\|$ is also the *length* of $\mathbf{a}$. Clearly,

$$\|\mathbf{a}\| = 0 \quad \Leftrightarrow \quad \mathbf{a} = \mathbf{0}$$

i.e., the all-zeros vector is the only vector of zero length. Also, if c is a (real) scaling factor,

$$\|c\mathbf{a}\| = \left(\sum_{i=1}^{m} c^2 a_i^2 \right)^{1/2} = |c| \cdot \|\mathbf{a}\|$$

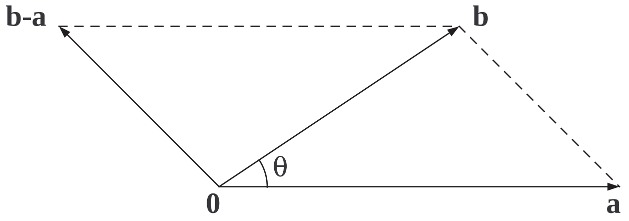


Figure 2.8: The geometry of vectors **a**, **b** and **b** − **a**.

2.10.2 Angles, Projections and Orthogonality

In general, two nonzero vectors **a** and **b** in $\mathbf{R}^{m\times 1}$ define a two-dimensional subspace (i.e., plane) through the origin. By convention, the angle between **a** and **b** takes value in $[0, \pi]$.

Applying the cosine rule to the triangle formed by **a**, **b** and the dotted vector (parallel to) $\mathbf{b} - \mathbf{a}$ in Figure 2.8, we obtain

$$\|\mathbf{b} - \mathbf{a}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\|\mathbf{a}\| \cdot \|\mathbf{b}\| \cdot \cos\theta$$

Using the linearity property stated earlier, the left-hand side can be also expressed as

$$\begin{aligned}\|\mathbf{b} - \mathbf{a}\|^2 &= \langle \mathbf{b} - \mathbf{a},\, \mathbf{b} - \mathbf{a}\rangle \\ &= \langle \mathbf{a}, \mathbf{a}\rangle + \langle \mathbf{b}, \mathbf{b}\rangle - 2\langle \mathbf{a}, \mathbf{b}\rangle \\ &= \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\langle \mathbf{a}, \mathbf{b}\rangle\end{aligned}$$

Thus

$$\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\langle \mathbf{a}, \mathbf{b}\rangle = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\|\mathbf{a}\| \cdot \|\mathbf{b}\| \cdot \cos\theta$$

and consequently

$$\cos\theta = \frac{\langle \mathbf{a}, \mathbf{b}\rangle}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|}$$

Another feature of the geometry of two vectors in $\mathbf{R}^{m\times 1}$ is the *projection* of one vector onto the other. Figure 2.9 shows the projection **f** of **b** onto **a**, which is of the form

$$\mathbf{f} = \lambda\mathbf{a}$$

where λ is a scaling factor having the same sign as $\cos\theta$. To determine the value of λ, note that the length of **f** equals

$$\|\mathbf{f}\| = \|\mathbf{b}\| \cdot |\cos\theta| = \frac{|\langle \mathbf{a}, \mathbf{b}\rangle|}{\|\mathbf{a}\|}$$

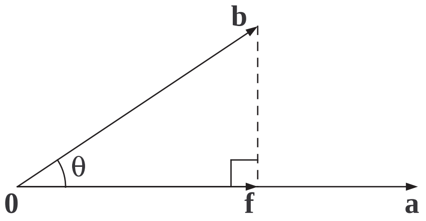


Figure 2.9: Projection of vector $\mathbf{b}$ onto vector $\mathbf{a}$.

The unit vector parallel to $\mathbf{a}$ is given by

$$\mathbf{e}^{(\mathbf{a})} = \frac{1}{\|\mathbf{a}\|}\mathbf{a}$$

and $\mathbf{f}$ is obtained by scaling $\mathbf{e}^{(\mathbf{a})}$ by the "signed" length of $\mathbf{f}$:

$$\mathbf{f} = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\|}\mathbf{e}^{(\mathbf{a})} = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\|^2}\mathbf{a}$$

Therefore $\lambda = \langle \mathbf{a}, \mathbf{b} \rangle / \|\mathbf{a}\|^2$.

Example 2.10.1. Let $\mathbf{a} = [-1\ 1\ \ -1]^T$ and $\mathbf{b} = [2\ \ -5\ 1]^T$ in $\mathbf{R}^{3\times 1}$. We then have

$$\|\mathbf{a}\|^2 = 3 \quad \Rightarrow \quad \|\mathbf{a}\| = \sqrt{3}$$

and

$$\|\mathbf{b}\|^2 = 30 \quad \Rightarrow \quad \|\mathbf{b}\| = \sqrt{30}$$

Also,

$$\langle \mathbf{a}, \mathbf{b} \rangle = -2 - 5 - 1 = -8 \quad \Rightarrow \quad \cos\theta = -\frac{8}{3\sqrt{10}}$$

The projection of $\mathbf{b}$ onto $\mathbf{a}$ is given by

$$\mathbf{f} = -\frac{8}{3}\mathbf{a}$$

while the projection of $\mathbf{a}$ onto $\mathbf{b}$ is given by

$$\mathbf{g} = -\frac{8}{30}\mathbf{b} = -\frac{4}{15}\mathbf{b} \qquad \square$$

Definition 2.10.1. We say that $\mathbf{a}$ and $\mathbf{b}$ are *orthogonal*, and denote that relationship by

$$\mathbf{a}\perp\mathbf{b}$$

if the angle between $\mathbf{a}$ and $\mathbf{b}$ equals $\pi/2$, or equivalently (since $\cos(\pi/2) = 0$),

$$\langle\mathbf{a},\mathbf{b}\rangle = 0 \qquad \Box$$

Clearly, if $\mathbf{a}$ and $\mathbf{b}$ are orthogonal, then the projection of either vector onto the other is the all-zeros vector $\mathbf{0}$.

2.10.3 Formulation of the Signal Approximation Problem

Our discussion of matrix inversion was partly motivated by the following signal approximation problem. Given n reference signals $\mathbf{v}^{(1)},\dots,\mathbf{v}^{(n)}$ in $\mathbf{R}^{m\times 1}$, how can we best approximate an arbitrary signal vector $\mathbf{s}$ in $\mathbf{R}^{m\times 1}$ by a linear combination of these signals? The approximation would be of the form

$$\hat{\mathbf{s}} = \sum_{r=1}^{n} c_r\mathbf{v}^{(r)} = \mathbf{V}\mathbf{c}$$

where the $m \times n$ matrix $\mathbf{V}$ has the n reference signals as its columns:

$$\mathbf{V} = \left[\begin{array}{ccc} \mathbf{v}^{(1)} & \dots & \mathbf{v}^{(n)} \end{array}\right]$$

Since the approximation $\hat{\mathbf{s}}$ is a linear combination of reference signals, it makes little sense to include in $\mathbf{V}$ a signal which is expressible as a linear combination of other such (reference) signals; that signal would be redundant. We therefore assume the following:

Assumption. The columns of $\mathbf{V}$, namely $\mathbf{v}^{(1)},\dots,\mathbf{v}^{(n)}$, will be assumed linearly independent for the remainder of this chapter.

The above assumption immediately implies that $n \le m$, i.e., the number of reference signals is no larger than the signal dimension (i.e., vector size). As we saw earlier, *any* set of m linearly independent vectors in $\mathbf{R}^{m\times 1}$ can be used to obtain *any* vector in that space (by means of linear combinations). It is therefore impossible to have a set of $m+1$ (or more) linearly independent vectors in $\mathbf{R}^{m\times 1}$.

The case $n = m$ has already been considered in the context of matrix inversion. If the columns the (square) matrix $\mathbf{V}$ are linearly independent, then $\mathbf{V}$ is nonsingular and the equation

$$\mathbf{V}\mathbf{c} = \mathbf{s}$$

has a unique solution $\mathbf{c}$ given by $\mathbf{c} = \mathbf{V}^{-1}\mathbf{s}$. Thus there is no need to approximate here; we have an exact representation of $\mathbf{s}$ as a linear combination of the $n = m$ reference signals.

Thus the only real case of interest is $n < m$, i.e., where the range $\mathcal{R}(\mathbf{V})$ of $\mathbf{V}$ is a linear subspace of $\mathbf{R}^{m\times 1}$ of dimension $d = n < m$. Any signal $\mathbf{s}$ that does not belong to $\mathcal{R}(\mathbf{V})$ will need to approximated by a $\hat{\mathbf{s}}$ in $\mathcal{R}(\mathbf{V})$.

To properly formulate this problem, we will evaluate each approximation $\hat{\mathbf{s}}$ based on its distance from $\mathbf{s}$. Thus the *best* approximation is one that *minimizes the length* $\|\hat{\mathbf{s}}-\mathbf{s}\|$ of the error vector $\hat{\mathbf{s}}-\mathbf{s}$ over all possible choices of $\hat{\mathbf{s}}$ in $\mathcal{R}(\mathbf{V})$. Clearly, that approximation will also minimize the square of that length:

$$\|\hat{\mathbf{s}} - \mathbf{s}\|^2 = \sum_{i=1}^{m} (\hat{s}_i - s_i)^2$$

By virtue of the expression on the right-hand side (i.e., sum of squares of entries of the error vector), this solution is known as the *least-squares* (linear) approximation of $\mathbf{s}$ based on $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}$.

2.10.4 Projection and the Signal Approximation Problem

In the case where there is only one reference vector $\mathbf{v}^{(1)}$ (i.e., $m > n = 1$), the solution of the signal approximation problem can be obtained using two-dimensional geometry. If, in Figure 2.9, we replace $\mathbf{b}$ by $\mathbf{s}$ and $\mathbf{a}$ by $\mathbf{v}^{(1)}$, we see that the point closest to $\mathbf{s}$ on the line generated by $\mathbf{v}^{(1)}$ is none other than the *projection* $\mathbf{f}$ of $\mathbf{s}$ onto $\mathbf{v}^{(1)}$. Thus the least-squares approximation of $\mathbf{s}$ based on $\mathbf{v}^{(1)}$ is given by $\hat{\mathbf{s}} = \mathbf{f}$. It is also interesting to note that the error vector $\hat{\mathbf{s}} - \mathbf{s}$ is orthogonal to $\mathbf{v}^{(1)}$, i.e.,

$$\hat{\mathbf{s}} - \mathbf{s} \perp \mathbf{v}^{(1)}$$

Similarly, three-dimensional geometry provides the solution to the approximation problem when two reference vectors $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$ are given (i.e., $m > n = 2$). Again, the solution $\hat{\mathbf{s}}$ is obtained by projecting $\mathbf{s}$ on the plane $\mathcal{R}(\mathbf{V})$. The error vector satisfies

$$\hat{\mathbf{s}} - \mathbf{s} \perp \mathbf{v}^{(1)} \qquad \text{and} \qquad \hat{\mathbf{s}} - \mathbf{s} \perp \mathbf{v}^{(2)}$$

i.e., it is orthogonal to both $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$. Figure 2.10 clearly shows that $\hat{\mathbf{s}} - \mathbf{s}$ is also orthogonal to *every* vector on the plane $\mathcal{R}(\mathbf{V})$.

Our intuition thus far suggests that the solution to the signal approximation problem for *any* number $n < m$ of reference vectors might be obtained

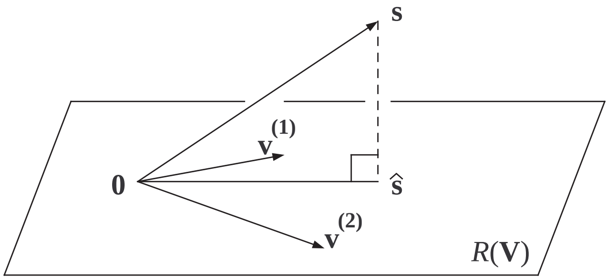


Figure 2.10: The projection of $\mathbf{s}$ on the plane generated by $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$.

by projecting $\mathbf{s}$ onto the subspace $\mathcal{R}(\mathbf{V})$ generated by $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}$; where the projection $\hat{\mathbf{s}} \in \mathcal{R}(\mathbf{V})$ satisfies

$$\hat{\mathbf{s}} - \mathbf{s} \perp \mathbf{v}^{(j)}$$

for every $1 \leq j \leq n$.

Before showing that this is indeed the case, we note the following:

- If $\hat{\mathbf{s}} - \mathbf{s} \perp \mathbf{v}^{(j)}$ for every vector $\mathbf{v}^{(j)}$, then $\hat{\mathbf{s}} - \mathbf{s} \perp \mathbf{a}$ for every $\mathbf{a}$ in $\mathcal{R}(\mathbf{V})$. This is because the inner product is linear in one of its arguments when the other argument is fixed; thus if $\mathbf{a}$ is a linear combination of $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}$ and $\langle \hat{\mathbf{s}} - \mathbf{s}, \mathbf{v}^{(j)} \rangle = 0$ for every j, then $\langle \hat{\mathbf{s}} - \mathbf{s}, \mathbf{a} \rangle = 0$ also.

- The n conditions $\langle \hat{\mathbf{s}} - \mathbf{s}, \mathbf{v}^{(i)} \rangle = (\mathbf{v}^{(i)})^T (\hat{\mathbf{s}} - \mathbf{s}) = 0$ can be expressed in terms of a single matrix-vector product as

$$\mathbf{V}^T (\hat{\mathbf{s}} - \mathbf{s}) = \mathbf{0}$$

where the vector $\mathbf{0}$ is n-dimensional; or equivalently, as

$$\mathbf{V}^T \hat{\mathbf{s}} = \mathbf{V}^T \mathbf{s}$$

We prove our conjecture by showing that *if* there exists $\hat{\mathbf{s}}$ in $\mathcal{R}(\mathbf{V})$ satisfying $\mathbf{V}^T \hat{\mathbf{s}} = \mathbf{V}^T \mathbf{s}$, its distance from $\mathbf{s}$ can be no larger than that of any other vector $\mathbf{y}$ in $\mathcal{R}(\mathbf{V})$; i.e.,

$$\|\mathbf{y} - \mathbf{s}\|^2 \geq \|\hat{\mathbf{s}} - \mathbf{s}\|^2$$

Indeed, we rewrite the left-hand side as

$$\|\mathbf{y}-\hat{\mathbf{s}}+\hat{\mathbf{s}}-\mathbf{s}\|^2 = \langle \mathbf{y}-\hat{\mathbf{s}}+\hat{\mathbf{s}}-\mathbf{s},\, \mathbf{y}-\hat{\mathbf{s}}+\hat{\mathbf{s}}-\mathbf{s}\rangle$$

Treating $\mathbf{y}-\mathbf{s}$ and $\hat{\mathbf{s}}-\mathbf{s}$ as single vectors, we obtain (as in the derivation of the formula for $\cos\theta$ in Subsection 2.10.2)

$$\|\mathbf{y}-\mathbf{s}\|^2 = \|\hat{\mathbf{s}}-\mathbf{s}\|^2 + \|\mathbf{y}-\hat{\mathbf{s}}\|^2 + 2\langle \mathbf{y}-\hat{\mathbf{s}},\, \hat{\mathbf{s}}-\mathbf{s}\rangle$$

The quantity $\|\mathbf{y}-\hat{\mathbf{s}}\|^2$ is nonnegative (and is zero if and only if $\mathbf{y}=\hat{\mathbf{s}}$). The inner product $\langle \mathbf{y}-\hat{\mathbf{s}},\, \hat{\mathbf{s}}-\mathbf{s}\rangle$ equals zero since $\mathbf{y}-\hat{\mathbf{s}}$ belongs to $\mathcal{R}(\mathbf{V})$ and $\hat{\mathbf{s}}-\mathbf{s}$ is orthogonal to every vector in $\mathcal{R}(\mathbf{V})$. We thefore conclude that

$$\|\mathbf{y}-\mathbf{s}\|^2 \geq \|\hat{\mathbf{s}}-\mathbf{s}\|^2$$

with equality if and only if $\mathbf{y}=\hat{\mathbf{s}}$. Although the existence of $\mathbf{s}$ in $\mathcal{R}(\mathbf{V})$ satisfying

$$\mathbf{V}^T\hat{\mathbf{s}} = \mathbf{V}^T\mathbf{s}$$

has not yet been shown, it follows from the “if and only if” statement (above) that there can only be one solution $\hat{\mathbf{s}}$ to the signal approximation problem.

2.10.5 Solution of the Signal Approximation Problem

It remains to show that

$$\mathbf{V}^T\hat{\mathbf{s}} = \mathbf{V}^T\mathbf{s}$$

has a solution $\hat{\mathbf{s}}$ in $\mathcal{R}(\mathbf{V})$; this means that $\hat{\mathbf{s}} = \mathbf{V}\mathbf{c}$ for some $\mathbf{c}$ to be determined. We rewrite the equation as

$$\mathbf{V}^T\mathbf{V}\mathbf{c} = \mathbf{V}^T\mathbf{s}$$

i.e.,

$$\mathbf{A}\mathbf{c} = \mathbf{b}$$

where:

- $\mathbf{A} = \mathbf{V}^T\mathbf{V}$ is an $n \times n$ *symmetric* matrix whose $(i,j)^{\text{th}}$ entry equals the inner product $\langle \mathbf{v}^{(i)}, \mathbf{v}^{(j)}\rangle$ (hence the symmetry); and

- $\mathbf{b} = \mathbf{V}^T\mathbf{s}$ is a $n \times 1$ vector whose i^{th} entry equals $\langle \mathbf{v}^{(i)}, \mathbf{s}\rangle$.

Note that once the inner products in $\mathbf{V}^T\mathbf{V}$ *and* $\mathbf{V}^T\mathbf{s}$ *have been computed, the signal dimension* m *altogether drops out of the picture, i.e., the size of the problem is solely determined by the number* n *of reference vectors.*

A unique solution $\mathbf{c}$ exists for every $\mathbf{s}$ provided the $n \times n$ matrix $\mathbf{V}^T\mathbf{V}$ is nonsingular, i.e.,

$$\mathbf{x} \neq \mathbf{0} \quad \Rightarrow \quad (\mathbf{V}^T\mathbf{V})\mathbf{x} \neq \mathbf{0}$$

To see why the above implication is indeed true, recall our earlier assumption that the columns of the $m \times n$ matrix $\mathbf{V}$ are linearly independent. Thus

$$\begin{aligned} \mathbf{x} \neq \mathbf{0} \quad & \Rightarrow \quad \mathbf{V}\mathbf{x} \neq \mathbf{0} \\ & \Leftrightarrow \quad \|\mathbf{V}\mathbf{x}\|^2 > 0 \end{aligned}$$

where the squared norm in the last expression can be also expressed as

$$\|\mathbf{V}\mathbf{x}\|^2 = (\mathbf{V}\mathbf{x})^T(\mathbf{V}\mathbf{x}) = \mathbf{x}^T\mathbf{V}^T\mathbf{V}\mathbf{x}$$

Thus

$$\mathbf{x} \neq \mathbf{0} \quad \Rightarrow \quad \mathbf{x}^T(\mathbf{V}^T\mathbf{V})\mathbf{x} > 0$$

and hence $(\mathbf{V}^T\mathbf{V})\mathbf{x}$ cannot be the all-zeros vector $\mathbf{0}$; if it were, then the scalar $\mathbf{x}^T(\mathbf{V}^T\mathbf{V})\mathbf{x}$ would equal zero also. This establishes that $\mathbf{V}^T\mathbf{V}$ is nonsingular.

We therefore obtain the solution to the least squares approximation problem as

$$\hat{\mathbf{s}} = \mathbf{V}\mathbf{c}$$

where

$$\mathbf{c} = (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{s}$$

A single expression for $\hat{\mathbf{s}}$ is

$$\hat{\mathbf{s}} = \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{s}$$

(Note that the $m \times m$ matrix $\mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T$ is also symmetric.)

Example 2.10.2. Consider the case $m = 3$ and $n = 2$, with $\mathbf{v}^{(1)} = [-1\ 1\ -1]^T$ and $\mathbf{v}^{(2)} = [2\ \ -5\ 1]^T$ (same vectors as in Example 2.10.1). Thus

$$\mathbf{V} = \begin{bmatrix} -1 & 2 \\ 1 & -5 \\ -1 & 1 \end{bmatrix}$$

and

$$\mathbf{V}^T\mathbf{V} = \begin{bmatrix} \|\mathbf{v}^{(1)}\|^2 & \langle \mathbf{v}^{(1)}, \mathbf{v}^{(2)} \rangle \\ \langle \mathbf{v}^{(2)}, \mathbf{v}^{(1)} \rangle & \|\mathbf{v}^{(2)}\|^2 \end{bmatrix} = \begin{bmatrix} 3 & -8 \\ -8 & 30 \end{bmatrix}$$

Let us determine the projection of $\hat{\mathbf{s}}$ of $\mathbf{s} = [2 \ \ -1 \ \ -1]^T$ onto the plane defined by $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$. We have

$$\mathbf{V}^T\mathbf{s} = \begin{bmatrix} \langle \mathbf{v}^{(1)}, \mathbf{s} \rangle \\ \langle \mathbf{v}^{(2)}, \mathbf{s} \rangle \end{bmatrix} = \begin{bmatrix} -2 \\ 8 \end{bmatrix}$$

and thus $\hat{\mathbf{s}} = \mathbf{V}\mathbf{c}$, where

$$\mathbf{c} = \begin{bmatrix} 3 & -8 \\ -8 & 30 \end{bmatrix}^{-1} \begin{bmatrix} -2 \\ 8 \end{bmatrix} = \frac{1}{13} \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

Therefore

$$\hat{\mathbf{s}} = \frac{2}{13}\mathbf{v}^{(1)} + \frac{4}{13}\mathbf{v}^{(2)} = \frac{2}{13} \begin{bmatrix} 3 \\ -9 \\ 1 \end{bmatrix} \qquad \square$$

As a final observation, note that the formulas

$$\mathbf{c} = (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{s}$$

and

$$\hat{\mathbf{s}} = \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{s}$$

which we derived for the case $n < m$ under the assumption of linear independence of the columns of $\mathbf{V}$, also give us the correct answers when $n = m$ and $\mathbf{V}^{-1}$ exists:

$$\mathbf{c} = (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{s} = \mathbf{V}^{-1}(\mathbf{V}^T)^{-1}\mathbf{V}^T\mathbf{s} = \mathbf{V}^{-1}\mathbf{s}$$

and

$$\hat{\mathbf{s}} = \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{s} = \mathbf{V}\mathbf{V}^{-1}\mathbf{s} = \mathbf{s}$$

2.11 Least-Squares Approximation in Practice

The basic technique of least-squares approximation developed in the previous section has numerous applications across the engineering disciplines and in most scientific fields where data analysis is important. Broadly speaking, least-squares methods allow us to to estimate parameters for a variety of models that have *linear structure*. To illustrate the adaptability of the least-squares technique, we consider two rather different examples.

Example 2.11.1. *Curve fitting.* We are given the following vector **s** consisting of ten measurements taken at regular time intervals (for simplicity, every second):

t	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
s	−0.7	−0.2	0.6	1.1	1.7	2.5	3.1	3.8	4.5	5.0

Suppose we are interested in fitting a straight line

$$\hat{s}(t) = at + b$$

through the discrete plot $\{(t_i, s_i),\ 1 \leq i \leq 10\}$ so as to minimize

$$\sum_{i=1}^{10} (\hat{s}(t_i) - s_i)^2$$

Letting $\hat{s}(t_i) = \hat{s}_i$, we see that the vector $\hat{\mathbf{s}}$ is given by

$$\hat{\mathbf{s}} = a\mathbf{t} + b\mathbf{1}$$

where $\mathbf{1}$ is a column vector of unit entries. We can then restate our problem as follows: choose coefficients a and b for $\mathbf{t}$ and $\mathbf{1}$, respectively, so as to minimize

$$\|\hat{\mathbf{s}} - \mathbf{s}\|^2$$

We thus have a least-squares problem where

- the length of the data vector is $m = 10$; and

- the number of reference vectors (i.e., $\mathbf{t}$ and $\mathbf{1}$) is $n = 2$.

Again, we begin by computing the matrix of inner products of the reference vectors, i.e., $\mathbf{V}^T\mathbf{V}$, where $\mathbf{V} = [\mathbf{t}\ \mathbf{1}]$. We have

$$\begin{aligned} \mathbf{t}^T\mathbf{t} &= \sum_{i=1}^{10} i^2 = 385 \\ \mathbf{t}^T\mathbf{1} &= \sum_{i=1}^{10} i = 55 \\ \mathbf{1}^T\mathbf{1} &= \sum_{i=1}^{10} 1 = 10 \end{aligned}$$

and therefore

$$\mathbf{V}^T\mathbf{V} = \begin{bmatrix} \mathbf{t}^T\mathbf{t} & \mathbf{t}^T\mathbf{1} \\ \mathbf{t}^T\mathbf{1} & \mathbf{1}^T\mathbf{1} \end{bmatrix} = \begin{bmatrix} 385 & 55 \\ 55 & 10 \end{bmatrix}$$

Also,

$$\begin{aligned} \mathbf{t}^T\mathbf{s} &= \sum_{i=1}^{10} i s_i = 171.2 \\ \mathbf{1}^T\mathbf{s} &= \sum_{i=1}^{10} s_i = 21.4 \end{aligned}$$

and thus

$$\mathbf{V}^T\mathbf{s} = \begin{bmatrix} \mathbf{t}^T\mathbf{s} \\ \mathbf{1}^T\mathbf{s} \end{bmatrix} = \begin{bmatrix} 171.2 \\ 21.4 \end{bmatrix}$$

The least squares solution is then given by

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 385 & 55 \\ 55 & 10 \end{bmatrix}^{-1} \begin{bmatrix} 171.2 \\ 21.4 \end{bmatrix} = \begin{bmatrix} 0.6485 \\ -1.4267 \end{bmatrix}$$

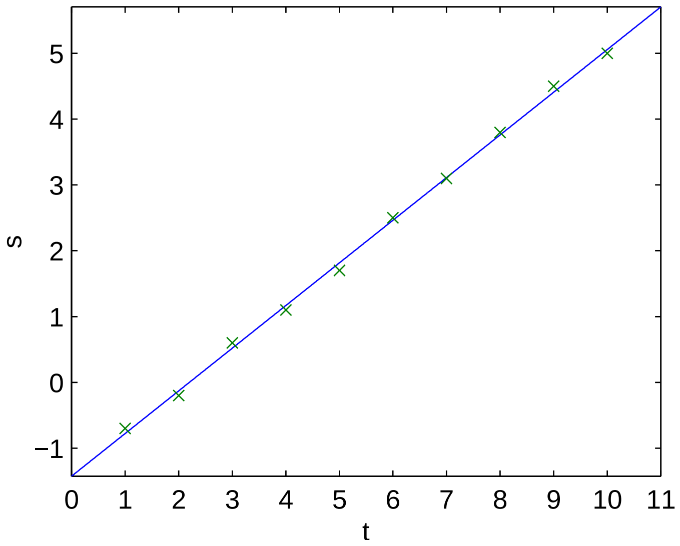
and the resulting straight line $\hat{s}(t) = at + b$ is plotted together with the discrete data points.

It can be seen that the straight-line fit is very good in this case. The mean (i.e., average) square error is given by

$$\frac{1}{10}\|\hat{\mathbf{s}} - \mathbf{s}\|^2 = 0.00501$$

Taking the square root

$$\sqrt{\frac{1}{10}\|\hat{\mathbf{s}} - \mathbf{s}\|^2} = 0.0708$$



Example 2.11.1

we obtain the *root mean square* (r.m.s.) error. The mean absolute error is

$$\frac{1}{10}\sum_{i=1}^{10}|\hat{s}_i - s_i| = 0.0648 \qquad \square$$

Remark. The same technique can be used for fitting any linear combination of functions through a discrete data set. For example, by expanding the set of reference vectors to include the squares, cubes, etc., of the entries of the abscissa (time in this case) vector $\mathbf{t}$, we can determine the optimal (in the least-squares sense) fit by a polynomial of any degree we choose.

Example 2.11.2. *Range estimation.* In this example, we assume that we have two signal sources, each emitting a pure sinusoidal waveform. The signals have the same amplitude (hence also power), but different frequencies and phase shifts. The sources are mobile, and their location at any time is unknown.

A receiver, who knows the two source frequencies Ω_1 and Ω_2, also knows that the signal power follows an *inverse-power law*, i.e., it decays as $R^{-\gamma}$, where R is the distance from the source and γ is a known positive constant. Based on this information, the receiver attempts to recover the range ratio R_1/R_2 (i.e., the receiver's relative distance from the two sources) from samples of the received signal

$$s(t) = \sigma R_1^{-\gamma/2} \cos(\Omega_1 t + \phi_1) + \sigma R_2^{-\gamma/2} \cos(\Omega_2 t + \phi_2) + z(t)$$

Here, $z(t)$ represents interference, or noise, that has no particular structure (at least none that can be used to improve the model for $s(t)$). Use of the common parameter σ here is consistent with the assumption that both sources emit at the same power.

Suppose that the receiver records m samples of $s(t)$ corresponding to times $t_1, \dots, t_m$. Since the distances R_1 and R_2 both appear in the amplitudes of the received sinusoidal components, it makes sense to estimate those amplitudes. Note that neither of the two components in their respective form shown above can be used as a reference signal for the least-squares solution—the unknown phase shifts must somehow be removed. To that end, we write

$$\cos(\Omega_k t + \phi_k) = \cos\phi_k \cos(\Omega_k t) - \sin\phi_k \sin(\Omega_k t)$$

and use two reference signals for each Ω_k, namely $\cos(\Omega_k t)$ and $\sin(\Omega_k t)$. In their discrete form, these signals are given by vectors

$$\begin{aligned}
\mathbf{u}^{(1)} &= \Big[\cos(\Omega_1 t_i)\Big]_{i=1}^{m} \\
\mathbf{w}^{(1)} &= \Big[\sin(\Omega_1 t_i)\Big]_{i=1}^{m} \\
\mathbf{u}^{(2)} &= \Big[\cos(\Omega_2 t_i)\Big]_{i=1}^{m} \\
\mathbf{w}^{(2)} &= \Big[\sin(\Omega_2 t_i)\Big]_{i=1}^{m}
\end{aligned}$$

We thus seek coefficients a_1, b_1, a_2 and b_2 such that

$$\hat{\mathbf{s}} = a_1\mathbf{u}^{(1)} + b_1\mathbf{w}^{(1)} + a_2\mathbf{u}^{(2)} + b_2\mathbf{w}^{(2)}$$

is the least-squares approximation (among all such linear combinations) to the vector of received samples $\mathbf{s}$. The solution is obtained in the usual fashion: if

$$\mathbf{V} = \begin{bmatrix} \mathbf{u}^{(1)} & \mathbf{w}^{(1)} & \mathbf{u}^{(2)} & \mathbf{w}^{(2)} \end{bmatrix}$$

then the optimal coefficients are given by

$$\begin{bmatrix} a_1 \\ b_1 \\ a_2 \\ b_2 \end{bmatrix} = (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{s}$$

Since, for each k, the sum $a_k^2+b_k^2$ is approximately equal to $\sigma^2 R_k^{-\gamma}(\cos^2\phi_k+\sin^2\phi_k)=\sigma^2 R_k^{-\gamma}$, the resulting estimate of the ratio R_1/R_2 is

$$\left(\frac{a_1^2+b_1^2}{a_2^2+b_2^2}\right)^{-1/\gamma} \qquad \Box$$

2.12 Orthogonality and Least-Squares Approximation

2.12.1 A Simpler Least-Squares Problem

The solution of the projection, or least-squares, problem is greatly simplified when the reference vectors $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}$ in $\mathbf{V}$ are (*pairwise*) *orthogonal.* This means that $\mathbf{v}^{(i)} \perp \mathbf{v}^{(j)}$ for any $i \neq j$, or equivalently,

$$\langle \mathbf{v}^{(i)}, \mathbf{v}^{(j)} \rangle = \begin{cases} \|\mathbf{v}^{(i)}\|^2, & i = j; \\ 0, & i \neq j. \end{cases}$$

Recall that $\langle \mathbf{v}^{(i)}, \mathbf{v}^{(j)} \rangle$ is the $(i,j)^{\text{th}}$ element of the inner product matrix $\mathbf{V}^T\mathbf{V}$. Thus if the columns of $\mathbf{V}$ are orthogonal, then

$$\mathbf{V}^T\mathbf{V} = \begin{bmatrix} \|\mathbf{v}^{(1)}\|^2 & 0 & \ldots & 0 \\ 0 & \|\mathbf{v}^{(2)}\|^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \|\mathbf{v}^{(n)}\|^2 \end{bmatrix}$$

and hence $\mathbf{V}^T\mathbf{V}$ is a diagonal matrix. Since none of $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}$ is an all-zeros vector (by the assumption of linear independence), it follows that all the squared norms on the diagonal are strictly positive, and thus

$$(\mathbf{V}^T\mathbf{V})^{-1} = \begin{bmatrix} \|\mathbf{v}^{(1)}\|^{-2} & 0 & \ldots & 0 \\ 0 & \|\mathbf{v}^{(2)}\|^{-2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \|\mathbf{v}^{(n)}\|^{-2} \end{bmatrix}$$

is well-defined. The projection $\hat{\mathbf{s}}$ of $\mathbf{s}$ on $\mathcal{R}(\mathbf{V})$ is then given by $\hat{\mathbf{s}} = \mathbf{V}\mathbf{c}$, where

$$\mathbf{c} = (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{s} = \begin{bmatrix} \|\mathbf{v}^{(1)}\|^{-2} & 0 & \ldots & 0 \\ 0 & \|\mathbf{v}^{(2)}\|^{-2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \|\mathbf{v}^{(n)}\|^{-2} \end{bmatrix} \begin{bmatrix} \langle \mathbf{v}^{(1)}, \mathbf{s} \rangle \\ \langle \mathbf{v}^{(2)}, \mathbf{s} \rangle \\ \vdots \\ \langle \mathbf{v}^{(n)}, \mathbf{s} \rangle \end{bmatrix}$$

i.e., for every i,

$$c_i = \frac{\langle \mathbf{v}^{(i)}, \mathbf{s} \rangle}{\|\mathbf{v}^{(i)}\|^2}$$

As a result,

$$\hat{\mathbf{s}} = \mathbf{V}\mathbf{c} = \sum_{i=1}^{n} \frac{\langle \mathbf{v}^{(i)}, \mathbf{s} \rangle}{\|\mathbf{v}^{(i)}\|^2} \mathbf{v}^{(i)}$$

The generic term in the above sum is the projection of $\mathbf{s}$ onto $\mathbf{v}^{(i)}$ (as we saw in Subsection 2.10.2). Thus *if the columns of* $\mathbf{V}$ *are orthogonal, then the projection of any vector* $\mathbf{s}$ *on* $\mathcal{R}(\mathbf{V})$ *is given by the (vector) sum of the projections of* $\mathbf{s}$ *onto each of the columns.*

This result is in agreement with three-dimensional geometry, as shown in Figure 2.11. It is also used extensively—and without elaboration—when dealing with projections on subspaces generated by two or more of the standard orthogonal unit vectors $\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(m)}$ in $\mathbf{R}^{m \times 1}$. For example, the projection of

$$\mathbf{s} = 3\mathbf{e}^{(1)} + 4\mathbf{e}^{(2)} + 7\mathbf{e}^{(3)}$$

on the plane generated by $\mathbf{e}^{(1)}$ and $\mathbf{e}^{(2)}$ equals

$$\hat{\mathbf{s}} = 3\mathbf{e}^{(1)} + 4\mathbf{e}^{(2)} ,$$

which is the sum of the projections of $\mathbf{s}$ on $\mathbf{e}^{(1)}$ (given by $3\mathbf{e}^{(1)}$) and on $\mathbf{e}^{(2)}$ (given by $4\mathbf{e}^{(2)}$).

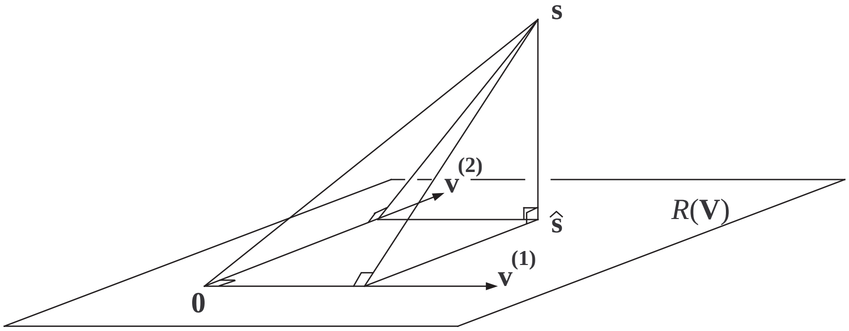


Figure 2.11: Projection on a plane generated by orthogonal vectors $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$.

2.12.2 Generating Orthogonal Reference Vectors

The result of the previous subsection clearly demonstrates the advantage of working with orthogonal reference vectors $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}$: least squares approximations can be obtained without solving simultaneous equations.

As it turns out, we can always work with orthogonal reference vectors, if we so choose:

Fact. *Any linearly independent set* $\{\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(n)}\}$ *of vectors in* $\mathbf{R}^{m\times 1}$ *(where* $n \leq m$*) is equivalent to a linearly independent* ***and orthogonal*** *set* $\{\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(n)}\}$*, where equivalence means that both sets produce the same subspace of linear combinations.* □

In matrix terms, if

$$\mathbf{V} = \left[\begin{array}{ccc} \mathbf{v}^{(1)} & \dots & \mathbf{v}^{(n)} \end{array}\right] ,$$

we claim that there exists

$$\mathbf{W} = \left[\begin{array}{ccc} \mathbf{w}^{(1)} & \dots & \mathbf{w}^{(n)} \end{array}\right]$$

such that $\mathbf{W}^T\mathbf{W}$ is diagonal and

$$\mathcal{R}(\mathbf{V}) = \mathcal{R}(\mathbf{W})$$

The last condition means that each $\mathbf{v}^{(i)}$ is expressible as a linear combination of $\mathbf{w}^{(j)}$'s and vice versa (with $\mathbf{v}$ and $\mathbf{w}$ interchanged). In other words, there exists a nonsingular $n \times n$ matrix $\mathbf{B}$ such that

$$\mathbf{V} = \mathbf{W}\mathbf{B}$$

and (equivalently)

$$\mathbf{W} = \mathbf{V}\mathbf{B}^{-1}$$

We will now show how, given any $\mathbf{V}$ with linearly independent columns, we can obtain such a matrix $\mathbf{B}$. The construction is based on the LU factorization of $\mathbf{V}^T\mathbf{V}$, a matrix known to be be

- *nonsingular*;
- *symmetric*; and
- *positive definite*, meaning that $\mathbf{x}^T\mathbf{V}^T\mathbf{V}\mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$.

The three properties listed above have the following implications (proofs are omitted):

- Nonsingularity of $\mathbf{V}^T\mathbf{V}$ implies that a *unique* normalized LU factorization of the form
$$\mathbf{V}^T\mathbf{V} = \mathbf{L}\mathbf{D}\mathbf{U}$$
exists where both $\mathbf{L}$ and $\mathbf{U}$ have unit diagonal elements and $\mathbf{D}$ has nonzero diagonal elements.

- Symmetry of of $\mathbf{V}^T\mathbf{V}$ in conjunction with the uniqueness of the above factorization implies
$$\mathbf{L} = \mathbf{U}^T$$
- Finally, positive definiteness of $\mathbf{V}^T\mathbf{V}$ implies that $\mathbf{D}$ has (strictly) positive diagonal elements.

The following example illustrates the above-mentioned features of the LU factorization of $\mathbf{V}^T\mathbf{V}$.

Example 2.12.1. Consider the matrix

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}^{(1)} & \mathbf{v}^{(2)} & \mathbf{v}^{(3)} \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 \\ 0 & 1 & -2 \\ 1 & 1 & -2 \\ -1 & 1 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$

Gaussian elimination on

$$\mathbf{V}^T\mathbf{V} = \begin{bmatrix} 4 & -4 & 2 \\ -4 & 13 & -14 \\ 2 & -14 & 18 \end{bmatrix}$$

yields

m	x_1	x_2	x_3
	$\underline{4}$	-4	2
1	-4	13	-14
$-1/2$	2	-14	18
	4	-4	2
	0	$\underline{9}$	-12
$4/3$	0	-12	17
	4	-4	2
	0	9	-12
	0	0	1

Therefore

$$\begin{aligned} \mathbf{V}^T\mathbf{V} &= \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1/2 & -4/3 & 1 \end{bmatrix} \begin{bmatrix} 4 & -4 & 2 \\ 0 & 9 & -12 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1/2 & -4/3 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 1/2 \\ 0 & 1 & -4/3 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

demonstrating that $\mathbf{V}^T\mathbf{V} = \mathbf{U^T}\mathbf{D}\mathbf{U}$, where $\mathbf{D}$ has strictly positive diagonal entries. □

Using the above-mentioned properties of $\mathbf{V}^T\mathbf{V}$, we will now show that the matrix

$$\mathbf{W} = \mathbf{V}\mathbf{U}^{-1}$$

has n nonzero orthogonal columns, thereby proving the claim made earlier (with $\mathbf{B} = \mathbf{U}$). Indeed,

$$\begin{aligned}
\mathbf{W}^T\mathbf{W} &= (\mathbf{V}\mathbf{U}^{-1})^T\mathbf{V}\mathbf{U}^{-1} \\
&= (\mathbf{U}^T)^{-1}\mathbf{V}^T\mathbf{V}\mathbf{U}^{-1} \\
&= (\mathbf{U}^T)^{-1}\mathbf{U}^T\mathbf{D}\mathbf{U}\mathbf{U}^{-1} \\
&= \mathbf{D}
\end{aligned}$$

and thus $\mathbf{W}$ has n orthogonal columns $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(n)}$ with norms given by

$$\|\mathbf{w}^{(i)}\| = \sqrt{d_{ii}} > 0$$

for all i.

Example 2.12.1. (*Continued.*) There is no need to compute $\mathbf{U}^{-1}$ explicitly in order to determine $\mathbf{W} = \mathbf{V}\mathbf{U}^{-1}$. We have, equivalently,

$$\mathbf{W}\mathbf{U} = \mathbf{V} \qquad \Leftrightarrow \qquad \mathbf{U}^T\mathbf{W}^T = \mathbf{V}^T$$

The last equation can be written as (note that $\mathbf{U}^T = \mathbf{L}$)

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1/2 & -4/3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{w}^{(1)T} \\ \mathbf{w}^{(2)T} \\ \mathbf{w}^{(3)T} \end{bmatrix} = \begin{bmatrix} \mathbf{v}^{(1)T} \\ \mathbf{v}^{(2)T} \\ \mathbf{v}^{(3)T} \end{bmatrix}$$

We solve this lower triangular system using standard forward elimination, treating the row vectors $\mathbf{w}^{(\cdot)T}$ and $\mathbf{v}^{(\cdot)T}$ as scalar variables and omitting the transpose throughout. Thus

$$\begin{aligned}
\mathbf{w}^{(1)} &= \mathbf{v}^{(1)} \\
\mathbf{w}^{(2)} &= \mathbf{v}^{(1)} + \mathbf{v}^{(2)} \\
\mathbf{w}^{(3)} &= -\frac{1}{2}\mathbf{v}^{(1)} + \frac{4}{3}(\mathbf{v}^{(1)} + \mathbf{v}^{(2)}) + \mathbf{v}^{(3)} = \frac{5}{6}\mathbf{v}^{(1)} + \frac{4}{3}\mathbf{v}^{(2)} + \mathbf{v}^{(3)}
\end{aligned}$$

We have obtained three orthogonal vectors with norms 2, 3 and 1 respectively. In extensive form,

$$\mathbf{W} = \left[\begin{array}{ccc} \mathbf{w}^{(1)} & \mathbf{w}^{(2)} & \mathbf{w}^{(3)} \end{array} \right] = \left[\begin{array}{rrr} -1 & 2 & 1/6 \\ 0 & 1 & -2/3 \\ 1 & 2 & 1/6 \\ -1 & 0 & -1/2 \\ -1 & 0 & 1/2 \end{array} \right] \qquad \square$$

We make the following final remarks on this topic.

- Dividing each $\mathbf{w}^{(i)}$ in the above transformation by its norm $\|\mathbf{w}^{(i)}\| = \sqrt{d_{ii}}$, we obtain a new set of orthogonal vectors, each having unit norm. *A set of vectors having these two properties (orthogonality and unit norm) is known as orthonormal.* The new matrix $\mathbf{W}$ is related to $\mathbf{V}$ via

  $$\mathbf{V} = \mathbf{W}\mathbf{D}^{1/2}\mathbf{U}$$

  and

  $$\mathbf{W} = \mathbf{V}\mathbf{U}^{-1}\mathbf{D}^{-1/2}$$

  where $\mathbf{D}^{1/2}$ and $\mathbf{D}^{-1/2}$ are diagonal matrices obtained by taking the same (respective) powers of the diagonal elements of $\mathbf{D}$.

- The upper triangular matrix $\mathbf{D}^{1/2}\mathbf{U}$ above can be computed in MATLAB using the function `CHOL`, with argument given by $\mathbf{V}^T\mathbf{V}$. This the same as producing the *Cholesky* form of the LU factorization of a positive definite symmetric matrix $\mathbf{A}$, where the lower and upper triangular parts are transposes of each other:

  $$\mathbf{A} = (\mathbf{D}^{1/2}\mathbf{U})^T(\mathbf{D}^{1/2}\mathbf{U})$$

2.13 Complex-Valued Matrices and Vectors

2.13.1 Similarities to the Real-Valued Case

Thus far we have considered matrices and vectors with real-valued entries only. From an applications perspective, this restriction is justifiable, since real-world signals are real-valued—it therefore makes sense to represent or approximate them using real-valued reference vectors. One class of reference vectors of particular importance in signal analysis comprises segments of sinusoidal signals in discrete time. As the analysis of sinusoids is simplified by considering complex sinusoids (or phasors), it pays to generalize the tools and concepts developed so far to include complex-valued matrices and vectors.

A $m \times n$ complex-valued matrix takes values in $\mathbf{C}^{m\times n}$, where $\mathbf{C}$ denotes the complex plane. Since each entry of the matrix consists of a real and an imaginary part, a complex-valued matrix stores *twice* as much information as a real-valued matrix of the same size. This fact has interesting implications about the representation of real-valued signals by complex-valued reference vectors.

Partitioning a matrix $\mathbf{A} \in \mathbf{C}^{m\times n}$ into its columns, we see that each column of $\mathbf{A}$ is a vector $\mathbf{z}$ in $\mathbf{C}^{m\times 1}$, which can be expressed as

$$\mathbf{z} = z_1\mathbf{e}^{(1)} + \cdots + z_m\mathbf{e}^{(m)}$$

Here, $\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(m)}$ are the standard (real-valued) unit vectors, and $z_1, \ldots, z_m$ are the complex-valued elements of $\mathbf{z}$:

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix}$$

Note that the "information-doubling" effect mentioned in the previous paragraph is due to the fact that each z_i is complex; the vector $\mathbf{z}$ itself has dimension m, as does any real-valued vector $\mathbf{x}$ (which is also a linear combination of $\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(m)}$, but with real-valued coefficients $x_1, \ldots, x_m$).

Complex-valued matrices and vectors behave in exactly the same manner as real-valued ones. Since all matrix computations are based on scalar addition and multiplication, one only needs to ensure that in the complex case, these (scalar) operations are replaced by complex addition and multiplication. We thus have the following straightforward extensions of properties and concepts defined earlier:

- *Matrix addition*: Defined as before (using complex addition).
- *Matrix multiplication*: Defined as before (using complex addition and multiplication).
- *Matrix transposition (T)*: Defined as before. No algebraic operations are needed here. A modified transpose will be defined in the next subsection.
- *Linear Independence*: Defined as before, allowing complex coefficients in linear combinations. Thus linear independence of the columns of $\mathbf{A}$ means that $\mathbf{z} = \mathbf{0}$ is the only solution of $\mathbf{Az} = \mathbf{0}$.
- *Matrix inverse*: As before, a square matrix is nonsingular (and thus has an inverse) provided its columns are linearly independent.
- *Solution of* $\mathbf{Az} = \mathbf{b}$: Gaussian elimination is still applicable. In terms of computational effort, this is a more intensive problem, since each complex addition involves two real additions; and each complex multiplication involves four real multiplications and three real additions. One can also write out the above matrix equation using real constants and unknowns; twice as many variables are needed, since each complex variable z_i has a real and an imaginary part.

2.13.2 Inner Products of Complex-Valued Vectors

The inner product of $\mathbf{v}$ and $\mathbf{w}$ in $\mathbf{C}^{m\times 1}$ is defined by

$$\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{i=1}^{m} v_i^* w_i = (\mathbf{v}^*)^T \mathbf{w}$$

where, as usual, the superscript * denotes complex conjugate. Note that if the two vectors are real-valued (i.e., have zero imaginary parts), the complex conjugate has no effect on the value of $\mathbf{v}$, and the above definition becomes that of the inner product of two real-valued vectors.

The combination of the complex conjugate $*$ and transpose T operators (in either order) is known as the *conjugate transpose*, or *Hermitian*, operator H. Thus

$$\langle \mathbf{v}, \mathbf{w} \rangle = \mathbf{v}^H \mathbf{w}$$

The following identities involving complex conjugates of scalars are useful in manipulating inner products and conjugate transposes of vectors:

$$(z^*)^* \quad = \quad z$$

$$\begin{aligned}(z_1+z_2)^* &= z_1^*+z_2^*\\ (z_1z_2)^* &= z_1^*z_2^*\\ |z|^2 &= zz^*\end{aligned}$$

We immediately obtain:

$$\langle \mathbf{w},\mathbf{v}\rangle = \sum_{i=1}^m w_i^* v_i = \left(\sum_{i=1}^m v_i^* w_i\right)^* = \langle \mathbf{v},\mathbf{w}\rangle^*$$

or equivalently,

$$\mathbf{w}^H\mathbf{v} = \left(\mathbf{v}^H\mathbf{w}\right)^*$$

Also, if c is a complex-valued scalar,

$$\begin{aligned}\langle \mathbf{v}, c\mathbf{w}\rangle &= c\langle \mathbf{v},\mathbf{w}\rangle\\ \langle c\mathbf{v}, \mathbf{w}\rangle &= c^*\langle \mathbf{v},\mathbf{w}\rangle\\ \langle c\mathbf{v}, c\mathbf{w}\rangle &= |c|^2\langle \mathbf{v},\mathbf{w}\rangle\end{aligned}$$

The norm (or length) of $\mathbf{v}$ in $\mathbf{C}^{m\times 1}$ is defined as

$$\|\mathbf{v}\| = \left(\sum_{i=1}^m |v_i|^2\right)^{1/2}$$

where $|\cdot|$ denotes the magnitude of the complex element v_i:

$$|v_i|^2 = v_i^* v_i = \Re e^2\{v_i\} + \Im m^2\{v_i\}$$

We thus see that $\|\mathbf{v}\|^2$ is given by the *sum of squares of all real and imaginary parts* contained in the vector $\mathbf{v}$. We also have

$$\|\mathbf{v}\| = \left(\sum_{i=1}^m v_i^* v_i\right)^{1/2} = \langle \mathbf{v},\mathbf{v}\rangle^{1/2}$$

which is *the same relationship between norm and inner product as in the real-valued case.* This is a key reason for introducing the complex conjugate in the definition of the inner product.

Example 2.13.1. Let

$$\mathbf{v} = \begin{bmatrix} 1+j\\ -j\\ 5-2j\end{bmatrix} \qquad \text{and} \qquad \mathbf{w} = \begin{bmatrix} 3\\ 1-j\\ 1+2j\end{bmatrix}$$

Then

$$\begin{aligned}\langle \mathbf{v}, \mathbf{w}\rangle &= \mathbf{v}^H\mathbf{w}\\ &= (1-j)\cdot 3 + j\cdot(1-j) + (5+2j)\cdot(1+2j) = 5+10j\end{aligned}$$

$$\langle \mathbf{w}, \mathbf{v}\rangle = \mathbf{w}^H\mathbf{v} = 5-10j$$

$$\begin{aligned}\|\mathbf{v}\|^2 &= \mathbf{v}^H\mathbf{v}\\ &= (1+1)+(1)+(25+4) = 32\end{aligned}$$

$$\begin{aligned}\|\mathbf{w}\|^2 &= \mathbf{w}^H\mathbf{w}\\ &= 9+(1+1)+(1+4) = 16\end{aligned}$$

□

Note that by separating complex vectors into their real and imaginary parts, i.e., $\mathbf{v} = \mathbf{a} + j\mathbf{b}$ and $\mathbf{w} = \mathbf{c} + j\mathbf{d}$, we can also express inner products and norms of complex vectors in terms or inner products of real vectors, e.g.,

$$\begin{aligned}\langle \mathbf{v}, \mathbf{w}\rangle &= (\mathbf{a}^T - j\mathbf{b}^T)(\mathbf{c} + j\mathbf{d})\\ &= \mathbf{a}^T\mathbf{c} + \mathbf{b}^T\mathbf{d} + j(\mathbf{a}^T\mathbf{d} - \mathbf{b}^T\mathbf{c})\end{aligned}$$

The following properties of the conjugate transpose H are identical to those of the transpose T:

$$\begin{aligned}(\mathbf{A}^H)^H &= \mathbf{A}\\ (\mathbf{AB})^H &= \mathbf{B}^H\mathbf{A}^H\\ (\mathbf{A}^{-1})^H &= (\mathbf{A}^H)^{-1}\end{aligned}$$

As a final reminder, in MATLAB:

- `.'` denotes T and `'` denotes H.
- In computations involving real matrices (exclusively), the same result will be obtained by either transpose.
- Where complex matrices are involved, the conjugate transpose (H or `'`) is usually needed.
- For changing the (row-column) orientation of complex vectors, the ordinary transpose `.'` *must* be used; use of `'` will result in an error (due to the conjugation).

2.13.3 The Least-Squares Problem in the Complex Case

Orthogonality of complex-valued vectors is defined in terms of their inner product:

$$\mathbf{v} \perp \mathbf{w} \quad \Leftrightarrow \quad \langle \mathbf{v}, \mathbf{w} \rangle = 0 \quad \Leftrightarrow \quad \langle \mathbf{w}, \mathbf{v} \rangle = 0$$

or equivalently:

$$\mathbf{v} \perp \mathbf{w} \quad \Leftrightarrow \quad \mathbf{v}^H \mathbf{w} = 0 \quad \Leftrightarrow \quad \mathbf{w}^H \mathbf{v} = 0$$

This is essentially the same definition as in the real-valued case.

Remark. Note that in geometric terms, it is considerably more difficult to visualize $\mathbf{C}^{m\times 1}$ than $\mathbf{R}^{m\times 1}$, and even simple concepts such as orthogonality can lead to intuitive pitfalls. For example, the scalars $v = 1$ and $w = j$ are at right angles to each other on the complex plane; yet viewed as vectors $\mathbf{v}$ and $\mathbf{w}$ in $\mathbf{C}^{1\times 1}$, they are not orthogonal since $\langle \mathbf{v}, \mathbf{w} \rangle = j$.

With the given definitions of the inner product and orthogonality for complex vectors, the formulation and solution of the least-squares approximation problem in the complex case turns out to be the same as in the real case. Assuming that $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}$ are linearly independent vectors in $\mathbf{C}^{m\times 1}$ (where $n \le m$), we seek to approximate $\mathbf{s} \in \mathbf{C}^{m\times 1}$ by $\mathbf{Vc}$ so that the squared norm

$$\|\mathbf{Vc} - \mathbf{s}\|^2$$

(i.e., the sum of squares of all real and imaginary parts contained in the error vector $\mathbf{Vc} - \mathbf{s}$) is minimized.

Following the same argument as in the real-valued case, we define the projection $\hat{\mathbf{s}}$ of $\mathbf{s}$ on $\mathcal{R}(\mathbf{V})$ by the following n relationships:

$$\hat{\mathbf{s}} - \mathbf{s} \perp \mathbf{v}^{(i)} \quad \Leftrightarrow \quad (\mathbf{v}^{(i)})^H \hat{\mathbf{s}} = (\mathbf{v}^{(i)})^H \mathbf{s}$$

where $i = 1, \ldots, n$. Equivalently, in matrix form we have

$$\mathbf{V}^H \hat{\mathbf{s}} = \mathbf{V}^H \mathbf{s}$$

and letting $\hat{\mathbf{s}} = \mathbf{Vc}$ as before, we obtain

$$\mathbf{V}^H \mathbf{Vc} = \mathbf{V}^H \mathbf{s}$$

Since H is the extension of T to the complex case, we have, in effect, the same formula for computing projections as before.

The proof that the projection $\hat{\mathbf{s}}$ on $\mathcal{R}(\mathbf{V})$ solves the least-squares approximation problem for complex vectors is *identical* to the proof given in

Subsection 2.10.4 for real vectors. Care should be exercised in writing out one equation: if $\mathbf{y}$ is the competing approximation, then

$$\|\mathbf{y}-\mathbf{s}\|^2 = \|\hat{\mathbf{s}}-\mathbf{s}\|^2 + \|\mathbf{y}-\hat{\mathbf{s}}\|^2 + \langle \mathbf{y}-\hat{\mathbf{s}},\, \hat{\mathbf{s}}-\mathbf{s}\rangle + \langle \hat{\mathbf{s}}-\mathbf{s},\, \mathbf{y}-\hat{\mathbf{s}}\rangle$$

The last two terms on the right-hand side (which are conjugates of each other) are, again, equal to zero by the assumption of orthogonality of $\hat{\mathbf{s}}-\mathbf{s}$ and any vector in $\mathcal{R}(\mathbf{V})$.

Problems

Section 2.1

***P* 2.1.** (MATLAB) Enter the matrix

```
A = [1 2 3 4; 5 6 7 8; 9 10 11 12]
```

(i) Find two-element arrays `I` and `J` such that `A(I,J)` is a 2×2 matrix consisting of the corner elements of `A`.

(ii) Suppose that `B=A` initially. Find two-element arrays `K` and `L` such that

```
B(:,K) = B(:,L)
```

swaps the first and fourth columns of `B`.

(iii) Explain the result of

```
C = A(:)
```

(iv) Study the function `RESHAPE`. Use it together with the transpose operator `.'` in a single (one-line) command to generate the matrix

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

from `A`.

***P* 2.2.** (MATLAB) Study the functions `FLIPUD`, `HILB` and `TOEPLITZ`.

Let `N` be integer. How would you define the vector `r` so that the following command string returns an all-zero `N`-by-`N` matrix?

```
R = 1./r;
A = toeplitz(R);
B = flipud(A);
B(1:N,1:N) - hilb(N)
```

Section 2.2

***P* 2.3.** If

$$\mathbf{B}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \\ -1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{B}\begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 7 \\ 2 \end{bmatrix},$$

determine

$$\mathbf{B}\begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

***P* 2.4.** Let $\mathbf{G}$ be a $m \times 2$ matrix such that

$$\mathbf{G}\begin{bmatrix} -1 \\ 1 \end{bmatrix} = \mathbf{u} \qquad \text{and} \qquad \mathbf{G}\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \mathbf{v}$$

Express

$$\mathbf{G}\begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

in terms of $\mathbf{u}$ and $\mathbf{v}$.

***P* 2.5.** The transformation $A : \mathbf{R}^3 \mapsto \mathbf{R}^3$ is such that

$$\mathbf{A}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ 2 \end{bmatrix}, \qquad \mathbf{A}\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -2 \\ 3 \\ -1 \end{bmatrix}, \qquad \mathbf{A}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix}$$

Write out the matrix $\mathbf{A}$ and compute $\mathbf{Ax}$, where

$$\mathbf{x} = \begin{bmatrix} 2 & 5 & -1 \end{bmatrix}^T$$

(ii) The transformation $B : \mathbf{R}^3 \mapsto \mathbf{R}^2$ is such that

$$\mathbf{B}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ -4 \end{bmatrix}, \qquad \mathbf{B}\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \end{bmatrix}, \qquad \mathbf{B}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

Determine the matrix $\mathbf{B}$.

***P* 2.6.** Compute by hand the matrix product $\mathbf{AB}$ in the following cases:

$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ 3 & -2 \\ 1 & 5 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} -2 & -3 & 3 \\ 4 & 0 & 7 \end{bmatrix};$$

$$\mathbf{A} = \begin{bmatrix} 2 & 0 & 0 \\ -2 & 4 & 0 \\ 3 & -5 & 1 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 2 & -2 & 3 \\ 0 & 4 & -5 \\ 0 & 0 & 1 \end{bmatrix}$$

***P* 2.7.** If

$$\mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

find vectors $\mathbf{u}$ and $\mathbf{v}$ such that

$$\mathbf{u}^T\mathbf{A}\mathbf{v} = b$$

***P* 2.8.** (MATLAB) Generate a 60 × 60 matrix `A` as follows:

```
c = zeros(1,60);
c(1) = -1; c(2) = 1;
r=c;
A = toeplitz(c,r);
```

(i) Write a command that displays the top left 6 × 6 block of `A`.

(ii) Generate four sinusoidal vectors `x1`, `x2`, `x3` and `x4` as follows:

```
n = 1:60;
x1 = cos(0*n)';
x2 = cos(pi*n/6)';
x3 = cos(pi*n/3)';
x4 = cos(pi*n/2)';
```

and compute the products

```
y1=A*x1 ; y2=A*x2 ; y3=A*x3 ; y4=A*x4;
```

(iii) Use the `SUBPLOT` feature to display bar graphs of all eight vectors (against `n`) on the same figure window.

(iv) One of the eight vectors consists mostly of zero values. Explain mathematically why this is so.

Section 2.3

***P* 2.9.** Consider the matrices

$$\mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{bmatrix} i & g & h \\ f & d & e \\ c & a & b \end{bmatrix}$$

Express $\mathbf{B}$ as $\mathbf{P}^{(r)}\mathbf{A}\mathbf{P}^{(c)}$, where $\mathbf{P}^{(r)}$ and $\mathbf{P}^{(c)}$ are permutation matrices.

***P* 2.10.** Let

$$\mathbf{x} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

If $\mathbf{P}$ and $\mathbf{Q}$ are 4×4 matrices such that

$$\mathbf{P}\mathbf{x} = \begin{bmatrix} c \\ a \\ b \\ d \end{bmatrix} \qquad \text{and} \qquad \mathbf{Q}\mathbf{x} = \begin{bmatrix} a \\ c \\ d \\ b \end{bmatrix}$$

determine the product $\mathbf{PQ}$.

***P* 2.11.** Matrices $\mathbf{A}$ and $\mathbf{B}$ are generated in MATLAB using the commands

```
A = [1:3; 4:6; 7:9] ;
B = [9:-1:7; 6:-1:4; 3:-1:1] ;
```

Find matrices $\mathbf{P}$ and $\mathbf{Q}$ such that $\mathbf{B} = \mathbf{PAQ}$.

***P* 2.12.** **(i)** Find a $(n \times n)$ matrix $\mathbf{P}$ such that if

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 & \ldots & x_n \end{bmatrix}^T ,$$

then

$$\mathbf{P}\mathbf{x} = \begin{bmatrix} x_n & x_{n-1} & x_{n-2} & \ldots & x_1 \end{bmatrix}^T .$$

(ii) Without performing any calculations, determine the r^{th} power $\mathbf{P}^r$ for any integer r.

(iii) Let the vector $\mathbf{x}$ be given by

$$x[k] = \cos(\omega k + \phi) , \qquad (1 \leq k \leq n)$$

Under what conditions on ω, ϕ and n is the relationship $\mathbf{P}\mathbf{x} = \mathbf{x}$ true?

***P* 2.13.** Express each of the matrices

$$\mathbf{A} = \begin{bmatrix} a & b & c & d \\ 2a & 2b & 2c & 2d \\ -a & -b & -c & -d \\ -2a & -2b & -2c & -2d \end{bmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{bmatrix} 1 & t & t^2 & t^3 \\ t^{-1} & 1 & t & t^2 \\ t^{-2} & t^{-1} & 1 & t \\ t^{-3} & t^{-2} & t^{-1} & 1 \end{bmatrix}$$

as a product of a row vector and a column vector.

Sections 2.4–2.5

***P* 2.14.** Consider the matrix

$$\mathbf{A} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

which represents a counterclockwise rotation by an angle θ.

(i) Without performing any matrix operations, derive $\mathbf{A}^{-1}$.

(ii) Verify that $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$ by computing the product on the left-hand side of the equation.

Section 2.6

***P* 2.15.** Solve the simultaneous equations

$$\textbf{(i)} \qquad \begin{aligned} x_1 + \frac{x_2}{2} + \frac{x_3}{3} &= 14 \\ \frac{x_1}{2} + \frac{x_2}{3} + \frac{x_3}{4} &= 1 \\ \frac{x_1}{3} + \frac{x_2}{4} + \frac{x_3}{5} &= -2 \end{aligned}$$

$$\textbf{(ii)} \qquad \begin{aligned} 3x_1 - 5x_2 - 5x_3 &= -37 \\ 5x_1 - 5x_2 - 2x_3 &= -17 \\ 2x_1 + 3x_2 + 4x_3 &= 32 \end{aligned}$$

using Gaussian elimination.

Section 2.7

***P* 2.16.** **(i)** Determine the LU and LDV factorizations of

$$\mathbf{A} = \begin{bmatrix} 4 & 8 & 4 & 0 \\ 1 & 5 & 4 & -3 \\ 1 & 4 & 7 & 2 \\ 1 & 3 & 0 & -2 \end{bmatrix}$$

(ii) Solve $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{b} = [28\ 13\ 23\ 4]^T$, by means of the two triangular systems

$$\mathbf{Ly} = \mathbf{b} \qquad \text{and} \qquad \mathbf{Ux} = \mathbf{y}$$

***P* 2.17.** Repeat P 2.16 for

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 3 \\ 2 & 1 & -1 & 1 \\ 3 & -1 & -1 & 2 \\ -1 & 2 & 6 & -1 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 4 \\ 1 \\ -3 \\ 4 \end{bmatrix}$$

Section 2.8

***P* 2.18.** **(i)** Use row pivoting to solve the simultaneous equations $\mathbf{Ax} = \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 4 & 1 \\ 2 & 8 & 6 & 6 \\ 3 & 10 & 8 & 8 \\ 4 & 12 & 10 & 6 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 21 \\ 52 \\ 79 \\ 82 \end{bmatrix}$$

(ii) Determine the permuted LU factorization $\mathbf{LU} = \mathbf{PA}$.

***P* 2.19.** Consider the simultaneous equations $\mathbf{Ax} = \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} 0.002 & 3.420 \\ 2.897 & 3.412 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 3.422 \\ 6.309 \end{bmatrix}$$

The exact solution is $x_1 = 1$, $x_2 = 1$. Solve using finite-precision Gaussian elimination (a) without row pivoting; (b) with row pivoting. Round to four significant digits after each computation.

Section 2.9

***P* 2.20.** Use Gaussian elimination to determine the inverses of

$$\mathbf{A} = \begin{bmatrix} 3 & 2 & 1 \\ -1 & 5 & 2 \\ 2 & 3 & 1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{bmatrix} 1 & 2 & 4 & 1 \\ 2 & 8 & 6 & 6 \\ 3 & 10 & 8 & 8 \\ 4 & 12 & 10 & 6 \end{bmatrix}$$

***P* 2.21.** Consider the following MATLAB script:

```
A = [1 2 -4; -1 -1 5; 2 7 -3];
I = [1 0 0; 0 1 0; 0 0 1];
X = A\I
```

Using Gaussian elimination, determine (by hand) the answer `X`.

***P* 2.22.** Consider the simultaneous equations $\mathbf{Ax} = \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} 3.000 & -4.000 & 2.000 \\ -1.000 & 3.000 & -2.000 \\ 1.001 & 1.999 & -2.001 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 1.000 \\ 1.000 \\ 3.000 \end{bmatrix}$$

Assuming that the entries of $\mathbf{A}$ are precise, examine how the solution $\mathbf{x}$ is affected by rounding the entries of $\mathbf{b}$ to four significant digits. To that end, let each entry of $\mathbf{b}$ vary by ±.0005. For each of the $2^3 = 8$ extremal values, compute the solution $\mathbf{x}$ using the \ (backslash) operator in MATLAB, and determine the maximum distance from the original solution (i.e., compute `norm(x-A\b)`).

***P* 2.23.** The $n \times n$ Hilbert matrix $\mathbf{A}$, whose entries are given by

$$a_{ij} = \frac{1}{i + j - 1},$$

is a classic example of an ill-conditioned matrix. This problem illustrates the effect of perturbing each entry of $\mathbf{A}$ by a small random amount.

(i) Use the MATLAB function `HILB` to display the 5×5 Hilbert matrix. The command `format rat` will display it in rational form. Also, use the function `INVHILB` to display its inverse, which has integer entries.

(ii) Enter

```
format long
n = 5;
A = hilb(n);
b = ones(n,1)
c = b + 0.001*rand(n,1)
```

Compare the values

```
max(abs(b-c))
```

and

```
max(abs((A\c)./(A\b)-1))
```

Repeat for `n=10`. What do you observe?

Section 2.10

***P* 2.24.** Consider the four-dimensional vectors $\mathbf{a} = [\ -1 \quad 7 \quad 2 \quad 4\ ]^T$ and $\mathbf{b} = [\ 3 \quad 0 \quad -1 \quad -5\ ]^T$.

(i) Compute $\|\mathbf{a}\|$, $\|\mathbf{b}\|$ and $\|\mathbf{b}-\mathbf{a}\|$.

(ii) Compute the angle θ between $\mathbf{a}$ and $\mathbf{b}$ (where $0 \leq \theta \leq \pi$).

(iii) Let $\mathbf{f}$ be the projection of $\mathbf{b}$ on $\mathbf{a}$, and $\mathbf{g}$ be the projection of $\mathbf{a}$ on $\mathbf{b}$. Express $\mathbf{f}$ and $\mathbf{g}$ as $\lambda\mathbf{a}$ and $\mu\mathbf{b}$, respectively (λ and μ are scalars).

(iv) Verify that $\mathbf{b} - \mathbf{f} \perp \mathbf{a}$ and $\mathbf{a} - \mathbf{g} \perp \mathbf{b}$.

***P* 2.25.** Consider the three-dimensional vectors $\mathbf{v}^{(1)} = [\ -1 \quad 1 \quad 1\ ]^T$, $\mathbf{v}^{(2)} = [\ 2 \quad -1 \quad 3\ ]^T$, and $\mathbf{s} = [\ 1 \quad 2 \quad 3\ ]^T$.

(i) Determine the projection $\hat{\mathbf{s}}$ of $\mathbf{s}$ on the plane defined by $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$.

(ii) Show that the projection of $\hat{\mathbf{s}}$ on $\mathbf{v}^{(1)}$ is the same as the projection of $\mathbf{s}$ on $\mathbf{v}^{(1)}$. (Is this result expected from three-dimensional geometry?)

***P* 2.26.** Let $\mathbf{a}^{(1)}$, $\mathbf{a}^{(2)}$, $\mathbf{a}^{(3)}$ and $\mathbf{a}^{(4)}$ be the columns of the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 1/2 & 1/4 & 1/8 \\ 0 & 1 & 1/2 & 1/4 \\ 0 & 0 & 1 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Determine the least squares approximation

$$\mathbf{p} = c_1\mathbf{a}^{(1)} + c_2\mathbf{a}^{(2)} + c_3\mathbf{a}^{(3)}$$

of $\mathbf{a}^{(4)}$ based on $\mathbf{a}^{(1)}$, $\mathbf{a}^{(2)}$ and $\mathbf{a}^{(3)}$. Also determine the relative (root mean square) error

$$\frac{\|\mathbf{p}-\mathbf{a}^{(4)}\|}{\|\mathbf{a}^{(4)}\|}$$

Section 2.11

***P* 2.27.** Consider the data set

t	0.10	0.15	0.22	0.34	0.44	0.51	0.62	0.75	0.89	1.00
s	0.055	0.358	0.637	1.073	1.492	1.677	2.181	2.299	2.862	3.184

Find the least squares approximation to the data set shown above in terms of

(i) a straight line $f_1(t) = a_1t + a_0$;

(ii) a quadratic $f_2(t) = b_2t^2 + b_1t + b_0$.

Plot two graphs, each showing the discrete data set and the approximating function.

***P* 2.28.** Consider the data set

t	-2	-1	0	1	2
s	-0.2	-0.2	0.8	1.6	3.1

Let $f(t) = c_1t^2 + c_2t + c_3$ be the parabola that yields the least squares fit to the above data set, and let $\mathbf{c} = [c_1\ c_2\ c_3]^T$.

(i) Determine a matrix **A** and a vector **b** such that $\mathbf{Ac} = \mathbf{b}$.

(ii) Is it possible to deduce the value of c_2 independently of the other variables c_1 and c_3 (i.e., without having to solve the entire system of equations)? If so, what is the value of c_2?

***P* 2.29.** Five measurements x_i taken at times t_i are shown below.

t_i	0.1	0.2	0.3	0.4	0.5
x_i	2.88	3.48	4.29	5.00	6.25

It is believed that the data follow an exponential law of the form

$$x(t) = ae^{bt}$$

or equivalently,

$$\ln x(t) = \ln a + bt$$

Determine the values of a and b in the least squares approximation of the vector $[s_i] = [\ln x_i]$ by the vector $[\ln a + bt_i]$ (where $i = 1, \ldots, 5$).

***P* 2.30.** Five measurements x_i taken at times t_i are shown below.

t_i	10	20	30	40	50
x_i	14.0	20.9	26.7	32.2	36.8

It is believed that the data obey a power law of the form

$$x(t) = at^r$$

or equivalently,

$$\ln x(t) = \ln a + r \ln t$$

Determine the values of a and r in the least squares approximation of the vector $[s_i] = [\ln x_i]$ by the vector $[\ln a + r \ln t_i]$ (where $i = 1, \ldots, 5$).

***P* 2.31.** The transient response of a passive circuit has the form

$$y(t) = a_1 e^{-1.5t} + a_2 e^{-1.2t} \qquad (t \geq 0)$$

where t is in seconds. The constants a_1 and a_2 provide information about the initial state (at $t = 0$) of the circuit.

The data vector $\mathbf{s}$ in `s1.txt` contains noisy measurements of $y(t)$ taken every 200 milliseconds starting at $t = 0$. Find the values a_1 and a_2 which provide the least squares approximation $\hat{\mathbf{s}}$ to the data $\mathbf{s}$. Plot $\mathbf{s}$ and $\hat{\mathbf{s}}$ on the same graph.

***P* 2.32.** The data vector $\mathbf{s}$ can be found in `s2.txt`. It consists of noisy samples of the sinusoid

$$x(t) = a_1 \cos(200\pi t) + a_2 \sin(200\pi t) + b_1 \cos(250\pi t) + b_2 \sin(250\pi t) + c$$

taken every 100 microseconds (starting at $t = 0$). Find the values of a_1, a_2, b_1, b_2 and c which provide the least squares approximation $\hat{\mathbf{s}}$ to the data $\mathbf{s}$. Plot $\mathbf{s}$ and $\hat{\mathbf{s}}$ on the same graph.

Section 2.12

***P* 2.33.** Let $\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$ and $\mathbf{v}^{(3)}$ be the columns of the matrix

$$\mathbf{V} = \begin{bmatrix} 2 & -1 & 3 \\ 4 & 1 & -1 \\ -1 & 2 & 2 \end{bmatrix}$$

(i) Show that $\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$ and $\mathbf{v}^{(3)}$ (and thus also $\mathbf{V}$ itself) are orthogonal. Display $\mathbf{V}^T\mathbf{V}$.

(ii) Without performing Gaussian elimination, solve

$$\mathbf{Vc} = \mathbf{s}$$

for $\mathbf{s} = [\ 7 \quad 2 \quad -5\ ]^T$.

(iii) Determine the projection $\hat{\mathbf{s}}$ of $\mathbf{s}$ on the plane defined by $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$. What is the relationship between the error vector $\hat{\mathbf{s}} - \mathbf{s}$ and the projection of $\mathbf{s}$ onto $\mathbf{v}^{(3)}$?

(iv) Scale the columns of $\mathbf{V}$ so as to obtain an orthonormal matrix.

***P* 2.34.** Consider the matrix

$$\mathbf{V} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix}$$

(i) Compute the inner product matrix $\mathbf{V}^T\mathbf{V}$.

(ii) If

$$\mathbf{x} = \begin{bmatrix} 0 & 1 & -2 & 3 \end{bmatrix}^T$$

determine a vector $\mathbf{c}$ such that $\mathbf{Vc} = \mathbf{x}$.

***P* 2.35.** Consider the 4×3 matrix

$$\mathbf{V} = \begin{bmatrix} 1 & 1/2 & 1/4 \\ 0 & 1 & 1/3 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

(i) The 3×3 inner product matrix $\mathbf{V}^T\mathbf{V}$ has an obvious normalized LU factorization of the form

$$\mathbf{V}^T\mathbf{V} = \mathbf{LDU}$$

where the diagonal entries of $\mathbf{L}$ and $\mathbf{U}$ are unity and $\mathbf{U} = \mathbf{L}^T$. What are $\mathbf{L}$, $\mathbf{D}$ and $\mathbf{U}$ in that factorization?

(ii) Show how the orthonormal matrix

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

can be formed by taking linear combinations of the columns of $\mathbf{V}$.

***P* 2.36.** Consider the symmetric matrix

$$\mathbf{A} = \begin{bmatrix} 1 & -3 & 2 \\ -3 & 7 & -4 \\ 2 & -4 & 5 \end{bmatrix}$$

(i) Using Gaussian elimination, find matrices $\mathbf{U}$ and $\mathbf{D}$ such that

$$\mathbf{A} = \mathbf{U}^T\mathbf{D}\mathbf{U}$$

where $\mathbf{U}$ is an upper triangular matrix whose main diagonal consists of 1's only; and $\mathbf{D}$ is a diagonal matrix.

(ii) Does there exist a $m \times 3$ matrix $\mathbf{V}$ (where m is arbitrary) such that $\mathbf{V}^T\mathbf{V} = \mathbf{A}$?

***P* 2.37.** Consider the matrix

$$\mathbf{V} = \begin{bmatrix} 1 & 4 & -1 \\ -2 & 1 & 5 \\ 1 & -3 & 3 \\ 0 & 1 & 1 \end{bmatrix}$$

Find a 3×3 nonsingular matrix $\mathbf{B}$ such that

$$\mathbf{W} = \mathbf{V}\mathbf{B}^{-1}$$

is an orthogonal matrix (i.e., its columns are orthogonal and have nonzero norm).

***P* 2.38.** Consider the 5×3 matrix

$$\mathbf{V} = \left[\begin{array}{ccc} \mathbf{v}^{(1)} & \mathbf{v}^{(2)} & \mathbf{v}^{(3)} \end{array} \right] = \left[\begin{array}{rrr} 2 & 3 & 2 \\ 1 & 4 & 2 \\ 1 & 1 & -1 \\ 1 & 3 & 4 \\ 1 & 2 & -1 \end{array} \right]$$

(i) Express the inner product matrix $\mathbf{V}^T\mathbf{V}$ in the form $\mathbf{LDL}^T$.

(ii) Using $\mathbf{U} = \mathbf{L}^T$, obtain a 5×3 matrix $\mathbf{W}$ such that $\mathcal{R}(\mathbf{W}) = \mathcal{R}(\mathbf{V})$ and the columns of $\mathbf{W}$ are orthogonal.

(iii) Determine the projection of $\mathbf{s} = [\ -6 \quad -1 \quad 3 \quad -1 \quad 8\ ]^T$ on $\mathcal{R}(\mathbf{V})$ (which is the same as $\mathcal{R}(\mathbf{W})$).

***P* 2.39.** Consider a $n \times 3$ real-valued matrix

$$\mathbf{V} = \left[\begin{array}{ccc} \mathbf{v}^{(1)} & \mathbf{v}^{(2)} & \mathbf{v}^{(3)} \end{array} \right]$$

which is such that

$$\mathbf{V}^T\mathbf{V} = \left[\begin{array}{rrr} 16 & -8 & -12 \\ -8 & 8 & 8 \\ -12 & 8 & 11 \end{array} \right]$$

(i) Determine matrices $\mathbf{D}$ (diagonal with positive entries) and $\mathbf{U}$ (upper triangular with unit diagonal entries) such that

$$\mathbf{V}^T\mathbf{V} = \mathbf{U}^T\mathbf{D}\mathbf{U}$$

(ii) Determine coefficients c_{ij} that result in orthogonal vectors $\mathbf{w}^{(1)}$, $\mathbf{w}^{(2)}$ and $\mathbf{w}^{(3)}$, where:

$$\begin{aligned} \mathbf{w}^{(1)} &= \mathbf{v}^{(1)} \\ \mathbf{w}^{(2)} &= c_{12}\mathbf{v}^{(1)} + \mathbf{v}^{(2)} \\ \mathbf{w}^{(3)} &= c_{13}\mathbf{v}^{(1)} + c_{23}\mathbf{v}^{(2)} + \mathbf{v}^{(3)} \end{aligned}$$

***P* 2.40.** (MATLAB) Consider the discrete-time signals `v1`, `v2`, `v3` and `v4` defined by

```
t = ((0:0.01:1)*(pi/4))';
v1 = sin(t);
v2 = sin(2*t);
v3 = sin(3*t);
v4 = sin(4*t);
```

(i) Plot all four signals on the same graph.

(ii) Use the function `CHOL` to obtain four orthogonal signals `w1`, `w2`, `w3` and `w4` which are linearly equivalent to the above signals. Verify that the signals obtained are indeed orthogonal.

(iii) Plot `w1`, `w2`, `w3` and `w4` on the same graph.

***P* 2.41.** **(i)** Use MATLAB to verify that the six sinusoidal vectors in $\mathbf{R}^6$ defined below are orthogonal:

```
n = (0:5).';
r1 = cos(0*n);
r2 = cos(pi*n/3);
r3 = sin(pi*n/3);
r4 = cos(2*pi*n/3);
r5 = sin(2*pi*n/3);
r6 = cos(pi*n);
```

(ii) Verify that the vector $\mathbf{s} = [\ 0 \quad -3 \quad 1 \quad 0 \quad -1 \quad 3\ ]^T$ can be represented as a linear combination of `r3` and `r5`, i.e., in terms of sines only.

(iii) Without performing any additional computations, express

$$\mathbf{x} = [\ 3 \quad 0 \quad 4 \quad 3 \quad 2 \quad 6\ ]^T$$

as a linear combination of the six given sinusoidal vectors.

***P* 2.42.** Fourier sinusoids are not the only class of orthogonal reference vectors used in signal approximation/representation. In *wavelet* analysis, reference vectors reflect the frequency content of the signal which is being analyzed, as well as its “localized” behavior at different times. This is accomplished by taking as reference vectors *scaled* and *shifted* versions of a basic finite-duration pulse (or *wavelet*).

As a simple example, consider the problem of approximating a ($m = 256$)-dimensional signal $\mathbf{s}$ using $n = 64$ reference vectors derived from the basic *Haar* wavelet, which is a succession of two square pulses of opposite amplitudes. The 64 columns of the matrix $\mathbf{V}$ are constructed as follows:

- The first column is an all-ones vector, and provides *resolution* of order $r = 0$.

- The second column is a full-length Haar wavelet: 128 $+1$’s followed by 128 -1’s. This column provides resolution of order $r = 1$.

- The third column is a half-length Haar wavelet: a pulse of 64 $+1$'s followed by 64 -1's, followed by 128 zeros. In the fourth column, the pulse is shifted to the right by 128 positions so that it does not overlap the pulse in the third column. These two columns together provide resolution of order $r = 2$.
- We continue in this fashion to obtain resolutions of order up to $r = 6$. Resolution of order r is provided by columns $2^{r-1} + 1$ through 2^r of $\mathbf{V}$. Each of these columns contains a scaled and shifted Haar wavelet of duration 2^{9-r}. The number of columns at that resolution is the maximum possible under the constraint that pulses across columns do not overlap in time (row index).

The matrix $\mathbf{V}$ is generated in MATLAB using the following script (also found in `haar1.txt`):

```
p = 8;                  % vector size = m = 2^p
rmax = 6;               % max. resolution; also, n = 2^rmax
a = ones(2^p,1);
V = a;                  % first column (r=0) is all-ones
for r = 1:rmax
  v = [a(1:2^(p-r)); -a(1:2^(p-r)); zeros(2^p-2^(p-r+1),1)];
  for k = 0:2^(r-1)-1
    V = [V v(1+mod(-k*2^(p-r+1):-k*2^(p-r+1)+2^p-1, 2^p),:)];
  end
end
```

(i) Compute $\mathbf{V}^T\mathbf{V}$ and hence verify that the columns of $\mathbf{V}$ are orthogonal. You can view $\mathbf{V}$ using a ribbon plot, where each column (i.e., reference vector) is plotted as a separate ribbon:

```
    ribbon(V)
        % Maximize the graph window for a better view
    view(15,45)
    pause
    view(45,45)
```

(ii) Plot the following two signals:

```
    s1 = [exp((0:255)/256)]';
    s2 = [exp((0:115)/256)   -0.5+exp((116:255)/256)].';
```

(iii) Using the command

```
y = A*((A'*A)\(A'*s));
```

where `A` and `s` are chosen appropriately, determine and plot the least-squares approximations of `s1` and `s2` based on the entire matrix $\mathbf{V}$. Repeat, using columns 33–64 only, which provide the highest resolution ($r = 6$).

(Note that projecting `s1` on the highest-resolution columns results in a vector of small values compared to the amplitude of `s1`; this is because `s1` looks rather smooth at high resolution. On the other hand, `s2` has a sharp discontinuity which is captured in the highest-resolution projection. The order of the resolution can be increased to $r = 8$, in which case the matrix $\mathbf{V}$ has 256 columns and can provide a complete representation of any 256-point signal. At the highest resolution, each column of V consists of 254 zeros, one $+1$ and one -1.)

Section 2.13

***P* 2.43.** Consider the equation $\mathbf{Az} = \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} -1+j & -2-3j \\ 1+j & 3 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 4 \\ 2-j \end{bmatrix}$$

and $\mathbf{z} \in \mathbf{C}^{2\times 1}$.

(i) Find a real-valued 4×4 matrix $\mathbf{F}$ and a real-valued 4×1 vector $\mathbf{c}$ such that the above equation is equivalent to

$$\mathbf{F} \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix} = \mathbf{c}$$

where $z_1 = x_1 + jy_1$, $z_2 = x_2 + jy_2$.

(ii) Use MATLAB to verify that `A\b` and `F\c` are indeed equal.

Chapter 3

The Discrete Fourier Transform

3.1 Complex Sinusoids as Reference Vectors

3.1.1 Introduction

We have seen how a discrete-time signal consisting of finitely many samples can be represented as a vector, and how a linear transformation of that vector can be described by a matrix. In what follows, we will consider signal vectors $\mathbf{x}$ of length N, i.e., containing N samples or *points*. The time index n will range from $n = 0$ to $n = N - 1$. Thus

$$\mathbf{x} = \left[\begin{array}{cccc} x[0] & x[1] & \ldots & x[N-1] \end{array} \right]^T$$

Note the shift relative to our earlier indexing scheme: $n = 0, \ldots, N - 1$ instead of $n = 1, \ldots, N$.

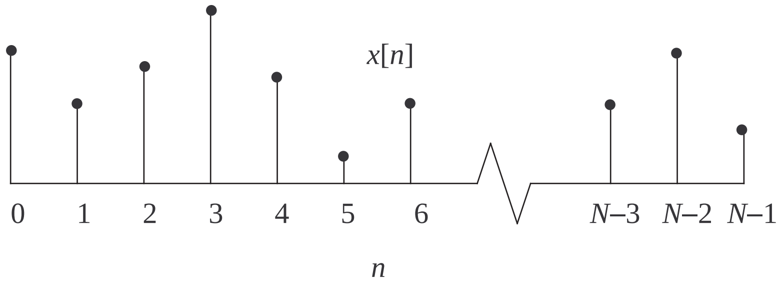


Figure 3.1: An N-point signal vector $\mathbf{x}$.

The sample values in $\mathbf{x}$ provide an immediate representation of the signal in terms of unit pulses at times $n = 0, \ldots, N - 1$:

$$\mathbf{x} = x[0]\mathbf{e}^{(0)} + x[1]\mathbf{e}^{(1)} + \cdots + x[N-1]\mathbf{e}^{(N-1)}$$

The most important feature of this representation is *localization in time*: each entry $x[n]$ of the vector $\mathbf{x}$ gives us the *exact* value of the signal at time n.

In signal analysis, we seek to *interpret* the information present in a signal. In particular, we are interested in how signal values at different times depend on each other. In general, this sort of information cannot be obtained by inspection of the vector $\mathbf{x}$. For example, if the samples in $\mathbf{x}$ follow a pattern closely described by a linear, exponential or sinusoidal function of time, that pattern cannot be easily discerned by looking at individual numbers in $\mathbf{x}$.

And while a trained eye could detect such a simple pattern by looking at a plot of **x**, it could miss a more complicated pattern such as a sum of five sinusoids of different frequencies.

Fourier analysis is the most common approach to signal analysis, and is based on representing or approximating a signal by a linear combination of sinusoids. The importance of sinusoids is due to (among other factors):

- their immunity to distortion when propagating through a linear medium, or when processed by a linear filter;
- our perception of real-world signals, e.g., our ability to detect and separate different frequencies in audio signals, as well as our ability to group harmonics into a single note.

In Fourier analysis, we represent **x** by a vector **X** of the same length, where each entry in **X** provides the (complex) amplitude of a complex sinusoid. The two representations are equivalent. In contrast to the vector **x** itself, whose values provide localization in time, the vector **X** provides *localization in frequency.*

3.1.2 Fourier Sinusoids

A *standard* complex sinusoid **x** of length N and frequency ω is given by

$$x[n] = e^{j\omega n} \ , \qquad n = 0, \dots, N-1$$

The designation "standard" means that the complex-valued amplitude is unity—equivalently, the real-valued amplitude is unity and the phase shift is zero. Note that regardless of the value of ω,

$$x[0] = 1$$

Recall from Subsection 1.5.3 that frequencies ω_1 and ω_2 related by

$$\omega_2 = \omega_1 + 2k\pi \qquad (k \in \mathbf{Z})$$

yield the same values for **x** and are thus equivalent. This enables us to limit the *effective* frequency range to a single interval of angles of length 2π. In what follows, we will usually assume that

$$0 \le \omega < 2\pi$$

If we extend the discrete time axis to $\mathbf{Z}$ (i.e., $n = -\infty$ to $n = \infty$), then

$$x'[n] = e^{j\omega n} \ , \qquad n \in \mathbf{Z}$$

defines a complex sinusoid having infinite duration. The sinusoid $x'[\cdot]$ will repeat itself every N samples provided

$$(\forall n) \qquad e^{j\omega(n+N)} = e^{j\omega n}$$

which reduces to

$$e^{j\omega N} = 1$$

and thus

$$\omega = k \cdot \frac{2\pi}{N}$$

for some integer k. (This expression was also obtained in Subsection 1.5.4.) Since the effective range of ω is $(0, 2\pi]$, it follows that k can be limited to the integers $0, \ldots, N-1$. Thus there are exactly N distinct frequencies ω for which $x'[n] = e^{j\omega n}$ is periodic with (fundamental) period equal to N or a submultiple of N. These frequencies are given by

$$\omega = 0, \frac{2\pi}{N}, \frac{4\pi}{N}, \ldots, \frac{(N-1)2\pi}{N}$$

Definition 3.1.1. The Fourier frequencies for an N-point vector are given by

$$\omega = k \cdot \frac{2\pi}{N}$$

where $k = 0, \ldots, N-1$. The corresponding Fourier sinusoids $\mathbf{v}^{(0)}, \mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(N-1)}$ are the N-point vectors given by

$$v^{(k)}[n] = e^{j(2\pi/N)kn}\ , \qquad n = 0, \ldots, N-1 \qquad \square$$

We note the following:

- All Fourier frequencies are multiples of $2\pi/N$.

- $\omega = 0$ (which results in a constant signal of value $+1$) is always a Fourier frequency, corresponding to $k = 0$ in the above definition.

- $\omega = \pi$ (which results in a signal whose value alternates between $+1$ and -1) is a Fourier frequency only when N is even.

- If $\omega \in (0, \pi)$ is a Fourier frequency, so is $2\pi - \omega \in (\pi, 2\pi)$. Put differently, if ω is the k^{th} Fourier frequency, then $2\pi - \omega$ is the $(N-k)^{\text{th}}$ Fourier frequency. The corresponding Fourier sinusoids are complex conjugates of each other:

$$e^{j(2\pi-\omega)n} = e^{-j\omega n} = (e^{j\omega n})^*$$

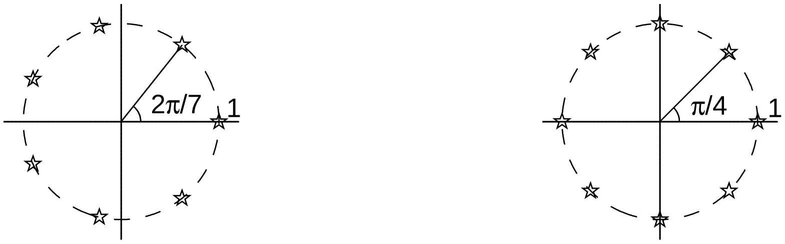


Figure 3.2: Fourier frequencies for $N = 7$ (left) and $N = 8$ (right), represented by asterisks on the unit circle.

The Fourier frequencies ω for $N = 7$ (odd) and $N = 8$ (even) are illustrated in Figure 3.2.

For the remainder of this chapter, Fourier sinusoids of length N will be used as reference vectors for signals in $\mathbf{R}^{N\times 1}$, and more generally, $\mathbf{C}^{N\times 1}$. The notation $\mathbf{v}^{(k)}$ for the k^{th} Fourier sinusoid is consistent with our earlier notation for reference vectors. In particular, we can arrange the Fourier sinusoids as columns of a $N \times N$ matrix $\mathbf{V}$:

$$\mathbf{V} = \left[\begin{array}{cccc} \mathbf{v}^{(0)} & \mathbf{v}^{(1)} & \ldots & \mathbf{v}^{(N-1)} \end{array} \right]$$

For $\mathbf{V}$, the row index $n = 0, \ldots, N-1$ represents time, while the column index $k = 0, \ldots, N-1$ represents frequency. The $(n, k)^{\text{th}}$ entry of $\mathbf{V}$ is given by

$$V_{nk} = v^{(k)}[n] = e^{j(2\pi/N)kn} = \cos\left(\frac{2\pi kn}{N}\right) + j\sin\left(\frac{2\pi kn}{N}\right)$$

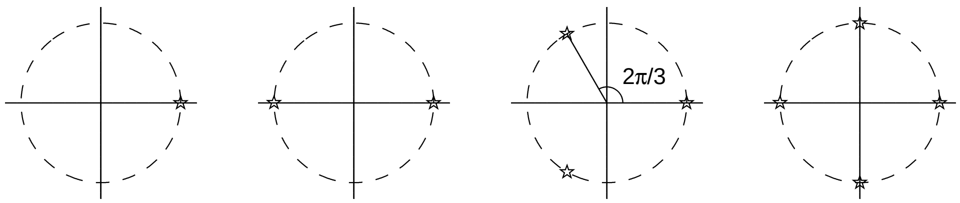
Note in particular that $V_{nk} = V_{kn}$, i.e., $\mathbf{V}$ is *always symmetric.*

The Fourier frequencies and matrices $\mathbf{V}$ for $N = 1, 2, 3$ and 4 are shown in Figure 3.3. Note that the $n = 0^{\text{th}}$ row and $k = 0^{\text{th}}$ column equal unity for all N. The k^{th} column is generated by moving counterclockwise on the unit circle in increments of $\omega = k(2\pi/N)$, starting from $z = 1$.

3.1.3 Orthogonality of Fourier Sinusoids

Fourier sinusoids of the same length N are orthogonal. As a result, they are well-suited as reference vectors. (Recall that the the least squares approximation of a signal by a set of reference vectors is the vector sum of the projections of the signal on each reference vector.)

Proof of the orthogonality of Fourier sinusoids involves the summation of geometric series, which we review below.



$$\begin{bmatrix} 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2}+j\frac{\sqrt{3}}{2} & -\frac{1}{2}-j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2}-j\frac{\sqrt{3}}{2} & -\frac{1}{2}+j\frac{\sqrt{3}}{2} \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

Figure 3.3: Fourier sinusoids for $N = 1, 2, 3$ and 4. The entries of each matrix are marked on the corresponding unit circle.

Fact. *If z is complex, then*

$$G_n(z) \overset{\text{def}}{=} 1 + z + z^2 + \cdots + z^n = \begin{cases} (1 - z^{n+1})/(1 - z), & z \neq 1; \\ n + 1, & z = 1. \end{cases}$$

The limit of $G_n(z)$ as $n \to \infty$ exists if and only $|z| < 1$, and is given by

$$G_\infty(z) = \frac{1}{1 - z}$$

The above results are from elementary calculus, and are easily obtained by noting that:

- If $z = 1$, then all terms in the geometric series are equal to 1.
- $G_n(z) - zG_n(z) = 1 - z^{n+1}$.
- If $|z| > 1$, then $|z|^{n+1} = |z^{n+1}|$ diverges to infinity.
- If $|z| = 1$ and $z \neq 1$, then z^{n+1} constantly rotates on the unit circle (hence it does not converge).
- If $|z| < 1$, then $|z|^{n+1}$ converges to zero. □

To establish the orthogonality of the N-point Fourier sinusoids $\mathbf{v}^{(k)}$ and $\mathbf{v}^{(\ell)}$ having frequencies $\omega = k(2\pi/N)$ and $\omega = \ell(2\pi/N)$, respectively, we need to compute the inner product of $\mathbf{v}^{(k)}$ and $\mathbf{v}^{(\ell)}$:

$$\begin{aligned}\langle \mathbf{v}^{(k)}, \mathbf{v}^{(\ell)} \rangle &= \sum_{n=0}^{N-1} \left(e^{j(2\pi/N)kn}\right)^* e^{j(2\pi/N)\ell n} \\ &= \sum_{n=0}^{N-1} e^{j(2\pi/N)(\ell-k)n} \\ &= \sum_{n=0}^{N-1} z^n \\ &= G_{N-1}(z)\end{aligned}$$

where

$$z = e^{j(2\pi/N)(\ell-k)}$$

If $k = \ell$, then $z = 1$ and

$$\langle \mathbf{v}^{(k)}, \mathbf{v}^{(k)} \rangle = \|\mathbf{v}^{(k)}\|^2 = N$$

(Note that each entry of $\mathbf{v}^{(k)}$ is on the unit circle, hence its squared magnitude equals unity.) Of course, this is also consistent with $G_{N-1}(1) = N$.

If $0 \leq k \neq \ell \leq N-1$, then $\ell - k$ cannot be a multiple of N, and thus $z \neq 1$. Using the formula for $G_{N-1}(z)$, we have

$$\langle \mathbf{v}^{(k)}, \mathbf{v}^{(\ell)} \rangle = \frac{1 - z^N}{1 - z}$$

But

$$z^N = e^{j(2\pi/N)(\ell-k)N} = e^{j2\pi(\ell-k)} = 1$$

and hence

$$\langle \mathbf{v}^{(k)}, \mathbf{v}^{(\ell)} \rangle = 0$$

as needed. We have thus obtained the following result.

Fact. *The N-point complex Fourier sinusoids comprising the columns of the matrix*

$$\mathbf{V} = \left[\begin{array}{cccc} \mathbf{v}^{(0)} & \mathbf{v}^{(1)} & \ldots & \mathbf{v}^{(N-1)} \end{array}\right]$$

are orthogonal, each having squared norm equal to N. □

In terms of the inner product matrix $\mathbf{V}^H\mathbf{V}$, we have

$$\mathbf{V}^H\mathbf{V} = N\mathbf{I}$$

Since $\mathbf{V}$ is a square matrix, the above relationship tells us that it is also nonsingular and has inverse

$$\mathbf{V}^{-1} = \frac{1}{N}\mathbf{V}^H$$

It follows that the N complex Fourier sinusoids $\mathbf{v}^{(0)}, \mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(N-1)}$ are also linearly independent.

In the context of signal approximation and representation, the above facts imply the following:

- The least-squares approximation of any real or complex-valued signal $\mathbf{s}$ based on any subset of the above N complex Fourier sinusoids is given by the sum of the projections of $\mathbf{s}$ onto each of the sinusoids; where the projection of $\mathbf{s}$ onto $\mathbf{v}^{(k)}$ is given by the standard expression

$$\frac{\langle \mathbf{v}^{(k)}, \mathbf{s}\rangle}{N}\mathbf{v}^{(k)}$$

- The least-squares approximation of $\mathbf{s}$ based on all N complex Fourier sinusoids is an exact representation of $\mathbf{s}$, and is given by

$$\mathbf{s} = \mathbf{V}\mathbf{c} \qquad \Leftrightarrow \qquad \mathbf{c} = \mathbf{V}^{-1}\mathbf{s} = \frac{1}{N}\mathbf{V}^H\mathbf{s}$$

Example 3.1.1. Let $N = 3$, and consider the signals

$$\mathbf{s} = [2 \quad -1 \quad -1]^T \qquad \text{and} \qquad \mathbf{x} = [2 \quad -1 \quad 0]^T$$

The Fourier exponentials are given by the columns of

$$\mathbf{V} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} + j\frac{\sqrt{3}}{2} & -\frac{1}{2} - j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} - j\frac{\sqrt{3}}{2} & -\frac{1}{2} + j\frac{\sqrt{3}}{2} \end{bmatrix}$$

We have

$$\mathbf{s} = \mathbf{V}\mathbf{c} \qquad \text{and} \qquad \mathbf{x} = \mathbf{V}\mathbf{d}$$

where

$$\begin{aligned} c_0 &= \frac{1}{3}\langle \mathbf{v}^{(0)}, \mathbf{s}\rangle = 0 \\ c_1 &= \frac{1}{3}\langle \mathbf{v}^{(1)}, \mathbf{s}\rangle = 1 \\ c_2 &= \frac{1}{3}\langle \mathbf{v}^{(2)}, \mathbf{s}\rangle = 1 \end{aligned}$$

and

$$\begin{aligned} d_0 &= \frac{1}{3}\langle \mathbf{v}^{(0)}, \mathbf{x}\rangle = \frac{1}{3} \\ d_1 &= \frac{1}{3}\langle \mathbf{v}^{(1)}, \mathbf{x}\rangle = \frac{5}{6} + j\frac{\sqrt{3}}{6} \\ d_2 &= \frac{1}{3}\langle \mathbf{v}^{(2)}, \mathbf{x}\rangle = \frac{5}{6} - j\frac{\sqrt{3}}{6} \end{aligned}$$

□

3.2 The Discrete Fourier Transform and its Inverse

3.2.1 Definitions

We begin our in-depth discussion of the representation of a N-point signal vector $\mathbf{s}$ by the Fourier sinusoids in $\mathbf{V}$ with a simple change of notation: we replace the coefficient vector $\mathbf{c}$ in the equations

$$\mathbf{s} = \mathbf{V}\mathbf{c} \qquad \Leftrightarrow \qquad \mathbf{c} = \frac{1}{N}\mathbf{V}^H\mathbf{s}$$

by $\mathbf{S}/N$, i.e., we define $\mathbf{S} = N\mathbf{c}$. We thus obtain

$$\mathbf{s} = \frac{1}{N}\mathbf{V}\mathbf{S} \qquad \Leftrightarrow \qquad \mathbf{S} = \mathbf{V}^H\mathbf{s}$$

Recall that the matrix $\mathbf{V}$ is symmetric:

$$V_{nk} = e^{j(2\pi/N)kn} = V_{kn}$$

Thus

$$\mathbf{V}^H = (\mathbf{V}^T)^* = \mathbf{V}^*$$

and

$$(\mathbf{V}^H)_{nk} = e^{-j(2\pi/N)kn}$$

Definition 3.2.1. The *discrete Fourier transform* (DFT) of the N-point vector $\mathbf{s}$ is the N-point vector $\mathbf{S}$ defined by

$$\mathbf{S} = \mathbf{V}^H\mathbf{s}$$

or equivalently,

$$S[k] = \sum_{n=0}^{N-1} s[n]e^{-j(2\pi/N)kn} \ , \qquad k = 0, \dots, N-1$$

$\mathbf{S}$ will also be referred to as the (complex) *spectrum* of $\mathbf{s}$. □

For the remainder of this chapter, the upper-case boldface symbols $\mathbf{S}$*,* $\mathbf{X}$ *and* $\mathbf{Y}$ *will be reserved for the discrete Fourier transforms of the signal vectors* $\mathbf{s}$*,* $\mathbf{x}$ *and* $\mathbf{y}$*, respectively.*

The equations in the definition of the DFT are also known as the *analysis* equations. They yield the coefficients needed in order to express the signal $\mathbf{s}$ as a sum of Fourier sinusoids. Since these sinusoids are the components of $\mathbf{s}$, computation of the coefficients $S[0], \dots, S[N-1]$ amounts to analyzing the signal into N distinct frequencies.

Definition 3.2.2. The *inverse discrete Fourier transform* (IDFT) of the N-point vector $\mathbf{X}$ is the N-point vector $\mathbf{x}$ defined by

$$\mathbf{x} = \frac{1}{N}\mathbf{V}\mathbf{X}$$

or equivalently,

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j(2\pi/N)kn} \ , \qquad n = 0, \ldots, N-1 \qquad \square$$

The equations in the definition of the IDFT are known as the *synthesis* equations, since they produce a signal vector $\mathbf{x}$ by summing together complex sinusoids with (complex) amplitudes given by the entries of $(1/N)\mathbf{X}$. Clearly,

- if $\mathbf{X}$ is the DFT of $\mathbf{x}$, then $\mathbf{x}$ is the IDFT of $\mathbf{X}$; and

- the analysis and synthesis equations are equivalent.

A signal $\mathbf{x}$ and its DFT $\mathbf{X}$ form a *DFT pair*, denoted by

$$\mathbf{x} \overset{\text{DFT}}{\longleftrightarrow} \mathbf{X} \qquad \text{or simply} \qquad \mathbf{x} \longleftrightarrow \mathbf{X}$$

Throughout our discussion, the lower-case symbol $\mathbf{x}$ will denote a signal in the *time domain*, i.e., evolving in time n. The DFT (or spectrum) $\mathbf{X}$ will be regarded as a signal in the *frequency domain*, since its values are indexed by the frequency parameter k (corresponding to a radian frequency $\omega = k(2\pi/N)$).

The two signals $\mathbf{x}$ and $\mathbf{X}$ have the same physical dimensions and units. Either signal can be assigned arbitrary values in $\mathbf{C}^{N\times 1}$; such assignment automatically determines the value of the other signal (through the synthesis or analysis equations). Thus any signal in the time domain is also perfectly valid as a signal in the frequency domain (i.e., as the spectrum of a *different* time domain signal), and vice versa. Even though time and frequency have a distinctly different *physical* meaning, the two entities are treated very similarly in the discrete Fourier transform and its inverse. This similarity is quite evident in the analysis and synthesis equations, which differ only in terms of a complex conjugate and a scaling factor.

3.2.2 Symmetry Properties of the DFT and IDFT Matrices

Recall that the $N \times N$ matrix

$$\mathbf{V} = \left[\begin{array}{cccc} \mathbf{v}^{(0)} & \mathbf{v}^{(1)} & \ldots & \mathbf{v}^{(N-1)} \end{array} \right]$$

is given by

$$V_{nk} = e^{j(2\pi/N)kn}$$

where the row index $n = 0, \ldots, N-1$ corresponds to time and the column index $k = 0, \ldots, N-1$ corresponds to frequency. $\mathbf{V}$ will be referred to as the *IDFT matrix* for a N-point vector, since it appears (without conjugation) in the synthesis equation. Whenever the value of N is not clear from the context, we will use $\mathbf{V}_N$ instead of $\mathbf{V}$.

Letting

$$v = v_N \overset{\text{def}}{=} e^{j(2\pi/N)} = \cos\left(\frac{2\pi}{N}\right) + j\sin\left(\frac{2\pi}{N}\right)$$

we obtain the simple expression

$$V_{nk} = v^{kn}$$

and can thus write

$$\mathbf{V} = \left[\begin{array}{ccccc} 1 & 1 & 1 & \ldots & 1 \\ 1 & v & v^2 & \ldots & v^{N-1} \\ 1 & v^2 & v^4 & \ldots & v^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & v^{N-1} & v^{2(N-1)} & \ldots & v^{(N-1)^2} \end{array} \right]$$

As noted earlier, the entries of $\mathbf{V}$ are symmetric about the main diagonal since $V_{nk} = v^{kn} = v^{nk}$. For other structural properties of $\mathbf{V}$, we turn to the $k = 1^{\text{st}}$ column

$$\mathbf{v}^{(1)} = \left[\begin{array}{ccccc} 1 & v & v^2 & \ldots & v^{N-1} \end{array} \right]^T$$

and note that the elements of that column represent equally spaced points on the unit circle. This is illustrated in Figure 3.4.

Note that

$$v^N = e^{j(2\pi/N)N} = e^{j2\pi} = 1$$

and thus for any integer r,

$$v^{n+rN} = v^n$$

Every integer power of v thus equals one of $1, v, \ldots, v^{N-1}$, which means that *all* columns of $\mathbf{V}$ can be formed using entries taken from the $k = 1^{\text{st}}$ column $\mathbf{v}^{(1)}$.

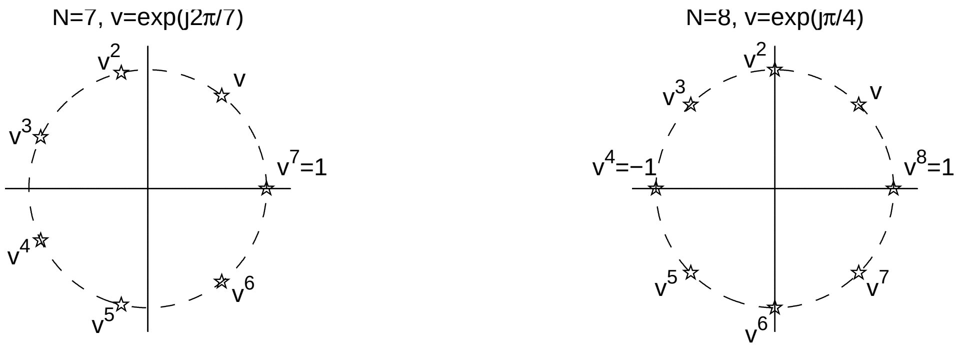


Figure 3.4: The entries of the first Fourier sinusoid marked on the unit circle for $N = 7$ (left) and $N = 8$ (right).

Example 3.2.1. For $N = 6$, the entries of $\mathbf{V} = \mathbf{V}_6$ can be expressed in terms of $1, v, v^2, v^3, v^4$ and v^5, where $v = e^{j\pi/3}$. Note that $v^3 = e^{j\pi} = -1$. We have

$$\mathbf{V} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & e^{j\pi/3} & e^{j2\pi/3} & -1 & e^{j4\pi/3} & e^{j5\pi/3} \\ 1 & e^{j2\pi/3} & e^{j4\pi/3} & 1 & e^{j2\pi/3} & e^{j4\pi/3} \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & e^{j4\pi/3} & e^{j2\pi/3} & 1 & e^{j4\pi/3} & e^{j2\pi/3} \\ 1 & e^{j5\pi/3} & e^{j4\pi/3} & -1 & e^{j2\pi/3} & e^{j\pi/3} \end{bmatrix}$$

or equivalently, (since $e^{j(2\pi-\theta)} = e^{-j\theta}$):

$$\mathbf{V} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & e^{j\pi/3} & e^{j2\pi/3} & -1 & e^{-j2\pi/3} & e^{-j\pi/3} \\ 1 & e^{j2\pi/3} & e^{-j2\pi/3} & 1 & e^{j2\pi/3} & e^{-2j\pi/3} \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & e^{-j2\pi/3} & e^{j2\pi/3} & 1 & e^{-j2\pi/3} & e^{j2\pi/3} \\ 1 & e^{-j\pi/3} & e^{-j2\pi/3} & -1 & e^{j2\pi/3} & e^{j\pi/3} \end{bmatrix} \qquad \square$$

The symmetric nature of $\mathbf{V} = \mathbf{V}_6$ is evident in Example 3.2.1. We note that in the general case (arbitrary N), the n^{th} and $(N-n)^{\text{th}}$ entries of $\mathbf{v}^{(k)}$ are complex conjugates of each other:

$$v^{k(N-n)} = v^{-kn} = (v^{kn})^*$$

This means that the elements of the $k = 1^{\text{st}}$ column exhibit a form of *conjugate symmetry.* The center of symmetry is the row index $N/2$, which is at equal distance from n and $N-n$:

$$n = \frac{N}{2} - \left(\frac{N}{2} - n\right) \qquad \text{and} \qquad N - n = \frac{N}{2} + \left(\frac{N}{2} - n\right)$$

The value $N/2$ is an actual row index if N is even, and is midway between two row indices if N is odd. We note also that the $n = 0^{\text{th}}$ entry is *not* part of a conjugate symmetric pair, since the highest row index is $n = N-1$, not $n = N$.

Clearly, the same conjugate symmetry arises in the *columns* of $\mathbf{V}$, with center of symmetry given by the column index $k = N/2$. This follows easily from the symmetry of $\mathbf{V}$ about its diagonal, i.e., from $\mathbf{V} = \mathbf{V}^T$.

We finally note that since

$$v^{(N-k)(N-n)} = v^{N^2-kN-nN+kn} = v^{kn}$$

we also have *radial symmetry* (without conjugation) about the point $(n,k) = (N/2, N/2)$:

$$V_{N-n,N-k} = V_{nk}$$

In particular, radial symmetry implies that entries $n = 1, \ldots, N-1$ in the $(N-k)^{\text{th}}$ column can be obtained by reversing the order of the same entries in the k^{th} column.

The symmetry properties of $\mathbf{V}$ are summarized in Figure 3.5.

Definition 3.2.3. The *DFT matrix* $\mathbf{W} = \mathbf{W}_N$ for a N-point vector is defined by

$$\mathbf{W} = \mathbf{V}^H = \mathbf{V}^*$$

Specifically, $(k,n)^{\text{th}}$ entry of $\mathbf{W}$ is given by

$$W_{kn} = w^{kn}$$

where

$$w = w_N \overset{\text{def}}{=} v_N^{-1} = e^{-j2\pi/N} \qquad \square$$

$\mathbf{W}$ is known as the DFT matrix because it appears without conjugation in the analysis equation:

$$\mathbf{X} = \mathbf{V}^H\mathbf{x} = \mathbf{W}\mathbf{x}$$

Note that in the case of $\mathbf{W}$, the frequency index k is the row index, while the time index n is the column index. This is also consistent with the indexing scheme used in the sum form of the analysis equation:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j(2\pi/N)kn}$$

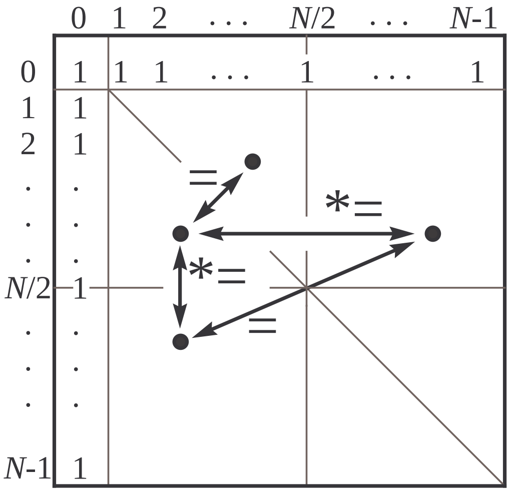


Figure 3.5: Symmetry properties of the IDFT matrix **V**. Equal values are denoted by "=" and conjugate values by "* =".

We note the following:

- The introduction of **W** allows us to write the equation

$$\mathbf{V}^H\mathbf{V} = \mathbf{V}\mathbf{V}^H = N\mathbf{I}$$

  in a variety of equivalent forms, using

$$\mathbf{V}^H = \mathbf{V}^* = \mathbf{W} \qquad \text{and} \qquad \mathbf{V} = \mathbf{W}^H = \mathbf{W}^*$$

- Since **W** is just the complex conjugate of **V**, it has the same symmetry properties as **V**.

3.2.3 Signal Structure and the Discrete Fourier Transform

The structure of a signal determines the structure of its spectrum. In order to understand and apply Fourier analysis, one needs to know how certain key features of a signal in the time domain are reflected in its spectrum in the frequency domain.

A fundamental property of the discrete Fourier transform is its linearity, which is due to the matrix-vector form of the DFT and IDFT transformations.

DFT 1. *(Linearity) If* $\mathbf{x} \longleftrightarrow \mathbf{X}$ *and* $\mathbf{y} \longleftrightarrow \mathbf{Y}$*, then for any real or complex scalars* α *and* β*,*

$$\mathbf{s} = \alpha\mathbf{x} + \beta\mathbf{y} \quad \longleftrightarrow \quad \mathbf{S} = \alpha\mathbf{X} + \beta\mathbf{Y}$$

This can be proved using either the analysis or the synthesis equation. Using the former,

$$\begin{aligned} \mathbf{S} &= \mathbf{Ws} \\ &= \mathbf{W}(\alpha\mathbf{x} + \beta\mathbf{y}) \\ &= \alpha\mathbf{Wx} + \beta\mathbf{Wy} \\ &= \alpha\mathbf{X} + \beta\mathbf{Y} \end{aligned}$$

as needed. □

Another fundamental property of the Fourier transform is the *duality* of the DFT and IDFT transformations. Duality stems from the fact that the two transformations are obtained using matrices $\mathbf{W} = \mathbf{V}^*$ and $(1/N)\mathbf{V}$ which differ only by a scaling factor and a complex conjugate. As a result, if $\mathbf{x} \longleftrightarrow \mathbf{X}$ is a DFT pair, then the *time-domain* signal $\mathbf{y} = \mathbf{X}$ has a spectrum $\mathbf{Y}$ whose structure is very similar to that of the original time-domain signal $\mathbf{x}$. The precise statement of the duality property will be given in Section 3.4.

Before continuing with our systematic development of DFT properties, we consider four simple signals and compute their spectra.

Example 3.2.2. Let $\mathbf{x}$ be the 0th unit vector $\mathbf{e}^{(0)}$, namely a single unit pulse at time $n = 0$:

$$\mathbf{x} = \mathbf{e}^{(0)} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix}^T$$

The DFT $\mathbf{X}$ is given by

$$\mathbf{X} = \mathbf{We}^{(0)}$$

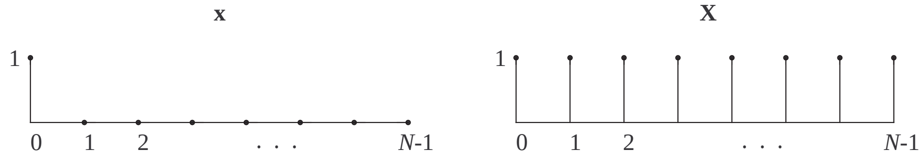
Right-multiplying $\mathbf{W}$ by a unit vector amounts to column selection. In this case, the 0th column of $\mathbf{W}$ is selected:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \end{bmatrix}^T = \mathbf{1}$$

i.e., $\mathbf{X}$ is the all-ones vector. The same result is obtained using the analysis equation in its sum form:

$$X[k] = \sum_{n=0}^{N-1} x[n]w^{kn} = 1 \cdot w^0 = 1$$

for all $k = 0, \dots, N-1$. □



Example 3.2.2

Example 3.2.3. We now delay the unit pulse by taking $\mathbf{x} = \mathbf{e}^{(m)}$, or equivalently,

$$x[n] = \begin{cases} 1, & n = m; \\ 0, & n \neq m, \end{cases}$$

where $0 \leq m \leq N-1$. The DFT becomes

$$\mathbf{X} = \mathbf{W}\mathbf{e}^{(m)}$$

i.e., $\mathbf{X}$ is the m^{th} column of $\mathbf{W}$, which is the same as the complex conjugate of $\mathbf{v}^{(m)}$ (the m^{th} column of $\mathbf{V}$). Thus $\mathbf{X}$ is a Fourier exponential of frequency $(N-m)(2\pi/N)$, or equivalently, $-m(2\pi/N)$:

$$X[k] = w^{km} = v^{-km} = e^{-j(2\pi/N)km}$$

Of course, the same result is obtained using the analysis equation in its sum form:

$$X[k] = \sum_{n=0}^{N-1} x[n]w^{kn} = 1 \cdot w^{km}$$

Note that in this example, the spectrum is complex-valued. It is purely real-valued (for all k) if and only if $m = 0$ (i.e., there is no delay) or $m = N/2$. The latter value of m is an actual time index only when the number of samples is even, in which case the resulting spectrum is given by $X[k] = (-1)^k$. □

Example 3.2.4. Building upon Example 3.2.3, we add a second pulse of unit height at time $n = N - m$ (where $m \neq 0$). The two pulses are now symmetric to each other relative to the time instant $n = N/2$. Denoting the new signal by $\mathbf{s}$, we have

$$\mathbf{s} = \mathbf{e}^{(m)} + \mathbf{e}^{(N-m)}$$

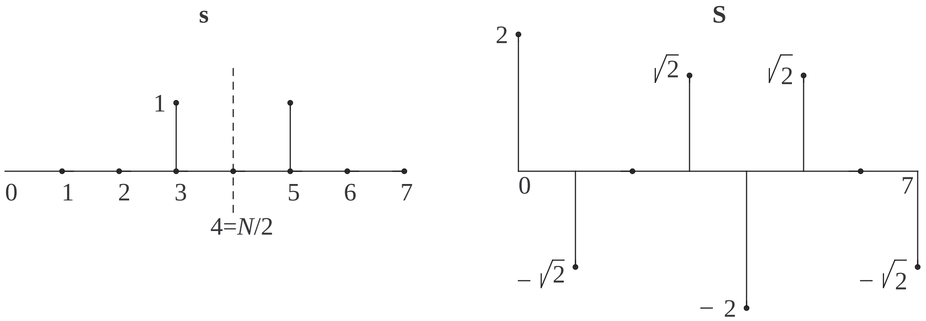
and thus by linearity of the DFT we obtain

$$\mathbf{S} = \mathbf{W}\mathbf{e}^{(m)} + \mathbf{W}\mathbf{e}^{(N-m)}$$

This is the sum of the DFT's obtained in Example 3.2.3 for delays m and $N - m$. In particular, we have

$$\begin{aligned} S[k] &= e^{-j(2\pi/N)km} + e^{-j(2\pi/N)k(N-m)} \\ &= e^{-j(2\pi/N)km} + e^{j(2\pi/N)km} \\ &= 2\cos\left(\frac{2\pi mk}{N}\right) \end{aligned}$$

The spectrum is now purely real-valued. The figure illustrates the case $N = 8$ and $m = 3$. □



Example 3.2.4 (with $N = 8$ and $m = 3$)

Example 3.2.5. Finally, we introduce unit pulses at every time instant, resulting in the constant signal $\mathbf{y} = \mathbf{1}$. The analysis equation gives

$$\mathbf{Y} = \mathbf{W1}$$

i.e., $\mathbf{Y}$ is the (componentwise) sum of all columns of $\mathbf{W}$. The sum form of the equation gives

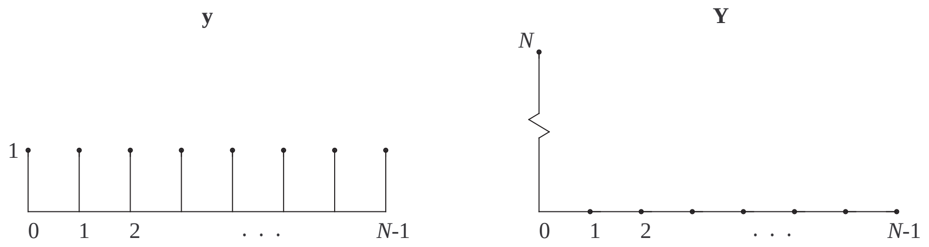
$$Y[k] = \sum_{n=0}^{N-1} 1 \cdot w^{kn}$$

which is the same as the geometric sum $G_{N-1}(w^k)$. It is also the inner product $\langle \mathbf{v}^{(k)}, \mathbf{v}^{(0)} \rangle$, which equals zero if $k \neq 0$ and N if $k = 0$. The resulting spectrum is

$$\mathbf{Y} = N\mathbf{e}^{(0)}$$

For a simpler way of deriving $\mathbf{Y}$, note that

$$\mathbf{y} = \mathbf{1} = \mathbf{v}^{(0)}$$



Example 3.2.5

i.e., $\mathbf{y}$ consists of a single Fourier sinusoid of frequency zero (corresponding to $k = 0$) and unit amplitude. This means that in the synthesis equation

$$\mathbf{y} = \frac{1}{N}\mathbf{V}\mathbf{Y} \ ,$$

the DFT vector $\mathbf{Y}$ selects the $k = 0^{\text{th}}$ column of $\mathbf{V}$ and multiplies it by N. Hence $\mathbf{Y} = N\mathbf{e}^{(0)}$. □

The graphs for Examples 3.2.2 and 3.2.5 are similar: essentially, time and frequency domains are interchanged with one of the signals undergoing scaling by N. This similarity is a consequence of the duality property of the DFT, which will be discussed formally in Section 3.4.

3.3 Structural Properties of the DFT: Part I

3.3.1 The Spectrum of a Real-Valued Signal

As we saw in the previous section, any complex-valued vector can be viewed as either

- a time-domain signal; or
- a frequency-domain signal, i.e., the DFT (or spectrum) of a time-domain signal.

Of course, all time-domain signals encountered in practice are real-valued; the generalization to complex-valued signals is necessary in order to include complex sinusoids (and, as we shall see later, complex exponentials) in our analysis. A natural question to ask is whether the spectrum $\mathbf{X}$ of a *real-valued* signal $\mathbf{x}$ has special properties that distinguish it from the spectrum of an *arbitrary complex-valued* signal in the time domain. The answer is affirmative, and is summarized below.

DFT 2. *(DFT of a Real-Valued Signal) If $\mathbf{x}$ is a real-valued N-point signal, then $\mathbf{X}$ satisfies*

$$X[0] = X^*[0]$$

and

$$X[k] = X^*[N-k]$$

for $k = 1, \ldots, N-1$.

To prove this property, we note first that

$$X[0] = \sum_{n=0}^{N-1} x[n]w^{0\cdot n} = \sum_{n=0}^{N-1} x[n]$$

which is real-valued since $\mathbf{x}$ has real-valued entries. For $k = 1, \ldots, N-1$, we have

$$X[k] = \sum_{n=0}^{N-1} x[n]w^{kn}$$

and

$$X[N-k] = \sum_{n=0}^{N-1} x[n]w^{(N-k)n} = \sum_{n=0}^{N-1} x[n]w^{-kn}$$

Taking complex conjugates across the second equation, we obtain

$$\begin{aligned} X^*[N-k] &= \left(\sum_{n=0}^{N-1} x[n]w^{-kn}\right)^* \\ &= \sum_{n=0}^{N-1} \left(x[n]w^{-kn}\right)^* \\ &= \sum_{n=0}^{N-1} x^*[n]w^{kn} \end{aligned}$$

Since $\mathbf{x}$ is real valued, we have $x^*[\cdot] = x[\cdot]$ and therefore

$$X[k] = X^*[N-k]$$

as needed. □

DFT 2 tells us that the DFT (or spectrum) of a real-valued signal exhibits the same kind of conjugate symmetry as was seen in the rows and columns of $\mathbf{W}$ and $\mathbf{V}$. This symmetry will be explored further in this chapter.

Expressing $X[k]$ in polar form, i.e.,

$$X[k] = |X[k]| \cdot e^{j\angle X[k]}$$

we obtain two new frequency-domain vectors indexed by $k = 0, \dots, N-1$. These are:

- The *amplitude* spectrum, given by $|X[\cdot]|$. If $\mathbf{x}$ is real-valued, then DFT 2 implies that

  $$|X[k]| = |X^*[N-k]|, \qquad k = 1, \dots, N-1$$

  i.e., the amplitude spectrum is *symmetric* in k with center of symmetry at $k = N/2$.

- The *phase* spectrum, given by the angle $\angle X[\cdot]$ *quoted in the interval* $[-\pi, \pi]$. If $\mathbf{x}$ is real-valued, then DFT 2 implies that

  $$\angle X[0] = 0 \text{ or } \pm\pi$$

  and

  $$\angle X[k] = -\angle X[N-k], \qquad k = 1, \dots, N-1$$

  i.e., the phase spectrum is *antisymmetric* in k.

Example 3.3.1. In Example 3.1.1, we considered the vector

$$\mathbf{x} = \begin{bmatrix} 2 & -1 & 0 \end{bmatrix}^T$$

and evaluated its DFT as

$$\mathbf{X} = \begin{bmatrix} 1 & \frac{5}{2} + j\frac{\sqrt{3}}{2} & \frac{5}{2} - j\frac{\sqrt{3}}{2} \end{bmatrix}^T$$

(note the scaling $\mathbf{X} = 3\mathbf{c}$). There is only one pair of conjugate symmetric entries here:

$$X[1] = X^*[2]$$

The amplitude spectrum is given by

$$\begin{bmatrix} 1 & 2.6458 & 2.6458 \end{bmatrix}^T$$

while the phase spectrum is given by

$$\begin{bmatrix} 0 & 0.3335 & -0.3335 \end{bmatrix}^T$$ □

It is always possible to express a real-valued signal vector x as a linear combination of *real-valued* sinusoids at the Fourier frequencies. Indeed, the conjugate symmetry of the spectrum $\mathbf{X}$ allows us to combine complex conjugate terms

$$X[k]e^{j(2\pi/N)kn}$$

and

$$X[N-k]e^{-j(2\pi/N)(N-k)n} = X^*[k]e^{-j(2\pi/N)kn}$$

into a single real-valued sinusoid:

$$2\Re e\left\{X[k]e^{j(2\pi/N)kn}\right\} = 2|X[k]|\cos\left(\frac{2\pi kn}{N} + \angle X[k]\right)$$

(Note that such pairs occur for values of k other than 0 or $N/2$.) The resulting real-valued form of the synthesis equation involves Fourier frequencies in the range $[0, \pi]$ only:

$$x[n] = \frac{1}{N}X[0] + \frac{1}{N}X[N/2](-1)^n + \frac{2}{N} \cdot \sum_{0<k<N/2} |X[k]|\cos\left(\frac{2\pi kn}{N} + \angle X[k]\right)$$

The second term (corresponding to frequency $\omega = \pi$) is present only when $N/2$ is an integer, i.e., when N is even.

Example 3.3.1. (*Continued.*) The representation of

$$\mathbf{x} = \begin{bmatrix} 2 & -1 & 0 \end{bmatrix}^T$$

using real sinusoids involves a constant term (corresponding to $k = 0$) together with sinusoid of frequency $2\pi/3$ (corresponding to $k = 1$ and $k = 2$). Substituting the values of $X[k]$ computed earlier into the last equation, we obtain

$$x[n] = 0.3333 + 1.7639 \cos\left(\frac{2\pi n}{3} + 0.3335\right)$$

for $n = 0, 1, 2$. □

3.3.2 Signal Permutations

Certain important properties of the DFT involve signal transformations in both time and frequency domains. These transformations are simple permutations of the elements of a vector and, as such, can be described by permutation matrices. Recall from Section 2.3 that the columns of a $N \times N$ permutation matrix are the unit vectors $\mathbf{e}^{(0)}, \ldots, \mathbf{e}^{(N-1)}$ listed in arbitrary order.

We introduce three permutations which are particularly important in our study of the DFT.

Circular (or Cyclic) Shift P: This permutation is defined by the relationship

$$P(x[0], x[1], \ldots, x[N-2], x[N-1]) = (x[N-1], x[0], \ldots, x[N-3], x[N-2])$$

and can be illustrated by placing the N entries of $\mathbf{x}$ on a "time wheel", in the same way as the Fourier frequencies appear on the unit circle; namely counterclockwise, starting at angle zero. A circular shift amounts to rotating the wheel counterclockwise by one notch (i.e., by an angle of $2\pi/N$), so that $x[N-1]$ appears at angle zero. The rotation is illustrated in Figure 3.6 for $N = 7$.

If $\mathbf{x}$ is a column vector, a circular shift on $\mathbf{x}$ produces the vector $\mathbf{Px}$, where $\mathbf{P}$ is a permutation matrix given by

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \vdots & 1 & 0 \end{bmatrix}$$

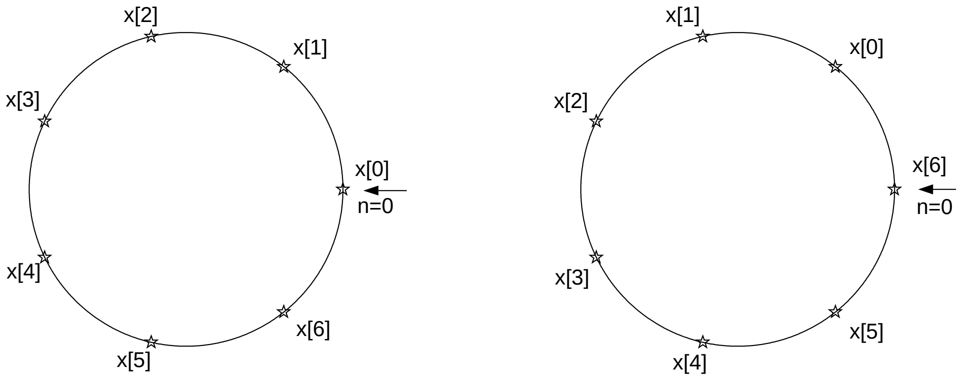


Figure 3.6: Vector $\mathbf{x}$ (left) and its circular shift $\mathbf{Px}$ (right).

Note that removal of the first row and last column of $\mathbf{P}$ yields a $(N-1) \times (N-1)$ identity matrix.

A circular shift is also known as a *rotation.* The term *periodic* is also used instead of *circular*, and *delay* is used instead of *shift.* Clearly, $\mathbf{P}^m$ represents a circular shift by m positions. For $m = N$, the time wheel undergoes one full rotation, i.e.,

$$\mathbf{x} = \mathbf{P}^N\mathbf{x} \qquad \text{or} \qquad \mathbf{P}^N = \mathbf{I}$$

Index Reversal Q: Also known as *linear* index reversal to distinguish it from the circular reversal discussed below, it is defined by

$$Q(x[0], x[1], \ldots, x[N-2], x[N-1]) = (x[N-1], x[N-2], \ldots, x[1], x[0])$$

and described in terms of the permutation matrix

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & \ldots & 0 & 1 \\ 0 & 0 & \ldots & 1 & 0 \\ \vdots & \vdots & \swarrow & \vdots & \vdots \\ 0 & 1 & \vdots & 0 & 0 \\ 1 & 0 & \vdots & 0 & 0 \end{bmatrix}$$

Note that $\mathbf{Q}$ is an identity matrix flipped left-to-right.

Circular Index Reversal R: Also known as *periodic* index reversal, this transformation differs from its linear counterpart in that reversal takes place among entries $x[1], \ldots, x[N-1]$ only, with $x[0]$ kept in the same position:

$$R(x[0], x[1], \ldots, x[N-2], x[N-1]) = (x[0], x[N-1], \ldots, x[2], x[1])$$

This can be illustrated using again a time wheel with the entries of $\mathbf{x}$ arranged in the usual fashion. A circular index reversal amounts to turning the wheel upside down. The 0^{th} always remains in the same position, as does the $(N/2)^{\text{th}}$ entry when N is even (as depicted in Figure 3.7 for the case $N = 8$).

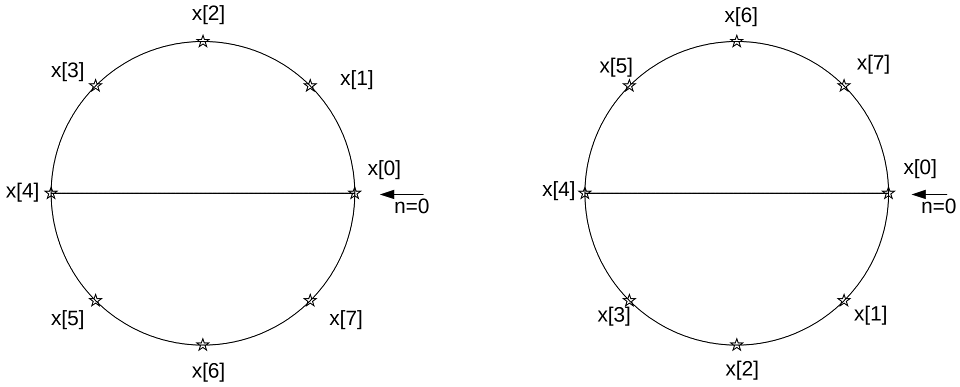


Figure 3.7: Vector $\mathbf{x}$ (left) and its circular time-reverse $\mathbf{R}\mathbf{x}$ (right).

The permutation matrix $\mathbf{R}$ for circular reversal is given by

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 1 & 0 \\ \vdots & \vdots & \swarrow & \vdots & \vdots \\ 0 & 1 & \vdots & 0 & 0 \end{bmatrix}$$

Note that

$$\mathbf{R} = \mathbf{P}\mathbf{Q}$$

i.e., circular reversal of a column vector can be implemented by linear reversal followed by circular shift.

Recall from Section 2.3 that a permutation matrix $\mathbf{\Pi}$ satisfies

$$\mathbf{\Pi}^T\mathbf{\Pi} = \mathbf{I} \qquad \Leftrightarrow \qquad \mathbf{\Pi}^{-1} = \mathbf{\Pi}^T$$

and is thus orthonormal. Since $\mathbf{Q}$ and $\mathbf{R}$ are symmetric, it follows that

$$\mathbf{Q}^{-1} = \mathbf{Q} \qquad \text{and} \qquad \mathbf{R}^{-1} = \mathbf{R}$$

(Note, on the other hand, that $\mathbf{P}$ is not symmetric.)

We note one final property of any permutation matrix $\mathbf{\Pi}$:

Fact. $\mathbf{\Pi}$ *acting on a column vector and* $\mathbf{\Pi}^{-1} = \mathbf{\Pi}^T$ *acting on a row vector both produce the same permutation of indices.*

This is established by noting that

$$(\mathbf{\Pi}\mathbf{x})^T = \mathbf{x}^T\mathbf{\Pi}^T = \mathbf{x}^T\mathbf{\Pi}^{-1} \qquad \square$$

The following example illustrates some of the signal transformations discussed so far.

Example 3.3.2. Let

$$\mathbf{x} = \begin{bmatrix} 1 & -2 & 5 & 3 & 4 & -1 \end{bmatrix}^T$$

Also, let

$$\mathbf{v}^{(3)} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}^T$$

be the third Fourier sinusoid for $N = 6$, corresponding to $\omega = \pi$. The following signals can be expressed compactly in terms of $\mathbf{x}$, $\mathbf{v}^{(3)}$ and the permutation matrices $\mathbf{P}$ and $\mathbf{R}$:

- $\mathbf{x}^{(1)} = \begin{bmatrix} 4 & -1 & 1 & -2 & 5 & 3 \end{bmatrix}^T$
- $\mathbf{x}^{(2)} = \begin{bmatrix} 1 & -1 & 4 & 3 & 5 & -2 \end{bmatrix}^T$
- $\mathbf{x}^{(3)} = \begin{bmatrix} 2 & -3 & 9 & 6 & 9 & -3 \end{bmatrix}^T$
- $\mathbf{x}^{(4)} = \begin{bmatrix} 0 & -1 & 1 & 0 & -1 & 1 \end{bmatrix}^T$
- $\mathbf{x}^{(5)} = \begin{bmatrix} 1 & 2 & 5 & -3 & 4 & 1 \end{bmatrix}^T$
- $\mathbf{x}^{(6)} = \begin{bmatrix} 0 & 4 & 0 & -6 & 0 & 2 \end{bmatrix}^T$

Indeed, we have:

- $\mathbf{x}^{(1)} = \mathbf{P}^2\mathbf{x}$
- $\mathbf{x}^{(2)} = \mathbf{R}\mathbf{x}$
- $\mathbf{x}^{(3)} = \mathbf{x} + \mathbf{R}\mathbf{x}$.
- $\mathbf{x}^{(4)} = \mathbf{x} - \mathbf{R}\mathbf{x}$
- For every k, $x^{(5)}[k] = x[k]v^{(3)}[k]$
- For every k, $x^{(6)}[k] = x[k]\left(v^{(3)}[k] - 1\right)$

3.4 Structural Properties of the DFT: Part II

3.4.1 Permutations of DFT and IDFT Matrices

In Subsection 3.2.2, we observed certain symmetries in the entries of $\mathbf{V}$ and $\mathbf{W}$. The key symmetry properties are:

- Symmetry about the main diagonal, i.e., $\mathbf{V} = \mathbf{V}^T$ and $\mathbf{W} = \mathbf{W}^T$;

- Conjugate symmetry with respect to row index $N/2$:

$$V_{N-n,k} = V^*_{nk} \qquad \text{and} \qquad W_{N-k,n} = W^*_{k,n}$$

  (and similarly with respect to column index $N/2$).

The conjugate symmetry property can be also expressed using the circular reversal matrix $\mathbf{R}$. Applied to either $\mathbf{V}$ or $\mathbf{W}$ as a row permutation, $\mathbf{R}$ leaves the $m = 0^{\text{th}}$ row in the same position, while interchanging rows m and $N - m$ for $0 < m < N/2$. Since these two rows are complex conjugates of each other, and the zeroth row is real-valued, the resulting matrix is the complex conjugate of the original one. Of course, the same is true about column permutations using $\mathbf{R}$, since both $\mathbf{V}$ and $\mathbf{W}$ are symmetric. We thus have

$$\mathbf{RV} = \mathbf{VR} = \mathbf{V}^* = \mathbf{W}$$

and

$$\mathbf{RW} = \mathbf{WR} = \mathbf{W}^* = \mathbf{V}$$

The effect of circular shift on the rows and columns of $\mathbf{V}$ and $\mathbf{W}$ can be explained better by introducing a diagonal matrix $\mathbf{F}$ whose diagonal elements are the entries of the $k = 1^{\text{st}}$ Fourier sinusoid:

$$\mathbf{F} \overset{\text{def}}{=} \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & v & 0 & \dots & 0 \\ 0 & 0 & v^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & v^{N-1} \end{bmatrix}$$

Since the k^{th} power of $\mathbf{F}$ is obtained by raising each diagonal element to that power, it follows that

$$\mathbf{v}^{(k)} = \mathbf{F}^k \mathbf{1}$$

where, as before, $\mathbf{1}$ is the all-ones column vector (same as $\mathbf{v}^{(0)}$). Thus all N Fourier sinusoids can be expressed using powers of $\mathbf{F}$:

$$\mathbf{V} = \left[\begin{array}{ccccc} \mathbf{1} & \mathbf{F1} & \mathbf{F}^2\mathbf{1} & \ldots & \mathbf{F}^{N-1}\mathbf{1} \end{array} \right]$$

A circular shift on the *columns* of $\mathbf{V}$ is obtained by right-multiplying it by $\mathbf{P}^T = \mathbf{P}^{-1}$ (not by $\mathbf{P}$—see the last fact in Subsection 3.3.2). The result is

$$\begin{aligned} \mathbf{VP}^{-1} &= \left[\begin{array}{ccccc} \mathbf{F}^{N-1}\mathbf{1} & \mathbf{1} & \mathbf{F1} & \ldots & \mathbf{F}^{N-2}\mathbf{1} \end{array} \right] \\ &= \left[\begin{array}{ccccc} \mathbf{F}^{-1}\mathbf{1} & \mathbf{1} & \mathbf{F1} & \ldots & \mathbf{F}^{N-2}\mathbf{1} \end{array} \right] \end{aligned}$$

and thus

$$\mathbf{VP}^{-1} = \mathbf{F}^{-1}\mathbf{V}$$

This can be easily generalized to any power m of $\mathbf{P}$:

$$\mathbf{VP}^m = \mathbf{F}^m\mathbf{V}$$

Taking transposes of both sides yields

$$\mathbf{P}^{-m}\mathbf{V} = \mathbf{VF}^m$$

Taking complex conjugates across the last two equations, and noting that

$$\mathbf{V}^* = \mathbf{W} \qquad \text{and} \qquad \mathbf{F}^* = \mathbf{F}^{-1}$$

we obtain

$$\mathbf{WP}^m = \mathbf{F}^{-m}\mathbf{W}$$

and

$$\mathbf{P}^{-m}\mathbf{W} = \mathbf{WF}^{-m}$$

In conclusion, a circular shift on the rows or columns of $\mathbf{V}$ (or $\mathbf{W}$) is equivalent to right- or left-multiplication by a diagonal matrix whose diagonal is given by a suitable Fourier sinusoid.

3.4.2 Summary of Identities

We have defined the following $N \times N$ matrices with the aid of $v = e^{j2\pi/N}$ and $w = v^{-1} = v^*$:

$\mathbf{V}$	:	IDFT matrix, defined by $V_{nk} = v^{kn}$
$\mathbf{W}$	:	DFT matrix, defined by $W_{kn} = w^{kn}$
$\mathbf{F}$	:	diagonal matrix defined by $F_{nn} = v^n$
$\mathbf{P}$	:	circular shift matrix
$\mathbf{R}$	:	circular reversal matrix

We have shown that

$$\mathbf{P}^{-1} = \mathbf{P}^T \tag{3.1}$$
$$\mathbf{R}^{-1} = \mathbf{R}^T = \mathbf{R} \tag{3.2}$$
$$\mathbf{RV} = \mathbf{VR} = \mathbf{V}^* = \mathbf{W} \tag{3.3}$$
$$\mathbf{RW} = \mathbf{WR} = \mathbf{W}^* = \mathbf{V} \tag{3.4}$$
$$\mathbf{VP}^m = \mathbf{F}^m\mathbf{V} \tag{3.5}$$
$$\mathbf{P}^m\mathbf{V} = \mathbf{VF}^{-m} \tag{3.6}$$
$$\mathbf{WP}^m = \mathbf{F}^{-m}\mathbf{W} \tag{3.7}$$
$$\mathbf{P}^m\mathbf{W} = \mathbf{WF}^m \tag{3.8}$$

The identities listed above will be used to derive structural properties of the DFT.

3.4.3 Main Structural Properties of the DFT

In what follows, we will consider the DFT pair

$$\mathbf{x} \longleftrightarrow \mathbf{X}$$

where, again, the time-domain signal appears on the left, and the frequency-domain signal on the right, of the arrow. The two signals are related via the analysis and synthesis equations:

$$\mathbf{X} = \mathbf{Wx} \tag{3.9}$$
$$\mathbf{x} = \frac{1}{N}\mathbf{VX} = \frac{1}{N}\mathbf{W}^*\mathbf{X} \tag{3.10}$$

We will investigate how certain systematic operations on the time-domain signal $\mathbf{x}$ affect its spectrum $\mathbf{X}$, and vice versa.

DFT 3. *(Complex Conjugation) Complex conjugation in the time domain is equivalent to complex conjugation together with circular time reversal in the frequency domain:*

$$\mathbf{y} = \mathbf{x}^* \quad \longleftrightarrow \quad \mathbf{Y} = \mathbf{RX}^* \tag{3.11}$$

Proof. We have

$$\begin{aligned} \mathbf{Y} &= \mathbf{Wy} \\ &= \mathbf{Wx}^* \\ &= \mathbf{RW}^*\mathbf{x}^* \\ &= \mathbf{RX}^* \end{aligned}$$

where the third equality follows from (3.4). □

Remark. Using DFT 3, we can easily deduce DFT 2. Indeed, if $\mathbf{x}$ is real-valued, then

$$\mathbf{x} = \mathbf{x}^*$$

and taking DFT's of both sides, we obtain

$$\mathbf{X} = \mathbf{R}\mathbf{X}^*$$

which is precisely the statement of DFT 2. This relationship expresses a type of conjugate symmetry with respect to circular reversal, which will be henceforth referred to as *circular conjugate symmetry.*

DFT 4. *(Circular Time Reversal) Circular reversal of the entries of* $\mathbf{x}$ *is equivalent to circular reversal of the entries of* $\mathbf{X}$*:*

$$\mathbf{y} = \mathbf{R}\mathbf{x} \quad \longleftrightarrow \quad \mathbf{Y} = \mathbf{R}\mathbf{X} \tag{3.12}$$

Proof. We have

$$\begin{aligned} \mathbf{Y} &= \mathbf{W}\mathbf{y} \\ &= \mathbf{W}\mathbf{R}\mathbf{x} \\ &= \mathbf{R}\mathbf{W}\mathbf{x} \\ &= \mathbf{R}\mathbf{X} \end{aligned}$$

where the third equality follows from (3.4). □

Remark. If we (circularly) time-reverse a *real-valued* signal $\mathbf{x}$, then the resulting signal $\mathbf{y} = \mathbf{R}\mathbf{x}$ has DFT

$$\mathbf{Y} = \mathbf{R}\mathbf{X} = \mathbf{X}^*$$

where the second equality is due to the circular conjugate symmetry of the DFT of a real-valued signal (i.e., DFT 2). In particular, the amplitude and phase spectra of the two signals $\mathbf{x}$ and $\mathbf{y} = \mathbf{R}\mathbf{x}$ are related by

$$\begin{aligned} |Y[k]| &= |X[k]| \\ \angle Y[k] &= -\angle X[k] \end{aligned}$$

for all values of k. The first equation tells us that the two signals contain exactly the same amounts (in terms of amplitudes) of each Fourier frequency. The second equation implies that the relative positions (in time) of these sinusoids will be very different in the two signals. This difference can have a drastic effect on the perceived signal; for example, music played backward bears little resemblance to the original sound. Thus in general, the phase spectrum can be as important as the amplitude spectrum and cannot be ignored in applications such as audio signal compression.

DFT 5. *(Circular Time Delay) Circular time shift of the entries of* $\mathbf{x}$ *by* m *positions is equivalent to multiplication of the entries of* $\mathbf{X}$ *by the corresponding entries of the* $(N-m)^{\text{th}}$ *Fourier sinusoid:*

$$\mathbf{y} = \mathbf{P}^m\mathbf{x} \quad \longleftrightarrow \quad \mathbf{Y} = \mathbf{F}^{-m}\mathbf{X} \tag{3.13}$$

i.e.,

$$Y[k] = v^{k(N-m)}X[k] = v^{-km}X[k] = w^{km}X[k]$$

for $k = 0, \ldots, N-1$.

Proof. We have

$$\begin{aligned} \mathbf{Y} &= \mathbf{W}\mathbf{P}^m\mathbf{x} \\ &= \mathbf{F}^{-m}\mathbf{W}\mathbf{x} \\ &= \mathbf{F}^{-m}\mathbf{X} \end{aligned}$$

where the second equality is due to (3.7). □

DFT 6. *(Multiplication by a Fourier Sinusoid) Entry-by-entry multiplication of* $\mathbf{x}$ *by the* m^{th} *Fourier sinusoid is equivalent to a circular shift of the entries of* $\mathbf{X}$ *by* m *frequency indices:*

$$\mathbf{y} = \mathbf{F}^m\mathbf{x} \quad \longleftrightarrow \quad \mathbf{Y} = \mathbf{P}^m\mathbf{X} \tag{3.14}$$

Proof. We have

$$\begin{aligned} \mathbf{Y} &= \mathbf{W}\mathbf{F}^m\mathbf{x} \\ &= \mathbf{P}^m\mathbf{W}\mathbf{x} \\ &= \mathbf{P}^m\mathbf{X} \end{aligned}$$

where the second equality is due to (3.8). □

Remark. Multiplication of a signal by a sinusoid is also known as *amplitude modulation* (abbreviated as *AM*). In real-world communication systems, amplitude modulation enables a *baseband* signal, i.e., one consisting of (relatively) low frequencies, to be transmitted over a frequency band centered at a much higher *carrier frequency.* A sinusoid (known as the *carrier*) of that frequency is multiplied by the baseband signal (e.g., an audio signal), resulting in the AM signal. The spectrum of the baseband signal and that of the AM signal are basically related by a shift (in frequency), similarly to what DFT 6 implies. For that reason, DFT 6 is also referred to as the *modulation* property.

The similarity between DFT 5 and DFT 6 cannot be overlooked: identical signal operations in two different domains result in *very similar* operations in the opposite domains (the only difference being a complex conjugate). This similarity is a recurrent theme in Fourier analysis, and is particularly prominent in the case of the DFT: basically, the analysis and synthesis equations are identical with the exception of a scaling factor and a complex conjugate. As a result, *computation of any DFT pair yields another DFT pair as a by-product.* This is known as the *duality* property of the DFT, and is stated below.

DFT 7. *(Duality) If* $\mathbf{x} \longleftrightarrow \mathbf{X}$*, then*

$$\mathbf{y} = \mathbf{X} \quad \longleftrightarrow \quad \mathbf{Y} = N\mathbf{R}\mathbf{x} \tag{3.15}$$

Proof. We have

$$\begin{aligned} \mathbf{Y} &= \mathbf{W}\mathbf{y} \\ &= \mathbf{W}\mathbf{X} \\ &= \mathbf{R}\mathbf{V}\mathbf{X} \\ &= N\mathbf{R}\mathbf{x} \end{aligned}$$

where the third and fourth equalities are due to (3.3) and (3.10) (the synthesis equation), respectively. □

Remark. We saw an instance of the duality property in Examples 3.2.2 and 3.2.5, where we showed that

$$\mathbf{e}^{(0)} \longleftrightarrow \mathbf{1}$$

and

$$\mathbf{1} \longleftrightarrow N\mathbf{e}^{(0)}$$

The second DFT pair can be obtained from the first one, and vice versa, by application of DFT 7. In this particular case, both signals are circularly symmetric, and thus circular reversal has no effect:

$$\mathbf{R}\mathbf{e}^{(0)} = \mathbf{e}^{(0)} \qquad \text{and} \qquad \mathbf{R}\mathbf{1} = \mathbf{1}$$

3.5 Structural Properties of the DFT: Examples

3.5.1 Miscellaneous Signal Transformations

Example 3.5.1. Consider the 4-point real-valued signal

$$\mathbf{x} = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}^T$$

We will first compute the DFT $\mathbf{X}$ of $\mathbf{x}$. Using $\mathbf{X}$, we will then derive the DFT's of the following signals:

- $\mathbf{x}^{(1)} = \begin{bmatrix} 1 & 4 & 3 & 2 \end{bmatrix}^T$
- $\mathbf{x}^{(2)} = \begin{bmatrix} 4 & 1 & 2 & 3 \end{bmatrix}^T$
- $\mathbf{x}^{(3)} = \begin{bmatrix} 3 & 4 & 1 & 2 \end{bmatrix}^T$
- $\mathbf{x}^{(4)} = \begin{bmatrix} 1 & -2 & 3 & -4 \end{bmatrix}^T$
- $\mathbf{x}^{(5)} = \begin{bmatrix} 5 & -1+j & -1 & -1-j \end{bmatrix}^T$
- $\mathbf{x}^{(6)} = \begin{bmatrix} 5 & -1 & -1 & -1 \end{bmatrix}^T$

The Fourier frequencies in this case are 0, $\pi/2$, π and $3\pi/2$. We have

$$\mathbf{X} = \mathbf{W}\mathbf{x} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 10 \\ -2+2j \\ -2 \\ -2-2j \end{bmatrix}$$

Note that $\mathbf{X}$ exhibits the conjugate symmetry common to the spectra of all real-valued signals (DFT 2).

Signals $\mathbf{x}^{(1)}$ through $\mathbf{x}^{(6)}$ are derived from either $\mathbf{x}$ or $\mathbf{X}$. Their spectra can be computed using known structural properties of the DFT.

- $\mathbf{x}^{(1)} = \begin{bmatrix} 1 & 4 & 3 & 2 \end{bmatrix}^T$ is the circular time-reverse of $\mathbf{x}$, i.e., $\mathbf{x}^{(1)} = \mathbf{R}\mathbf{x}$. By DFT 4,

$$\mathbf{X}^{(1)} = \mathbf{R}\mathbf{X} = \begin{bmatrix} 10 & -2-2j & -2 & -2+2j \end{bmatrix}^T$$

- $\mathbf{x}^{(2)} = \begin{bmatrix} 4 & 1 & 2 & 3 \end{bmatrix}^T$ is the circular delay of $\mathbf{x}$ by one time unit, i.e., $\mathbf{x}^{(2)} = \mathbf{P}\mathbf{x}$. By DFT 5, the k^{th} entry of $\mathbf{X}$ is multiplied by $w^k = (-j)^k$, for each value of k:

$$\mathbf{X}^{(2)} = \mathbf{F}^{-1}\mathbf{X} = \begin{bmatrix} 10 & 2+2j & 2 & 2-2j \end{bmatrix}^T$$

- $\mathbf{x}^{(3)} = \left[\begin{array}{cccc} 3 & 4 & 1 & 2 \end{array}\right]^T$ is the circular delay of $\mathbf{x}$ by *two* time units, i.e., $\mathbf{x}^{(3)} = \mathbf{P}^2\mathbf{x}$. Again by DFT 5, the k^{th} entry of $\mathbf{X}$ is multiplied by $w^{2k} = (-1)^k$, for each value of k:

$$\mathbf{X}^{(3)} = \mathbf{F}^{-2}\mathbf{X} = \left[\begin{array}{cccc} 10 & 2-2j & -2 & 2+2j \end{array}\right]^T$$

- $\mathbf{x}^{(4)} = \left[\begin{array}{cccc} 1 & -2 & 3 & -4 \end{array}\right]^T$ is obtained by multiplying the entries of $\mathbf{x}$ by those of the Fourier sinusoid $(-1)^n = v^{2n}$, i.e., $\mathbf{x}^{(4)} = \mathbf{F}^2\mathbf{x}$. By DFT 6, the spectrum is shifted by $m = 2$ frequency indices, or, in terms of actual frequencies, by π radians. The resulting spectrum is

$$\mathbf{X}^{(4)} = \mathbf{P}^2\mathbf{X} = \left[\begin{array}{cccc} -2 & -2-2j & 10 & -2+2j \end{array}\right]^T$$

- $\mathbf{x}^{(5)} = \left[\begin{array}{cccc} 5 & -1+j & -1 & -1-j \end{array}\right]^T$ equals $\mathbf{X}/2$. By DFT 7 (duality), we have that

$$\mathbf{X}^{(6)} = 4\mathbf{R}(\mathbf{x}/2) = \left[\begin{array}{cccc} 2 & 8 & 6 & 4 \end{array}\right]^T$$

- $\mathbf{x}^{(6)} = \left[\begin{array}{cccc} 5 & -1 & -1 & -1 \end{array}\right]^T$ equals the real part of $\mathbf{x}^{(5)}$, i.e.,

$$\mathbf{x}^{(6)} = \frac{\mathbf{x}^{(5)} + \left(\mathbf{x}^{(5)}\right)^*}{2}$$

  By DFT 3, complex conjugation in the time domain is equivalent to complex conjugation together with circular reversal in the frequency domain. Thus

$$\mathbf{X}^{(6)} = \frac{1}{2}\left[\begin{array}{cccc} 2 & 8 & 6 & 4 \end{array}\right]^T + \frac{1}{2}\left[\begin{array}{cccc} 2 & 4 & 6 & 8 \end{array}\right]^T = \left[\begin{array}{cccc} 2 & 6 & 6 & 6 \end{array}\right]^T \quad \square$$

3.5.2 Exploring Symmetries

The spectrum of a real-valued signal is always circularly conjugate-symmetric; this was established in DFT 2 and was illustrated in Examples 3.3.1 and 3.5.1. It turns out that this property also holds with the time and frequency domains interchanged, i.e., a signal which exhibits circular conjugate symmetry in the time domain has a real-valued spectrum. This can be seen by taking DFT's on both sides of the identity

$$\mathbf{x} = \mathbf{R}\mathbf{x}^*$$

and then applying DFT 4 to obtain

$$\mathbf{X} = \mathbf{R}\mathbf{R}\mathbf{X}^* = \mathbf{X}^*$$

This means that $\mathbf{X}$ is real-valued.

It follows that if a signal $\mathbf{x}$ happens to be *both* real-valued and circularly symmetric, then its spectrum $\mathbf{X}$ will have the same properties. (Conjugate symmetry and symmetry are equivalent notions for a real-valued signal, which always equals its complex conjugate.)

We summarize the above observations as follows.

Fact. *If a signal is real-valued in either domain (time or frequency), then it exhibits circular conjugate symmetry in the other domain. A signal exhibiting both properties in one domain also exhibits both properties in the other domain. The terms* ***symmetry*** *and* ***conjugate symmetry*** *are equivalent for real-valued signals.*

The following example illustrates operations on circularly symmetric, real-valued signals that preserve both properties.

Example 3.5.2. Consider the 16-point signal $\mathbf{x}$ whose spectrum $\mathbf{X}$ is given by

$$\mathbf{X} = \begin{bmatrix} 4 & 3 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 \end{bmatrix}^T$$

$\mathbf{X}$ is shown in Figure E.3.5.2 (i). Clearly, $\mathbf{X}$ is both real-valued and circularly symmetric. By the foregoing discussion, the time-domain signal $\mathbf{x}$ has the same properties (and can be computed quite easily by issuing the command `x = ifft(X)` in MATLAB).

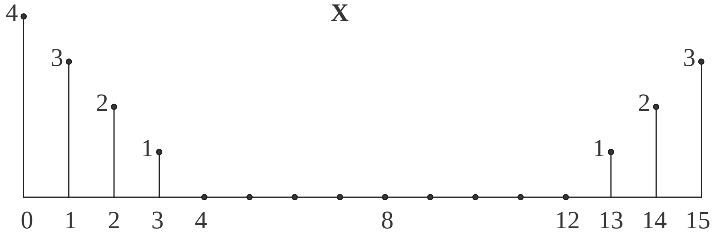


Figure E.3.5.2 (i)

We are interested in determining whether the two fundamental *time-domain* operations:

- circular delay by m time units

- multiplication by the m^{th} Fourier sinusoid

preserve either or both properties of $\mathbf{x}$, (i.e., real values and circular symmetry).

A circular time delay of m units in $\mathbf{x}$ clearly preserves the real values in $\mathbf{x}$, and thus also preserves the circular symmetry in $\mathbf{X}$. To see how circular symmetry of $\mathbf{x}$ is affected by this operation, we consider the spectrum $\mathbf{X}$, which undergoes multiplication by the m^{th} Fourier sinusoid: each $X[k]$ is multiplied by $e^{-j(\pi/8)km}$. Unless $m = 0$ (no delay) or $m = 8$, the resulting spectrum will contain complex values, which means that the (delayed) time-domain signal will no longer be circularly symmetric. Thus the only nontrivial value of m that preserves circular symmetry is $m = 8$. The delayed signal is given by

$$\mathbf{x}^{(1)} = \mathbf{P}^8\mathbf{x}$$

and its spectrum equals

$$X^{(1)}[k] = e^{-j\pi k}X[k] = (-1)^k X[k], \qquad k = 0, \dots, 15$$

The spectrum $\mathbf{X}^{(1)}$ is plotted in Figure E.3.5.2 (ii).

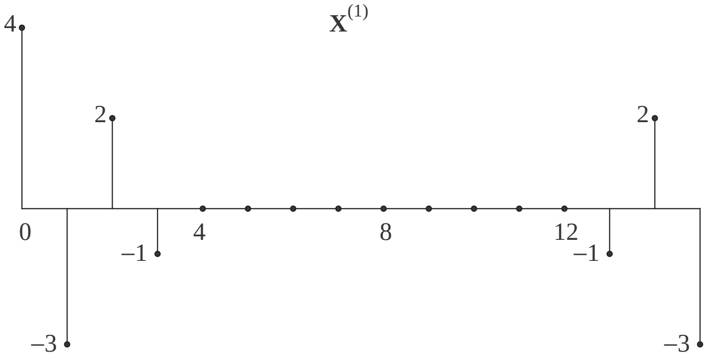


Figure E.3.5.2 (ii)

We note also that summing together two versions of $\mathbf{x}$ that have been delayed by complementary amounts, i.e., m and $-m$ (or $16-m$) time units, will also preserve the circular symmetry in $\mathbf{x}$ regardless of the value of m. This is because

$$e^{-j(\pi/8)km} + e^{j(\pi/8)km} = 2\cos\left(\frac{\pi km}{8}\right)$$

and therefore

$$\mathbf{x}^{(2)} = \mathbf{P}^m\mathbf{x} + \mathbf{P}^{-m}\mathbf{x} \;\longleftrightarrow\; X^{(2)}[k] = 2\cos\left(\frac{\pi k m}{8}\right) X[k]$$

Since the spectrum $\mathbf{X}^{(2)}$ is real-valued, $\mathbf{x}^{(2)}$ is circularly symmetric. Figure E.3.5.2 (iii) illustrates $\mathbf{X}^{(2)}$ for the case $m = 4$. The graph was generated by multiplying each entry of $\mathbf{X}$ by the corresponding entry of

$$\begin{bmatrix} 2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 \end{bmatrix}^T$$

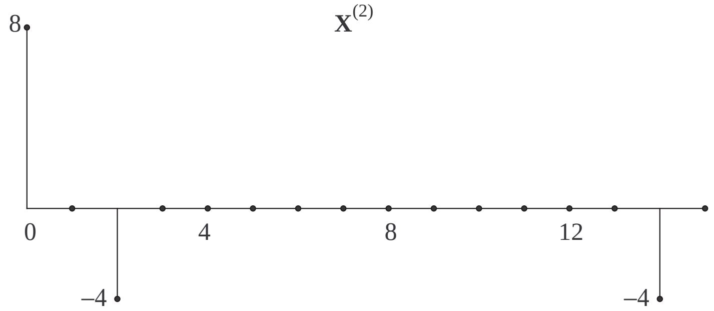


Figure E.3.5.2 (iii)

Multiplication of $\mathbf{x}$ by a complex Fourier sinusoid of frequency $m\pi/8$ will result in certain entries of $\mathbf{x}$ taking complex values, unless $m = 0$ or $m = 8$. Correspondingly, the spectrum will undergo circular shift by m frequency indices, and circular symmetry will be preserved only in the cases $m = 0$ and $m = 8$. Thus the only nontrivial value of m that preserves real values in $\mathbf{x}$ is $m = 8$, for which

$$x^{(3)}[n] = e^{j\pi n}x[n] = (-1)^n x[n], \qquad n = 0, \ldots, 15$$

and

$$\mathbf{X}^{(3)} = \mathbf{P}^8\mathbf{X}$$

The spectrum $\mathbf{X}^{(3)}$ is plotted in Figure E.3.5.2 (iv).

We also note that since the Fourier sinusoids are circularly conjugate-symmetric, the element-wise product of $\mathbf{x}$ and $\mathbf{v}^{(m)}$ will also be circularly conjugate-symmetric, whether it is real or complex.

Finally, we see that if we multiply $\mathbf{x}$ by the sum of two complex Fourier sinusoids which are conjugates of each other, i.e., have frequency indices m

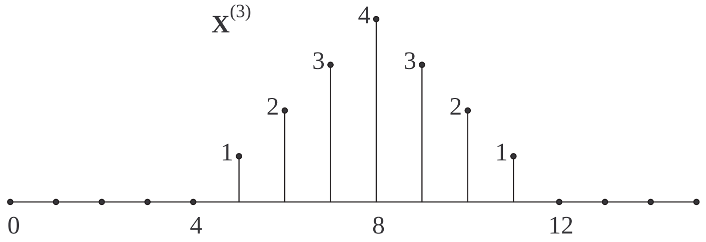


Figure E.3.5.2 (iv)

and $16 - m$ (or simply $-m$), the resulting signal will be real-valued, since

$$e^{-j(\pi/8)mn} + e^{j(\pi/8)mn} = 2\cos\left(\frac{\pi mn}{8}\right)$$

(same identity as encountered earlier). Also,

$$x^{(4)}[n] = 2\cos\left(\frac{\pi mn}{8}\right)x[n] \quad\longleftrightarrow\quad \mathbf{X}^{(4)} = \mathbf{P}^m\mathbf{X} + \mathbf{P}^{-m}\mathbf{X}$$

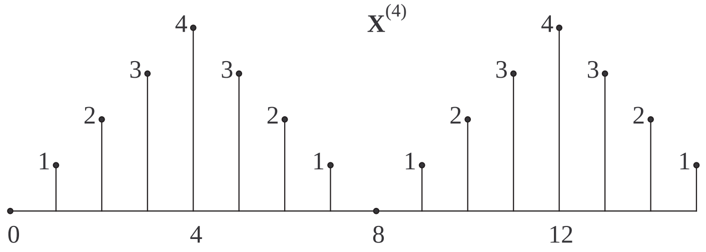


Figure E.3.5.2 (v)

Figure E.3.5.2 (v) illustrates $\mathbf{X}^{(4)}$ in the case $m = 4$. Note the circular symmetry in $\mathbf{X}^{(4)}$, as well as the relationship between $\mathbf{X}^{(4)}$ and the original spectrum $\mathbf{X}$. In this case, each entry of $\mathbf{x}$ (in the time domain) is multiplied by the corresponding entry of

$$\begin{bmatrix} 2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 \end{bmatrix}^T \qquad \square$$

3.6 Multiplication and Circular Convolution

The structural properties of the discrete Fourier transform considered thus far involved operations on a single signal vector of fixed length. In this section, we examine two specific ways of combining together two (or more) arbitrary signals in the time domain, and study the spectra of the resulting signals.

3.6.1 Duality of Multiplication and Circular Convolution

We have seen (in DFT 1) that DFT is a linear transformation, i.e., the DFT of a linear combination of two signals is the linear combination of their DFT's:

$$\alpha\mathbf{x} + \beta\mathbf{y} \;\longleftrightarrow\; \alpha\mathbf{X} + \beta\mathbf{Y}$$

There are other important ways in which two signal vectors $\mathbf{x}$ and $\mathbf{y}$ can be combined in both time and frequency domains. We will examine two such operations.

Definition 3.6.1. The (element-wise) product of N-point vectors $\mathbf{x}$ and $\mathbf{y}$ is the vector $\mathbf{s}$ defined by

$$s[n] = x[n]y[n]\;, \qquad n = 0, \dots, N-1 \qquad \square$$

A special case of the product was encountered earlier, where $\mathbf{y} = \mathbf{v}^{(m)}$ (the m^{th} Fourier sinusoid).

Let us examine the DFT $\mathbf{S}$ of the product signal $\mathbf{s}$, which is given by the analysis equation:

$$S[k] = \sum_{n=0}^{N-1} x[n]y[n]w^{kn}\;, \qquad k = 0, \dots, N-1$$

(We note that in this case, we do not have a compact expression for the *entire* vector $\mathbf{S}$ as a product of the vectors $\mathbf{x}$, $\mathbf{y}$ and the matrix $\mathbf{W}$.) Noting that w^{kn} is the $(n,n)^{\text{th}}$ entry of the diagonal matrix $\mathbf{F}^{-k}$, we can write

$$S[k] = \begin{bmatrix} x[0] & x[1] & \dots & x[N-1] \end{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & w^k & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w^{k(N-1)} \end{bmatrix} \begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-1] \end{bmatrix}$$

i.e.,

$$S[k] = \mathbf{x}^T\mathbf{F}^{-k}\mathbf{y}$$

We seek an expression for $S[k]$ in terms of the spectra $\mathbf{X}$ and $\mathbf{Y}$. To that end, we use the matrix forms of the analysis and synthesis equations, as well as the known identities

$$\mathbf{V} = \mathbf{V}^T, \qquad \mathbf{V}\mathbf{F}^{-k} = \mathbf{P}^k\mathbf{V} \qquad \text{and} \qquad \mathbf{V} = \mathbf{R}\mathbf{W}$$

We obtain

$$\begin{aligned} S[k] &= \left(\frac{1}{N}\mathbf{V}\mathbf{X}\right)^T \mathbf{F}^{-k}\mathbf{y} \\ &= \frac{1}{N}\mathbf{X}^T\mathbf{V}\mathbf{F}^{-k}\mathbf{y} \\ &= \frac{1}{N}\mathbf{X}^T\mathbf{P}^k\mathbf{V}\mathbf{y} \\ &= \frac{1}{N}\mathbf{X}^T\mathbf{P}^k\mathbf{R}\mathbf{W}\mathbf{y} \\ &= \frac{1}{N}\mathbf{X}^T\mathbf{P}^k\mathbf{R}\mathbf{Y} \end{aligned}$$

The operation involving $\mathbf{X}$, $\mathbf{Y}$ and the permutation matrices $\mathbf{R}$ (circular reversal) and $\mathbf{P}$ (circular shift) in the last expression is known as the *circular convolution* of the vectors $\mathbf{X}$ and $\mathbf{Y}$. It is defined for any two vectors of the same length.

Definition 3.6.2. The *circular convolution* of the N-point vectors $\mathbf{a}$ and $\mathbf{b}$ is the N-point vector denoted by

$$\mathbf{a} \circledast \mathbf{b}$$

and given by

$$(\mathbf{a} \circledast \mathbf{b})[n] = \mathbf{a}^T\mathbf{P}^n\mathbf{R}\mathbf{b}\ , \qquad n = 0, \dots, N-1$$ □

We can now write our result as follows.

DFT 8. *(Multiplication of Two Signals) If* $\mathbf{x} \longleftrightarrow \mathbf{X}$ *and* $\mathbf{y} \longleftrightarrow \mathbf{Y}$, *then*

$$x[n]y[n] \quad \longleftrightarrow \quad \frac{1}{N}\mathbf{X} \circledast \mathbf{Y}$$ □

Fact. *Since* $x[n]y[n] = y[n]x[n]$, *we also have* $\mathbf{X} \circledast \mathbf{Y} = \mathbf{Y} \circledast \mathbf{X}$, *i.e., circular convolution is symmetric in its two arguments.* □

Multiplication in the time domain has many applications in communications (e.g., amplitude modulation, signal spreading) and signal processing (e.g., windowing in spectral analysis and filter design). By DFT 8, multiplication in the time domain is equivalent to (scaled) circular convolution in the frequency domain.

As it turns out, convolution is equally (if not more) important as a *time-domain* operation, since it can be used to compute the response of a linear system to an arbitrary input vector; this feature will be explored further in the following chapter. In the meantime, we note that DFT 8 has a dual property: circular convolution in the time domain corresponds to multiplication in the frequency domain.

DFT 9. *(Circular Convolution of Two Signals) If* $\mathbf{x} \longleftrightarrow \mathbf{X}$ *and* $\mathbf{y} \longleftrightarrow \mathbf{Y}$*, then*

$$\mathbf{x} \circledast \mathbf{y} \quad \longleftrightarrow \quad X[k]Y[k]$$

Proof. We follow an argument parallel to the proof of DFT 8. We consider the product signal $S[k] = X[k]Y[k]$ (now in the frequency domain), and use the synthesis equation (which differs from the analysis equation by a complex conjugate and the factor N) to express the time-domain signal $\mathbf{s}$ as

$$s[n] = \frac{1}{N}\mathbf{X}^T\mathbf{F}^n\mathbf{Y}$$

We need to show that $s[n]$ is the n^{th} entry of $\mathbf{x} \circledast \mathbf{y}$. Indeed,

$$\begin{aligned} s[n] &= \left(\frac{1}{N}\mathbf{W}\mathbf{x}\right)^T \mathbf{F}^n\mathbf{Y} \\ &= \frac{1}{N}\mathbf{x}^T\mathbf{W}\mathbf{F}^n\mathbf{Y} \\ &= \frac{1}{N}\mathbf{x}^T\mathbf{P}^n\mathbf{W}\mathbf{Y} \\ &= \frac{1}{N}\mathbf{x}^T\mathbf{P}^n\mathbf{R}(\mathbf{V}\mathbf{Y}) \\ &= \mathbf{x}^T\mathbf{P}^n\mathbf{R}\mathbf{y} \\ &= (\mathbf{x} \circledast \mathbf{y})[n] \end{aligned}$$

as needed. Alternatively, this result can be proved using DFT 7 (duality) on the pairs $\mathbf{x} \longleftrightarrow \mathbf{X}$, $\mathbf{y} \longleftrightarrow \mathbf{Y}$ and $x[n]y[n] \longleftrightarrow N^{-1}(\mathbf{X} \circledast \mathbf{Y})$. □

3.6.2 Computation of Circular Convolution

The computation of the circular convolution $\mathbf{x} \circledast \mathbf{y}$ for two N-point vectors $\mathbf{x}$ and $\mathbf{y}$ can be summarized as follows:

- Circularly reverse $\mathbf{y}$ to obtain $\mathbf{Ry}$.
- For each $n = 0, \ldots, N-1$, shift $\mathbf{Ry}$ circularly by n indices to obtain $\mathbf{P}^n\mathbf{Ry}$.
- Compute $(\mathbf{x} \circledast \mathbf{y})[n]$ as $\mathbf{x}^T\mathbf{P}^n\mathbf{Ry}$, i.e., as the (unconjugated) dot product of $\mathbf{x}$ and $\mathbf{P}^n\mathbf{Ry}$.

By symmetry of circular convolution, $\mathbf{x}$ and $\mathbf{y}$ can be interchanged in the above computation.

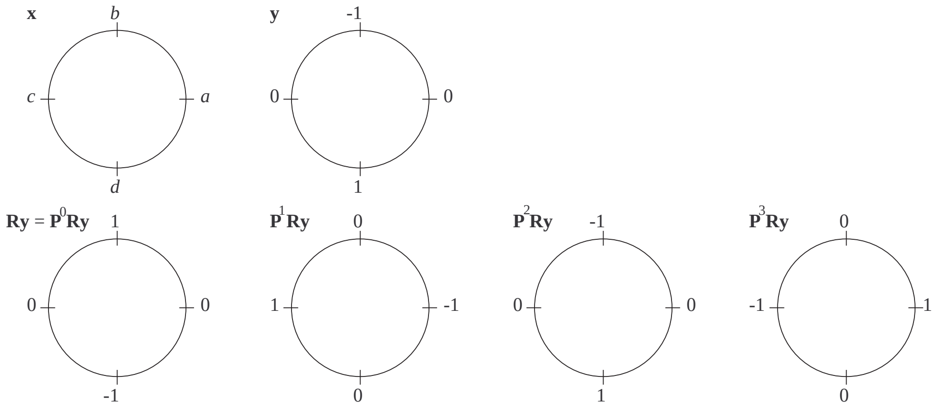
Example 3.6.1. Consider the four-point vectors

$$\mathbf{x} = \begin{bmatrix} a & b & c & d \end{bmatrix}^T \qquad \text{and} \qquad \mathbf{y} = \begin{bmatrix} 0 & -1 & 0 & 1 \end{bmatrix}^T$$

and let

$$\mathbf{s} = \mathbf{x} \circledast \mathbf{y}$$

The vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{P}^n\mathbf{Ry}$ for $n = 0, 1, 2, 3$ are depicted using four-point time wheels.



Example 3.6.1

We have

$$\mathbf{P}^0\mathbf{Ry} \;=\; \begin{bmatrix} 0 & 1 & 0 & -1 \end{bmatrix}^T$$

$$\begin{aligned}
\mathbf{P}^1\mathbf{R}\mathbf{y} &= \begin{bmatrix} -1 & 0 & 1 & 0 \end{bmatrix}^T \\
\mathbf{P}^2\mathbf{R}\mathbf{y} &= \begin{bmatrix} 0 & -1 & 0 & 1 \end{bmatrix}^T \\
\mathbf{P}^3\mathbf{R}\mathbf{y} &= \begin{bmatrix} 1 & 0 & -1 & 0 \end{bmatrix}^T
\end{aligned}$$

and thus

$$\begin{aligned}
s[0] &= \mathbf{x}^T\mathbf{P}^0\mathbf{R}\mathbf{y} = b-d \\
s[1] &= \mathbf{x}^T\mathbf{P}^1\mathbf{R}\mathbf{y} = c-a \\
s[2] &= \mathbf{x}^T\mathbf{P}^2\mathbf{R}\mathbf{y} = d-b \\
s[3] &= \mathbf{x}^T\mathbf{P}^3\mathbf{R}\mathbf{y} = a-c
\end{aligned}$$

Therefore

$$\mathbf{s} = \begin{bmatrix} b-d & c-a & d-b & a-c \end{bmatrix}^T$$

□

Example 3.6.2. We verify DFT 9 for

$$\mathbf{x} = \begin{bmatrix} 2 & -3 & 5 & -3 \end{bmatrix}^T \qquad \text{and} \qquad \mathbf{y} = \begin{bmatrix} 0 & -1 & 0 & 1 \end{bmatrix}^T$$

The signal $\mathbf{y}$ is the same as in Example 3.6.1, thus

$$\mathbf{s} = \mathbf{x} \circledast \mathbf{y} = \begin{bmatrix} 0 & 3 & 0 & -3 \end{bmatrix}^T$$

The DFT's are easily computed as:

$$\begin{aligned}
\mathbf{X} &= \begin{bmatrix} 1 & -3 & 13 & -3 \end{bmatrix}^T \\
\mathbf{Y} &= \begin{bmatrix} 0 & 2j & 0 & -2j \end{bmatrix}^T \\
\mathbf{S} &= \begin{bmatrix} 0 & -6j & 0 & 6j \end{bmatrix}^T
\end{aligned}$$

Indeed,

$$S[k] = X[k]Y[k]$$

for all values of k. □

3.7 Periodic and Zero-Padded Extensions of a Signal

3.7.1 Definitions

We now shift our focus to signals derived from a basic signal vector **s** by extending its length. In what follows, we will assume that **s** has length L, and will derive two distinct types of signals of variable length $N \geq L$ based on **s**.

Definition 3.7.1. The N-point *periodic extension* of **s** is the signal **x** defined by

$$x[n] = \begin{cases} s[n], & 0 \leq n \leq L-1; \\ x[n-L], & L \leq n \leq N-1. \end{cases} \qquad \square$$

Definition 3.7.2. The N-point *zero-padded extension* of **s** is the signal **y** defined by

$$y[n] = \begin{cases} s[n], & 0 \leq n \leq L-1; \\ 0, & L \leq n \leq N-1. \end{cases} \qquad \square$$

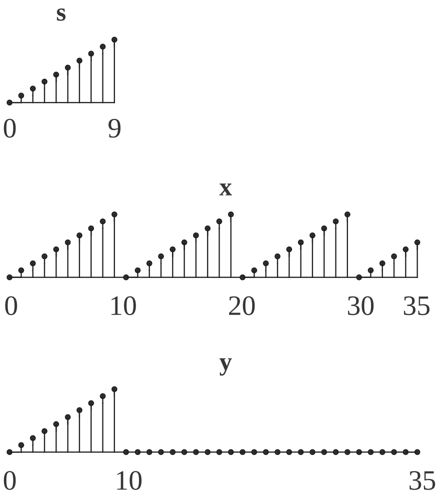


Figure 3.8: The 10-point signal **s**, its 36-point periodic extension **x** and its 36-point zero-padded extension **y**.

The two types of extensions are illustrated in Figure 3.8, where $L = 10$ and $N = 36$. Note that the signals **x** and **y** are fundamentally different. In the case of $\mathbf{x}$, the same activity is observed every L time units; $\mathbf{y}$, on the other hand, exhibits no activity after time $n = L - 1$.

Since the extensions $\mathbf{x}$ and $\mathbf{y}$ are formed by either replicating the basic signal $\mathbf{s}$ or appending zeros to it, both DFT's $\mathbf{X}$ and $\mathbf{Y}$ are computed using elements of the vector $\mathbf{s}$. However, the relationships between $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{S}$ are not at all apparent. This is because the Fourier frequencies for $\mathbf{X}$ and $\mathbf{Y}$ depend on N, and so do the associated Fourier sinusoids. Since the elements of $\mathbf{X}$ and $\mathbf{Y}$ are inner products involving such Fourier sinusoids, there are no identifiable correspondences or similarities between $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{S}$ in the general case (i.e., where N is arbitrary). One important exception is the special case where N is an integer multiple of L.

Fact. *If $N = ML$, where M is an integer, then the set of Fourier frequencies for an L-point signal is formed by taking every M^{th} Fourier frequency for an N-point signal, starting with the zeroth frequency ($\omega = 0$).*

This fact is easily established by noting that the k^{th} Fourier frequency for an L-point signal is given by

$$k\left(\frac{2\pi}{L}\right) = kM\left(\frac{2\pi}{ML}\right) = kM\left(\frac{2\pi}{N}\right)$$

and thus equals the kM^{th} frequency for an N-point signal. □

Figure 3.9 illustrates this fact in the case where $L = 4$, $M = 3$ and $N = 12$.

3.7.2 Periodic Extension to an Integer Number of Periods

Assuming that $N = ML$, we have the following relationship between $\mathbf{X}$ and $\mathbf{S}$.

Fact. *If $N = ML$, then the DFT of the N-point periodic extension $\mathbf{x}$ of an L-point signal $\mathbf{s}$ is given by*

$$X[r] = \begin{cases} MS[r/M], & \textit{if } r/M = \textit{integer;} \\ 0, & \textit{otherwise.} \end{cases}$$

Proof. The assumption $N = ML$ implies that $\mathbf{x}$ contains an exact number of copies of $\mathbf{s}$. We know that

$$\mathbf{s} = \frac{1}{L}\mathbf{V}_L\mathbf{S}$$

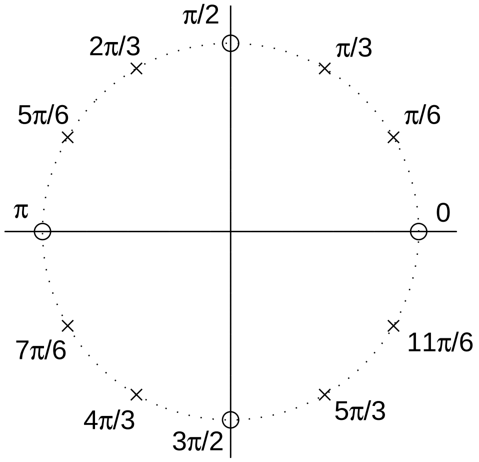


Figure 3.9: Fourier frequencies for a 4-point signal (∘) and the remaining Fourier frequencies for a 12-point signal (×).

where $\mathbf{V}_L$ is the matrix of Fourier sinusoids for an L-point signal. By replicating the above equation M times, we obtain

$$\begin{bmatrix} \mathbf{s} \\ \mathbf{s} \\ \vdots \\ \mathbf{s} \end{bmatrix} = \frac{1}{L} \begin{bmatrix} \mathbf{V}_L \\ \mathbf{V}_L \\ \vdots \\ \mathbf{V}_L \end{bmatrix} \mathbf{S} \tag{3.16}$$

The $N \times 1$ vector on the left-hand side of the equation is simply $\mathbf{x}$. The k^{th} column of the $N \times L$ matrix on the right-hand side is the N-point periodic extension of

$$e^{j(2\pi/L)kn} \ , \quad n = 0, \ldots L-1$$

which is the k^{th} Fourier sinusoid for an L-point signal. Since for every integer p,

$$e^{j(2\pi/L)kn} = e^{j(2\pi/L)k(n+pL)}$$

it follows that the k^{th} column of the $N \times L$ matrix is simply given by

$$e^{j(2\pi/L)kn} = e^{j(2\pi/N)kMn} \ , \quad n = 0, \ldots N-1$$

But this is just the $(kM)^{\text{th}}$ Fourier sinusoid for an N-point signal, i.e., the $(kM)^{\text{th}}$ column of the matrix $\mathbf{V}_N$. Thus (3.16) implies that $\mathbf{x}$ is a linear combination of no more than L Fourier sinusoids. Not surprisingly, these sinusoids have the same frequencies as the Fourier components of $\mathbf{s}$.

We have therefore established that $X[r] = 0$ if r is not a multiple of M. To determine the value of $X[kM]$ for $k = 0, \ldots, L-1$, we compare (3.16) with the synthesis equation

$$\mathbf{x} = \frac{1}{N}\mathbf{V}_N\mathbf{X}$$

and obtain the identity

$$\frac{1}{L}\begin{bmatrix} \mathbf{V}_L \\ \mathbf{V}_L \\ \vdots \\ \mathbf{V}_L \end{bmatrix}\mathbf{S} = \frac{1}{N}\mathbf{V}_N\mathbf{X}$$

This can only be true if $X[kM] = (N/L)S[k] = MS[k]$ for $k = 0, \ldots, L-1$. □

Example 3.7.1. If

$$\mathbf{s} = \begin{bmatrix} a & b & c & d \end{bmatrix}^T$$

has DFT

$$\mathbf{S} = \begin{bmatrix} A & B & C & D \end{bmatrix}^T ,$$

then

$$\mathbf{x} = \begin{bmatrix} a & b & c & d & a & b & c & d & a & b & c & d \end{bmatrix}^T$$

has DFT

$$\mathbf{X} = \begin{bmatrix} 3A & 0 & 0 & 3B & 0 & 0 & 3C & 0 & 0 & 3D & 0 & 0 \end{bmatrix}^T$$ □

In summary, the DFT allows us to express an L-point time-domain signal $\mathbf{s}$ as a linear combination of L sinusoids at frequencies which are multiples of $2\pi/L$. The same L sinusoids, with the same coefficients, could be used to represent the N-point periodic extension $\mathbf{x}$ of $\mathbf{s}$. This representation would not, in general, be consistent with the one provided by Fourier analysis (i.e., the DFT) of $\mathbf{x}$; this is because the Fourier frequencies for $\mathbf{s}$ may not form a subset of the Fourier frequencies for $\mathbf{x}$. In the special case where $\mathbf{x}$ consists of an integer number of copies of $\mathbf{s}$, then all the original frequencies for $\mathbf{s}$ will be Fourier frequencies for $\mathbf{x}$, and $\mathbf{x}$ can be represented as a linear combination of L Fourier sinusoids only; none of the remaining $N - L$ Fourier sinusoids will appear in $\mathbf{x}$.

3.7.3 Zero-Padding to a Multiple of the Signal Length

We now turn to the zero-padded extension $\mathbf{y}$ of $\mathbf{s}$ in the case where $N = ML$. We have the following relationship between the spectra $\mathbf{Y}$ and $\mathbf{S}$.

Fact. *If $N = ML$, then the DFT of an L-point signal $\mathbf{s}$ can be obtained from the DFT of its N-point zero-padded extension $\mathbf{y}$ by sampling:*

$$S[k] = Y[kM] \ , \qquad k = 0, \dots, L-1$$

Proof. As we noted earlier, the frequency $\omega = k(2\pi/L)$ is both the k^{th} Fourier frequency for $\mathbf{s}$ and the $(kM)^{\text{th}}$ Fourier frequency for $\mathbf{y}$; the corresponding DFT values are $S[k]$ and $Y[kM]$, respectively. We have

$$\begin{aligned} Y[kM] &= \sum_{n=0}^{N-1} y[n] e^{-j(2\pi/N)kMn} \\ &= \sum_{n=0}^{L-1} s[n] e^{-j(2\pi/L)kn} \\ &= S[k] \end{aligned}$$

where the first equality follows from the definition of $\mathbf{y}$ (i.e., the fact that $y[n]$ coincides with $s[n]$ for the first L time indices and equals zero thereafter). □

Example 3.7.2. If

$$\mathbf{y} = \left[\begin{array}{cccccccccccc} a & b & c & d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right]^T$$

has DFT

$$\mathbf{Y} = \left[\begin{array}{cccccccccccc} Y_0 & Y_1 & Y_2 & Y_3 & Y_4 & Y_5 & Y_6 & Y_7 & Y_8 & Y_9 & Y_{10} & Y_{11} \end{array} \right]^T$$

then

$$\mathbf{s} = \left[\begin{array}{cccc} a & b & c & d \end{array} \right]^T$$

has DFT

$$\mathbf{S} = \left[\begin{array}{cccc} Y_0 & Y_3 & Y_6 & Y_9 \end{array} \right]^T$$ □

The DFT of a finite-length signal vector can thus be obtained by sampling the DFT of its zero-padded extension, provided the number of appended zeros is an integer multiple of the (original) signal length.

3.8 Detection of Sinusoids Using the Discrete Fourier Transform

3.8.1 Introduction

An important application of signal analysis is the identification of different components present in a signal. For example, given a musical recording, we may be interested in identifying all instruments being played at a particular time instant. If the recording contains vocals, we may also want to count and, if possible, identify the different voices that are being heard. With minimal training, the human ear can perform most of the above mentioned tasks quite reliably. With additional training, the human ear can provide more detailed information about the content of the signal: it can identify specific notes (i.e., frequencies), and can detect whether an instrument is out of tune.

Of course, the human auditory system cannot yield quantitative measures of signal parameters, nor can it process anything other than acoustic signals in the audible frequency band (from approximately 20 Hz to 20 kHz). (Analogous statements can be made about the human eye, which is particularly adept at identifying visual patterns.) *Signal processing*, which is the quantitative analysis of signals, allows us to identify and separate essential components of a signal based on its numerical samples (rather than its perceptual attributes). In many cases, these tasks can be carried out automatically, i.e., with little or no human intervention.

Sinusoids comprise a particularly important class of signals, since they arise in a diverse class of physical systems which exhibit linear properties. Such systems were introduced in Chapter 2 and will be the focus of our discussion in Chapter 4. In many applications, it is desirable to identify and isolate certain sinusoidal components present in a signal. For example, in restoring a musical recording that has been corrupted by static noise or deterioration of the recording medium, one would be interested in identifying the note being played during a noisy segment of the piece. A note consists of sinusoids at multiples of a certain fundamental frequency; identifying those sinusoids would be an essential step towards removing the noise from the signal. Sinusoids could also represent unwanted components of a signal, e.g., interference from a power supply at 50 Hz.

The discrete Fourier transform is well-suited for the detection of sinusoids in a discrete-time signal, since it provides a decomposition of the signal into sinusoidal components. For an N-point signal, these components are at the so-called Fourier frequencies, which are the integer multiples of $2\pi/N$.

First, let us consider a vector consisting of N uniform samples of a real-valued sinusoid of frequency Ω_0 rad/sec, obtained at a rate of f_s samples per second. If $\omega_0 = \Omega_0/f_s$ *happens to be* a Fourier frequency for sample size N, then the N-point DFT will will contain exactly two nonzero entries at frequencies ω_0 and $2\pi - \omega_0$, the remaining entries being zero. The frequency ω_0 (hence also Ω_0) can be determined by inspection of the DFT, provided that no aliasing has occurred. Also, the amplitude and phase of the sinusoid can be easily obtained from the corresponding entries of the DFT vector, and the continuous-time signal can be reconstructed perfectly.

Example 3.8.1. The continuous-time sinusoid

$$x(t) = 5\cos(600\pi t + 3\pi/4) \ , \qquad t \in \mathbf{R}$$

is sampled at a rate of 1,000 samples/sec to yield the discrete-time sinusoid

$$x[n] = 5\cos(0.6\pi n + 3\pi/4) \ , \qquad n \in \mathbf{N}$$

If $\mathbf{s} = x[0 : N-1]$ (using MATLAB notation for indices), then $\omega_0 = 0.6\pi$ (i.e., the frequency of $x[\cdot]$) is a Fourier frequency for $\mathbf{s}$ provided

$$k\frac{2\pi}{N} = 0.6\pi$$

or equivalently,

$$N = \frac{10k}{3}$$

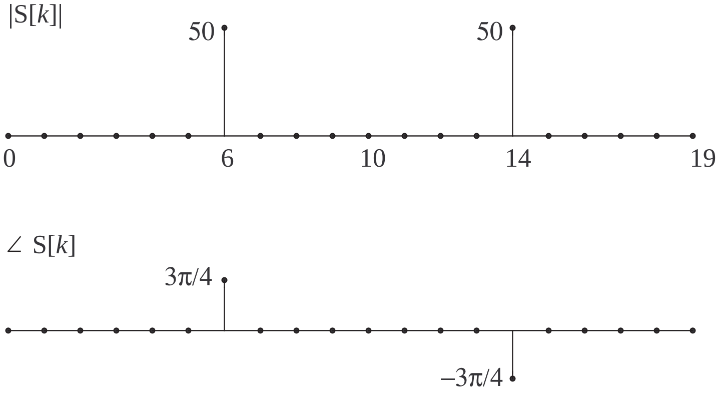
for some integer k. For example, if $N = 20$, then ω_0 is the $k = 6^{\text{th}}$ Fourier frequency for $\mathbf{s}$. The resulting amplitude and phase spectra are shown in the figure. □

In general, the frequency ω_0 will *not* be an exact Fourier frequency. In that case, the DFT will analyze the sinusoidal signal into N complex sinusoidal components at frequencies unrelated to ω_0, and as a result, it will not exhibit the features seen in the graphs of Example 3.8.1. The question is whether we can still use a DFT-based method to estimate the frequency ω_0. The answer is affirmative.

3.8.2 The DFT of a Complex Sinusoid

Let $\mathbf{s}$ be the L-point complex sinusoid given by

$$s[n] = e^{j\omega_0 n} \ , \qquad n = 0, \ldots, L-1$$



Example 3.8.1

where ω_0 is a fixed frequency. Let $\mathbf{v}$ represent a complex sinusoid of variable frequency ω:

$$v[n] = e^{j\omega n} \ , \qquad n = 0, \ldots, L-1$$

The inner product $\langle \mathbf{v}, \mathbf{s} \rangle$ is a function of ω. Its value at $\omega = 2k\pi/L$ (where k is integer) can be computed as the the k^{th} entry in the DFT of $\mathbf{s}$:

$$\langle \mathbf{v}, \mathbf{s} \rangle = \sum_{n=0}^{L-1} v^*[n]s[n] = \sum_{n=0}^{L-1} s[n]e^{-j(2\pi/L)kn} = S[k]$$

Similarly, the inner product $\langle \mathbf{v}, \mathbf{s} \rangle$ at $\omega = 2k\pi/N$, where $N > L$, can be computed via the DFT of $\mathbf{s}$ *zero-padded* to total length N. As N increases, the spacing $2\pi/N$ between consecutive Fourier frequencies decreases to zero, and the resulting DFT provides a very dense sampling of $\langle \mathbf{v}, \mathbf{s} \rangle$ over the entire frequency range $[0, 2\pi)$.

Let us evaluate $\langle \mathbf{v}, \mathbf{s} \rangle$ at frequency ω. (This was done in Subsection 3.1.3 for the special case where both ω and ω_0 are Fourier frequencies, in order to establish the orthogonality of the complex sinusoids.)

For $\omega = \omega_0$, we have $\mathbf{v} = \mathbf{s}$ and thus

$$\langle \mathbf{v}, \mathbf{s} \rangle = \|s\|^2 = L$$

For $\omega \neq \omega_0$, we have

$$\langle \mathbf{v}, \mathbf{s} \rangle \ = \ \sum_{n=0}^{L-1} v^*[n]s[n]$$

$$
\begin{aligned}
&= \sum_{n=0}^{L-1} e^{-j\omega n} e^{j\omega_0 n} \\
&= \sum_{n=0}^{L-1} e^{j(\omega_0-\omega)n} \\
&= \frac{1-e^{jL(\omega_0-\omega)}}{1-e^{j(\omega_0-\omega)}}
\end{aligned}
$$

where the last equality is the familiar expression for the geometric sum. (Note that the denominator cannot be equal to zero, since $\omega \neq \omega_0$ and both frequencies are assumed to be in $[0, 2\pi)$.) The inner product is therefore complex. It can be written as the product of a real term and a complex term of unit magnitude by first applying the identity $1 - z^2 = z(z^{-1} - z)$ to both numerator and denominator:

$$\frac{1-e^{jL(\omega_0-\omega)}}{1-e^{j(\omega_0-\omega)}} = \frac{e^{jL(\omega_0-\omega)/2}}{e^{j(\omega_0-\omega)/2}} \cdot \frac{e^{-jL(\omega_0-\omega)/2} - e^{jL(\omega_0-\omega)/2}}{e^{-j(\omega_0-\omega)/2} - e^{j(\omega_0-\omega)/2}}$$

and then recalling that $e^{j\theta} - e^{-j\theta} = 2j\sin\theta$. The resulting expression is

$$
\begin{aligned}
\langle \mathbf{v}, \mathbf{s} \rangle &= e^{-j(L-1)(\omega-\omega_0)/2} \cdot \frac{\sin(L(\omega-\omega_0)/2)}{\sin((\omega-\omega_0)/2)} \\
&= e^{-j(L-1)(\omega-\omega_0)/2} \cdot \mathcal{D}_L(\omega-\omega_0) \qquad (3.17)
\end{aligned}
$$

where the function $\mathcal{D}_L(\cdot)$ is defined by

$$\mathcal{D}_L(\theta) = \frac{\sin(L\theta/2)}{\sin(\theta/2)}$$

Since $\left|e^{-j(L-1)(\omega-\omega_0)/2}\right| = 1$, we have that

$$|\langle \mathbf{v}, \mathbf{s} \rangle| = |\mathcal{D}_L(\omega-\omega_0)| = \left|\frac{\sin(L(\omega-\omega_0)/2)}{\sin((\omega-\omega_0)/2)}\right| \qquad (3.18)$$

Evaluated at $\omega = 2k\pi/L$, the above expression gives the k^{th} entry in the *amplitude* spectrum of $\mathbf{s}$, i.e., $|S[k]|$. Evaluated at $\omega = 2k\pi/N$ (where $N > L$), it yields the k^{th} entry in the amplitude spectrum of the N-point zero-padded extension of $\mathbf{s}$.

The absolute value of the function $\mathcal{D}_L(\theta)$ is shown in Figure 3.10 for $L = 6$ and $L = 9$, in each case for θ varying over $(-2\pi, 2\pi)$ (which includes the range of values of $\omega - \omega_0$ in (3.17)).

We make the following key observations about the function $\mathcal{D}_L(\theta)$:

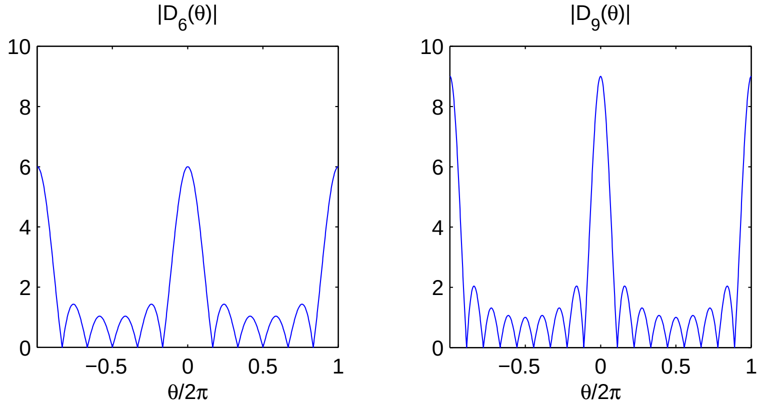


Figure 3.10: The function $|\mathcal{D}_L(\theta)|$ for $L = 6$ (left) and $L = 9$ (right).

- At $\theta = 0$, both the numerator and the denominator in the definition of $\mathcal{D}_L(\theta)$ equal 0. $\mathcal{D}_L(0)$ is then defined as the limit of $\mathcal{D}_L(\theta)$ as θ approaches zero, which equals L. This also gives the correct result for the inner product when $\omega = \omega_0$.

- $\mathcal{D}_L(\theta)$ is periodic with period 2π.

- $\mathcal{D}_L(\theta) = 0$ for all values of θ which are integer multiples of $2\pi/L$, except those which are also integer multiples of 2π ($\mathcal{D}_L(2\pi k) = L$).

- In each period, the graph of $\mathcal{D}_L(\theta)$ contains one *main lobe* of width $4\pi/L$ and height L; and $L-2$ *side lobes* of width $2\pi/L$ and varying height. The first side lobe is the tallest one (in absolute value), its height being approximately $2/3\pi$ that of the main one.

We conclude this lengthy discussion with the following observation.

Fact. *Let ω_0 be a fixed frequency in $[0, 2\pi)$ and ω be a variable frequency in the same interval. Then the magnitude of the inner product $\langle \mathbf{v}, \mathbf{s} \rangle$, where*

$$s[n] = e^{j\omega_0 n} \quad \text{and} \quad v[n] = e^{j\omega n} \ , \qquad n = 0, \ldots, L-1$$

achieves its unique maximum (as ω varies) when $\omega = \omega_0$. Thus if the frequency ω_0 of a given complex sinusoid $\mathbf{s}$ is not known, it can be estimated with arbitrary accuracy by locating the maximum value of $|\langle \mathbf{v}, \mathbf{s} \rangle|$, computed for a sufficiently dense set of frequencies ω in $[0, 2\pi)$.

This fact follows from (3.18):

$$|\langle \mathbf{v}, \mathbf{s} \rangle| = |\mathcal{D}_L(\omega - \omega_0)|$$

For fixed ω_0, the difference $\theta = \omega - \omega_0$ lies in the interval $[2\pi - \omega_0,\ -\omega_0)$. This is a subinterval of $(-2\pi,\ 2\pi)$ which includes the origin. The maximum value of $|\mathcal{D}_L(\theta)|$ over that interval will be achieved at $\theta = 0$, as illustrated in Figure 3.10. Thus the maximum value of $|\langle \mathbf{v}, \mathbf{s} \rangle| = |\mathcal{D}_L(\omega - \omega_0)|$ occurs at $\omega = \omega_0$, and equals L.

3.8.3 Detection of a Real-Valued Sinusoid

As we showed in the previous subsection, the frequency of a L-point complex-valued sinusoid

$$s[n] = e^{j\omega_0 n}\ , \qquad n = 0, \ldots, L-1$$

can be determined by zero-padding the signal to obtain a reasonably dense set of Fourier frequencies in $[0, 2\pi)$, then locating the maximum of the amplitude spectrum.

A real-valued sinusoid is the sum of two complex-valued sinusoids at conjugate frequencies:

$$A\cos(\omega_0 n + \phi) = \frac{A}{2} e^{j(\omega_0 n + \phi)} + \frac{A}{2} e^{-j(\omega_0 n + \phi)}$$

We assume that ω_0 lies in $[0, \pi]$, which is the effective frequency range for real sinusoids. In most practical situations, the observed signal vector will have other components, as well. We thus model it as

$$x[n] = A\cos(\omega_0 n + \phi) + r[n]\ , \qquad n = 0, \ldots, L-1$$

where the remaining components of $\mathbf{x}$ are collectively represented by $\mathbf{r}$. We then have

$$\mathbf{x} = \frac{A}{2} e^{j\phi} \mathbf{s} + \frac{A}{2} e^{-j\phi} \mathbf{s}^* + \mathbf{r}$$

We argue that a satisfactory estimate of frequency ω_0 can be obtained from $\mathbf{x}$ using the method developed earlier for the complex exponential $\mathbf{s}$. Again, let

$$v[n] = e^{j\omega n}\ , \qquad n = 0, \ldots, L-1$$

be the variable-frequency sinusoid used in the computation of the DFT, and consider the inner product

$$\langle \mathbf{v}, \mathbf{x} \rangle = \frac{A}{2} e^{j\phi} \langle \mathbf{v}, \mathbf{s} \rangle + \frac{A}{2} e^{-j\phi} \langle \mathbf{v}, \mathbf{s}^* \rangle + \langle \mathbf{v}, \mathbf{r} \rangle$$

which yields the DFT of $\mathbf{x}$ and its zero-padded extensions by appropriate choice of ω.

We have already studied the behavior of the first two terms on the right-hand side. In particular, we know that

$$|\langle \mathbf{v}, \mathbf{s} \rangle| = |\mathcal{D}_L(\omega - \omega_0)| = \left| \frac{\sin(L(\omega - \omega_0)/2)}{\sin((\omega - \omega_0)/2)} \right|$$

and

$$|\langle \mathbf{v}, \mathbf{s}^* \rangle| = |\mathcal{D}_L(\omega + \omega_0)| = \left| \frac{\sin(L(\omega + \omega_0)/2)}{\sin((\omega + \omega_0)/2)} \right|$$

As long as ω_0 is not too close to 0 or π, the side lobes of $|\langle \mathbf{v}, \mathbf{s}^* \rangle|$ will not interfere significantly with the main lobe of $|\langle \mathbf{v}, \mathbf{s} \rangle|$, and the location of the maximum of

$$|\langle \mathbf{v}, \mathbf{s} \rangle + \langle \mathbf{v}, \mathbf{s}^* \rangle|$$

over the range $[0, \pi]$ will be negligibly different from ω_0.

If, similarly, $|\langle \mathbf{v}, \mathbf{r} \rangle|$ does not vary significantly near ω_0, where "significant" is understood relatively to the height and curvature of the main lobe of $(A/2)|\langle \mathbf{v}, \mathbf{s} \rangle|$, then the maximum of $|\langle \mathbf{v}, \mathbf{x} \rangle|$ over the range $[0, \pi]$ will occur very close to the unknown frequency ω_0.

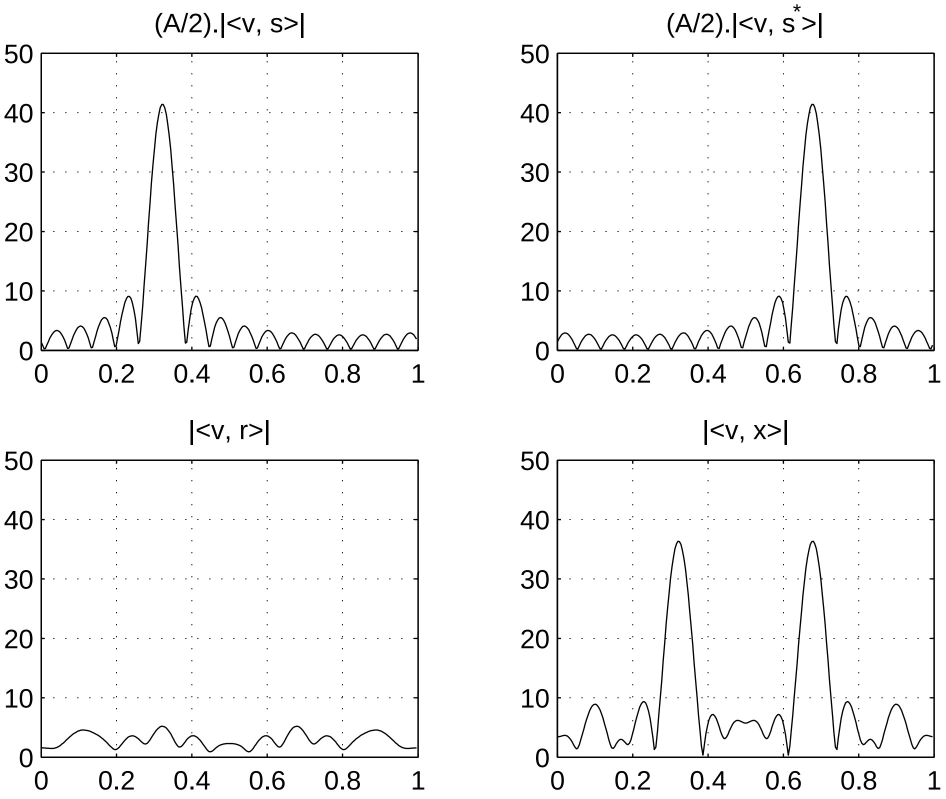
Example 3.8.2. Consider the 16-point signal

$$x[n] = 5.12 \cos(2\pi(0.3221)n + 1.39) + r[n]$$

Here $r[n]$ is white Gaussian noise with mean zero and standard deviation 0.85 (roughly a sixth of the amplitude of the sinusoid). In the current notation, we have $A = 5.12$, $\omega_0 = 2\pi(0.3221)$ and $\phi = 1.39$.

The figure shows plots of $(A/2)|\langle \mathbf{v}, \mathbf{s} \rangle|$, $(A/2)|\langle \mathbf{v}, \mathbf{s}^* \rangle|$, $|\langle \mathbf{v}, \mathbf{r} \rangle|$ and $|\langle \mathbf{v}, \mathbf{x} \rangle|$, all computed using 256-point DFT's, against cyclic frequency $f = \omega/2\pi$ (cycles/sample). Note that $|\langle \mathbf{v}, \mathbf{r} \rangle|$ and $|\langle \mathbf{v}, \mathbf{x} \rangle|$ are symmetric about $\omega = \pi$ (or $f = 1/2$), since $\mathbf{x}$ and $\mathbf{r}$ are real-valued vectors. Both $|\langle \mathbf{v}, \mathbf{s} \rangle|$ and $|\langle \mathbf{v}, \mathbf{x} \rangle|$ achieve their maximum over the range $0 \leq \omega \leq \pi$ at $\omega = 2\pi(0.3202)$, which corresponds to frequency index $k = 82$ in the 256-point DFT. Additional zero-padding would give an estimate closer to the true value $2\pi(0.3221)$. □

Conclusion. If a signal vector $\mathbf{x}$ has a single strong sinusoidal component, then the frequency of that component can be estimated reasonably accurately from the position of the maximum in the left half of the amplitude spectrum, where the DFT is computed after padding the signal with sufficiently many zeros.



Example 3.8.2

3.8.4 Final Remarks

The methodology discussed above can be extended to a signal containing two or more strong sinusoidal components. Simply put, as long as these components are well-separated on the frequency axis, their (unknown) frequencies can be estimated well by locating maxima on the left half of the amplitude spectrum. Good separation in frequency typically means that the main lobes corresponding to different components are sufficiently far apart. Since the main lobe width is $4\pi/L$, the only way to obtain sharper peaks is to increase L, i.e., take more samples of the signal. *Increasing the number of points N in the DFT does not solve the problem, since lobe width does not depend on N.*

Finally, we should note that it is possible to improve the detection and frequency estimation technique outlined above (in the case where multiple sinusoids may be present) by scaling the values of the data vector $\mathbf{x}$ using certain standard time-domain functions (known as *windows*) before computing the DFT. As a result of this preprocessing, the height ratio between the main lobe and the side lobes is increased by orders of magnitude, at the expense of only a moderate increase in main lobe width. Such techniques provide better resolution for sinusoids that are close in frequency.

Problems

Section 3.1

***P* 3.1.** Let

$$\alpha = \frac{1}{2} \qquad \text{and} \qquad \beta = \frac{\sqrt{3}}{2}$$

(i) Determine a complex number z such that the vector

$$\mathbf{v} = \begin{bmatrix} 1 & \alpha + j\beta & -\alpha + j\beta & -1 & -\alpha - j\beta & \alpha - j\beta \end{bmatrix}^T$$

equals

$$\begin{bmatrix} 1 & z & z^2 & z^3 & z^4 & z^5 \end{bmatrix}^T$$

(ii) If

$$\mathbf{s} = \begin{bmatrix} 3 & 2 & -1 & 0 & -1 & 2 \end{bmatrix}^T$$

determine the least-squares approximation $\hat{\mathbf{s}}$ of $\mathbf{s}$ in the form of a linear combination of $\mathbf{1}$ (i.e., the all-ones vector), $\mathbf{v}$ and $\mathbf{v}^*$. Clearly show the numerical values of the elements of $\hat{\mathbf{s}}$.

***P* 3.2.** Let $\mathbf{v}^{(0)}$, $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(7)}$ denote the complex Fourier sinusoids of length $N = 8$ at frequencies $\omega = 0$, $\omega = \pi/4$ and $\omega = 7\pi/4$, respectively.

Determine the least-squares approximation $\hat{\mathbf{s}}$ of

$$\mathbf{s} = \begin{bmatrix} 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 \end{bmatrix}$$

based on $\mathbf{v}^{(0)}$, $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(7)}$. Compute the squared approximation error $\|\hat{\mathbf{s}} - \mathbf{s}\|^2$.

***P* 3.3.** Consider the eight-point vector

$$\mathbf{x} = \begin{bmatrix} 2\sqrt{2} - 1 & 1 & -2\sqrt{2} - 1 & 5 & -2\sqrt{2} - 1 & 1 & 2\sqrt{2} - 1 & -3 \end{bmatrix}^T$$

(i) Show that $\mathbf{x}$ contains only three Fourier frequencies, i.e., it is the linear combination of exactly three Fourier sinusoids $\mathbf{v}^{(k)}$. Determine the coefficients of these sinusoids in the expression for $\mathbf{x}$.

(ii) Find an equivalent expression for $\mathbf{x}$ in terms of real-valued sinusoids of the form $\cos(\omega n + \phi)$, where $0 \le \omega \le \pi$.

***P* 3.4.** The columns of the matrix

$$\mathbf{V} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

are the complex Fourier sinusoids of length $N = 4$.

Express the vector

$$\mathbf{s} = \begin{bmatrix} 1 & 4 & -2 & 5 \end{bmatrix}^T$$

as a linear combination of the above sinusoids. In other words, find a vector $\mathbf{c} = [c_0\ c_1\ c_2\ c_3]^T$ such that $\mathbf{s} = \mathbf{Vc}$.

Section 3.2

***P* 3.5.** Let $u = e^{j(2\pi/9)}$ and $z = e^{j(\pi/5)}$.

(i) Write out the entries of the DFT matrix $\mathbf{W}_9$ (corresponding to a nine-point signal) using the real number 1 and complex numbers u^m, where m is a nonzero integer between -4 and 4.

(ii) Write out the entries of the IDFT matrix $\mathbf{V}_{10}$ (corresponding to a ten-point signal) using real numbers 1, -1 and complex numbers z^m, where m is a nonzero integer between -4 and 4.

***P* 3.6.** **(i)** Sketch the six-point vectors $\mathbf{x}$ and $\mathbf{y}$ defined by

$$\begin{aligned} x[n] &= \begin{cases} 1, & n = 2, 4 \\ 0, & \text{otherwise;} \end{cases} \\ y[n] &= \cos(2\pi n/3) \end{aligned}$$

(ii) Prove that

$$\mathbf{y} = \frac{\mathbf{v}^{(2)} + \mathbf{v}^{(4)}}{2}$$

where $\mathbf{v}^{(k)}$ is the k^{th} Fourier sinusoid for $N = 6$.

(iii) Compute the DFT's $\mathbf{X}$ and $\mathbf{Y}$.

Observe the similarities between $\mathbf{x}$ *and* $\mathbf{Y}$*, as well as between* $\mathbf{y}$ *and* $\mathbf{X}$*.*

Section 3.3

***P* 3.7.** A five-point *real-valued* signal $\mathbf{x}$ has DFT given by

$$\mathbf{X} = \begin{bmatrix} 4 & 1+j & 3-j & z_1 & z_2 \end{bmatrix}^T$$

(i) Compute $x[0]+x[1]+x[2]+x[3]+x[4]$ using one entry of $\mathbf{X}$ *only.*

(ii) Determine the values of z_1 and z_2.

(iii) Compute the amplitude and phase spectra of $\mathbf{x}$, displaying each as a vector.

(iv) Express $x[n]$ as a linear combination of three real-valued sinusoids.

***P* 3.8.** Consider the real-valued signal $\mathbf{x}$ given by

$$x[n] = 3(-1)^n + 7\cos\left(\frac{\pi n}{4} + 1.2\right) + 2\cos\left(\frac{\pi n}{2} - 0.8\right), \qquad n = 0, \ldots, 7$$

(i) Which Fourier frequencies (for an eight-point sample) are present in the signal $\mathbf{x}$?

(ii) Determine the amplitude spectrum of $\mathbf{x}$, displaying your answer in the form

$$\begin{bmatrix} A_0 & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 \end{bmatrix}^T$$

(iii) Determine the phase spectrum of $\mathbf{x}$, displaying your answer in the form

$$\begin{bmatrix} \phi_0 & \phi_1 & \phi_2 & \phi_3 & \phi_4 & \phi_5 & \phi_6 & \phi_7 \end{bmatrix}^T$$

***P* 3.9.** Consider the N-point time-domain signal $\mathbf{x}$ given by

$$x[n] = r^n, \qquad n = 0, \ldots, N-1$$

where r is a real number other than -1 or $+1$.

(i) Using the formula for the geometric sum, show that the DFT $\mathbf{X}$ of $\mathbf{x}$ is given by

$$X[k] = \frac{1-r^N}{1-rw^k}, \qquad k = 0, \ldots, N-1$$

where, as usual, $w = e^{-j(2\pi/N)}$.

(ii) Using the formula $|z|^2 = zz^*$, show that the square amplitude spectrum is given by

$$|X[k]|^2 = \frac{(1-r^N)^2}{1+r^2-2r\cos(2k\pi/N)}, \qquad k = 0, \ldots, N-1$$

(iii) For the case $r = 0.7$ and $N = 16$, use MATLAB to produce a (discrete) plot of the square amplitude spectrum in the equation above. Compare your answer to

```
N=16; r = 0.7;
n=(0:N-1)’;
x = r.^n;
A = abs(fft(x));
bar(n,A.^2)
```

Section 3.4

***P* 3.10.** Let

$$\mathbf{x} = \begin{bmatrix} 2 & 1 & -1 & -2 & -3 & 3 \end{bmatrix}^T$$

Display (as vectors) and sketch the following signals:

- $\mathbf{x}^{(1)} = \mathbf{P}\mathbf{x}$
- $\mathbf{x}^{(2)} = \mathbf{P}^5\mathbf{x}$
- $\mathbf{x}^{(3)} = \mathbf{P}\mathbf{x} + \mathbf{P}^5\mathbf{x}$
- $\mathbf{x}^{(4)} = \mathbf{x} + \mathbf{R}\mathbf{x}$ (*Note the symmetry.*)
- $\mathbf{x}^{(5)} = \mathbf{x} - \mathbf{R}\mathbf{x}$ (*Note the symmetry.*)
- $\mathbf{x}^{(6)} = \mathbf{F}^3\mathbf{x}$
- $\mathbf{x}^{(7)} = (\mathbf{I} - \mathbf{F}^3)\mathbf{x}$

***P* 3.11.** The time-domain signal

$$\mathbf{x} = \begin{bmatrix} a & b & c & d & e & f \end{bmatrix}^T$$

has DFT

$$\mathbf{X} = \begin{bmatrix} A & B & C & D & E & F \end{bmatrix}^T$$

Using the given parameters, and defining

$$\lambda = \frac{1}{2} \qquad \text{and} \qquad \mu = \frac{\sqrt{3}}{2}$$

for convenience, write out the components of the DFT vector $\mathbf{X}^{(r)}$ for each of the following time-domain signals $\mathbf{x}^{(r)}$.

$$\begin{aligned}
\mathbf{x}^{(1)} &= \begin{bmatrix} a & -b & c & -d & e & -f \end{bmatrix}^T \\
\mathbf{x}^{(2)} &= \begin{bmatrix} a & 0 & c & 0 & e & 0 \end{bmatrix}^T \\
\mathbf{x}^{(3)} &= \begin{bmatrix} a & f & e & d & c & b \end{bmatrix}^T \\
\mathbf{x}^{(4)} &= \begin{bmatrix} d & e & f & a & b & c \end{bmatrix}^T \\
\mathbf{x}^{(5)} &= \begin{bmatrix} b & a & f & e & d & c \end{bmatrix}^T \\
\mathbf{x}^{(6)} &= \begin{bmatrix} f+b & a+c & b+d & c+e & d+f & e+a \end{bmatrix}^T \\
\mathbf{x}^{(7)} &= \begin{bmatrix} 0 & \mu b & \mu c & 0 & -\mu e & -\mu f \end{bmatrix}^T \\
\mathbf{x}^{(8)} &= \begin{bmatrix} A & B & C & D & E & F \end{bmatrix}^T
\end{aligned}$$

***P* 3.12.** A *real-valued* eight-point signal vector

$$\mathbf{x} = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{bmatrix}^T$$

has DFT $\mathbf{X}$ given by

$$\mathbf{X} = \begin{bmatrix} 4 & 5-j & -1+3j & -2 & -7 & S_5 & S_6 & S_7 \end{bmatrix}^T$$

Without inverting $\mathbf{X}$, compute the numerical values in the DFT $\mathbf{Y}$ of

$$\mathbf{y} = \begin{bmatrix} 2x_0 & x_1+x_7 & x_2+x_6 & x_3+x_5 & 2x_4 & x_5+x_3 & x_6+x_2 & x_7+x_1 \end{bmatrix}^T$$

***P* 3.13.** An eight-point signal $\mathbf{x}$ has DFT

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \end{bmatrix}^T$$

Without inverting $\mathbf{X}$, compute the DFT $\mathbf{Y}$ of $\mathbf{y}$, which is given by the equation

$$y[n] = x[n] \cdot \cos\left(\frac{\pi n}{4}\right) , \qquad n = 0, \ldots, 7$$

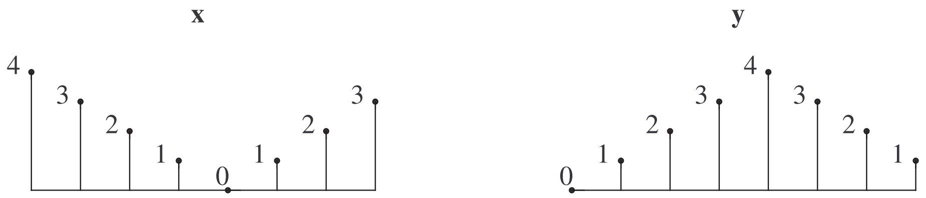
Section 3.5

***P* 3.14.** Consider the signal $\mathbf{x}$ shown in the figure (on the left). Its spectrum is given by

$$\mathbf{X} = \begin{bmatrix} A & B & C & D & E & F & G & H \end{bmatrix}^T$$

(i) The DFT vector shown above contains duplicate values. What are those values?

(ii) Express the DFT $\mathbf{Y}$ of the signal $\mathbf{y}$ (shown on the right) in terms of the entries of $\mathbf{X}$.

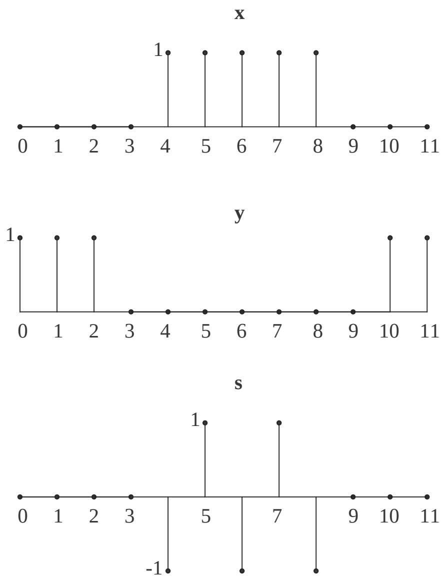


Problem P 3.14

***P* 3.15.** Consider the twelve-point vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{s}$ shown in the figure. If the DFT $\mathbf{X}$ is given by

$$\mathbf{X} = \begin{bmatrix} X_0 & X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 & X_8 & X_9 & X_{10} & X_{11} \end{bmatrix}^T$$

express the DFT's $\mathbf{Y}$ and $\mathbf{S}$ in terms of the entries of $\mathbf{X}$.



Problem P 3.15

***P* 3.16.** Run the MATLAB script

```
n = (0:63)’;
X =[ones(11,1); zeros(43,1); ones(10,1)];
bar(X), axis tight
max(imag(ifft(X)))      % See (i) below
x = real(ifft(X));
bar(x)
cs = cos(3*pi*n/4);     % See (ii) below
y = x.*cs;
bar(y);
max(imag(fft(y)))       % See (iii) below
Y = real(fft(y));
bar(Y)                  % See (iv) below
```

(i) Why is this value so small?
(ii) Is this a Fourier sinusoid for this problem?
(iii) Why is this value so small?
(iv) Derive the answer for `Y` analytically, i.e., based on known properties of the DFT.

***P* 3.17.** The *energy spectrum* of the signal vector $\mathbf{x}$ is defined as the square of the amplitude spectrum, namely $|X[k]|^2$ (for $k = 0, \ldots, N-1$).

Show that the total energy of the signal vector $\mathbf{x}$ equals the sum of the entries in the energy spectrum divided by N, i.e.,

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2$$

This relationship, known as *Parseval's identity*, can be also written as

$$\|\mathbf{x}\|^2 = \frac{1}{N} \|\mathbf{X}\|^2$$

(and is easier to prove in this form). In geometric terms, this result is consistent with the fact that the length of a vector can be computed (via the Pythagorean theorem) using the projections of that vector on *any* orthornormal set of reference vectors.

Section 3.6

***P* 3.18.** Determine the circular convolution $\mathbf{s} = \mathbf{x} \circledast \mathbf{y}$, where

$$\mathbf{x} = \left[\begin{array}{cccc} 1 & 2 & 3 & 4 \end{array}\right]^T \qquad \text{and} \qquad \mathbf{y} = \left[\begin{array}{cccc} a & b & c & d \end{array}\right]^T$$

Also, express $\mathbf{s}$ as $\mathbf{My}$, where $\mathbf{M}$ is a 4×4 matrix of numerical values.

***P* 3.19.** Consider two time-domain vectors $\mathbf{x}$ and $\mathbf{y}$ whose DFT's are given by

$$\mathbf{X} = \left[\begin{array}{cccc} 1 & -2 & 3 & -4 \end{array}\right]^T \qquad \text{and} \qquad \mathbf{Y} = \left[\begin{array}{cccc} A & B & C & D \end{array}\right]^T$$

Without explicitly computing $\mathbf{x}$ or $\mathbf{y}$, determine the DFT of their element-wise product

$$s[n] = x[n]y[n], \qquad n = 0, 1, 2, 3$$

***P* 3.20.** The time-domain signals $\mathbf{x}$ and $\mathbf{y}$ have DFT's

$$\mathbf{X} = \left[\begin{array}{cccc} 1 & 0 & 1 & -1 \end{array}\right]^T$$

and

$$\mathbf{Y} = \left[\begin{array}{cccc} 3 & 5 & 8 & -4 \end{array}\right]^T$$

(i) Is either $\mathbf{x}$ or $\mathbf{y}$ real-valued?

(ii) Does either $\mathbf{x}$ or $\mathbf{y}$ have circular conjugate symmetry?

(iii) Without inverting $\mathbf{X}$ or $\mathbf{Y}$, determine the DFT of the signal $\mathbf{s}^{(1)}$ defined by

$$s^{(1)}[n] = x[n]y[n] \ , \qquad n = 0, 1, 2, 3$$

(iv) Without inverting $\mathbf{X}$ or $\mathbf{Y}$, determine the DFT of the signal $\mathbf{s}^{(2)}$ defined by

$$\mathbf{s}^{(2)} = \mathbf{x} \circledast \mathbf{y}$$

***P* 3.21.** The time-domain signals

$$\mathbf{x} = \begin{bmatrix} 2 & 0 & 1 & 3 \end{bmatrix}^T$$

and

$$\mathbf{y} = \begin{bmatrix} 1 & -1 & 2 & -4 \end{bmatrix}^T$$

have DFT's $\mathbf{X}$ and $\mathbf{Y}$ given by

$$\mathbf{X} = \begin{bmatrix} X_0 & X_1 & X_2 & X_3 \end{bmatrix}^T$$

and

$$\mathbf{Y} = \begin{bmatrix} Y_0 & Y_1 & Y_2 & Y_3 \end{bmatrix}^T$$

Determine the time-domain signal $\mathbf{s}$ whose DFT is given by

$$\mathbf{S} = \begin{bmatrix} X_0Y_2 & X_1Y_3 & X_2Y_0 & X_3Y_1 \end{bmatrix}^T$$

***P* 3.22.** The *circular cross-correlation* of two N-point vectors $\mathbf{x}$ and $\mathbf{y}$ is the N-point vector $\mathbf{s}$ defined by

$$s[n] = \langle \mathbf{P}^n\mathbf{y}, \mathbf{x} \rangle = \mathbf{x}^T\mathbf{P}^n\mathbf{y}^* \ , \qquad n = 0, \ldots, N-1$$

(i) Show that $\mathbf{s}$ also equals $\mathbf{x} \circledast \mathbf{R}\mathbf{y}^*$.

(ii) Using **(i)** above, show that the DFT $\mathbf{S}$ of $\mathbf{s}$ is given by

$$S[k] = X[k]Y^*[k], \qquad k = 0, \ldots, N-1$$

(iii) Show that in the special case where $\mathbf{x} = \mathbf{y}$, $\mathbf{S}$ is just the energy spectrum of $\mathbf{x}$, i.e.,

$$S[k] = |X[k]|^2, \qquad k = 0, \ldots, N-1$$

What symmetry properties does $\mathbf{s}$ have in this case?

(iv) Compute **s** and **S** for

$$\mathbf{x} = \mathbf{y} = \begin{bmatrix} 1+j & 2 & 1-j & 0 \end{bmatrix}^T$$

Section 3.7

***P* 3.23.** The signal

$$\mathbf{x} = \begin{bmatrix} a & b & c & d & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

has DFT **X** given by

$$\mathbf{X} = \begin{bmatrix} A & B & C & D & E & F & G & H \end{bmatrix}^T$$

Express the following DFT's in terms of the entries of **X**:
(i) The DFT **Y** of

$$\mathbf{y} = \begin{bmatrix} 0 & 0 & 0 & 0 & a & b & c & d \end{bmatrix}^T$$

(ii) The DFT **S** of

$$\mathbf{s} = \begin{bmatrix} a & b & c & d & a & b & c & d \end{bmatrix}^T$$

***P* 3.24.** The DFT of the signal

$$\mathbf{x} = \begin{bmatrix} 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

is given by

$$\mathbf{X} = \begin{bmatrix} X_0 & X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 \end{bmatrix}^T$$

(i) What are the values of X_0, X_2, X_4 and X_6?
(ii) Display the time-domain signal **y** whose DFT is given by

$$\mathbf{Y} = \begin{bmatrix} X_0 & 0 & 0 & X_2 & 0 & 0 & X_4 & 0 & 0 & X_6 & 0 & 0 \end{bmatrix}^T$$

***P* 3.25.** **(i)** The signal **x** has spectrum (DFT)

$$\mathbf{X} = [A \quad 0 \quad 0 \quad B \quad 0 \quad 0 \quad C \quad 0 \quad 0 \quad D \quad 0 \quad 0]^T$$

What special properties does **x** have?
(ii) The signal **y** has spectrum

$$\mathbf{Y} = [A \quad B \quad C \quad D \quad A \quad B \quad C \quad D \quad A \quad B \quad C \quad D]^T$$

What special properties does **y** have?

***P* 3.26.** The twelve-point signal

$$\mathbf{x} = \left[\begin{array}{cccccccccccc} a & b & c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]^T$$

has DFT $\mathbf{X}$ given by

$$\mathbf{X} = \left[\begin{array}{cccccccccccc} X_0 & X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 & X_8 & X_9 & X_{10} & X_{11} \end{array}\right]^T$$

Express the following DFT's in terms of the entries of $\mathbf{X}$:

(i) The DFT $\mathbf{Y}$ of

$$\mathbf{y} = \left[\begin{array}{cccccc} a & b & c & 0 & 0 & 0 \end{array}\right]^T$$

(ii) The DFT $\mathbf{S}$ of

$$\mathbf{s} = \left[\begin{array}{cccccccccccc} a & b & c & a & b & c & a & b & c & a & b & c \end{array}\right]^T$$

***P* 3.27.** In MATLAB notation, consider the 4-point vector

```
s = [a b c d].’
```

and its zero-padded extension

```
x = [s ; zeros(12,1)]
```

Let `X` denote the DFT of `x`. Express the DFT's of the following vectors using the entries of `X`:

```
x1 = s
x2 = [ s ; s ]
x3 = [ s ; s ; s ; s ; s ]  % length=20
x4 = [ s ; z4 ]
x5 = [ z4 ; s ]
x6 = [ s ; z4 ; s ; z4 ]
x7 = [ s ; s ; z4 ; z4 ]
```

where `z4 = zeros(4,1)`.

***P* 3.28.** Consider a real-valued vector $\mathbf{s}$ of length L, and define a vector $\mathbf{x}$ of length $2L+1$ by

$$\mathbf{x} = \begin{bmatrix} a \\ \mathbf{s} \\ \mathbf{Qs} \end{bmatrix}$$

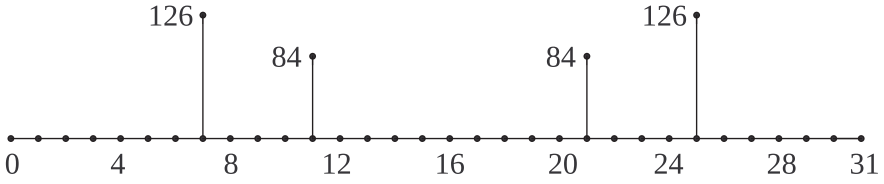
where $\mathbf{Q}$ is the *linear* reversal matrix and $a \in \mathbf{R}$.

Let $\mathbf{y}$ be the vector obtained by padding $\mathbf{s}$ with $L+1$ zeros. Show that the DFT $\mathbf{X}$ of $\mathbf{x}$ has the following expression in terms of the DFT $\mathbf{Y}$ of $\mathbf{y}$ and the complex number $w = e^{-j2\pi/(2L+1)}$:

$$X[k] = a + 2 \cdot \Re e \left\{ w^k Y[k] \right\} \ , \quad k = 0, \ldots, 2L$$

Section 3.8

***P* 3.29.** A continuous-time signal consists of two sinusoids at frequencies f_1 and f_2 (Hz). The signal is sampled at a rate of 500 samples/sec (assumed to be greater than the Nyquist rate), and 32 consecutive samples are recorded. The figure shows the graph of magnitude of the 32-point DFT (i.e., without zero-padding) as a function of the frequency index k.



Problem P 3.29

(i) What are the frequencies f_1 and f_2 (in Hz)?

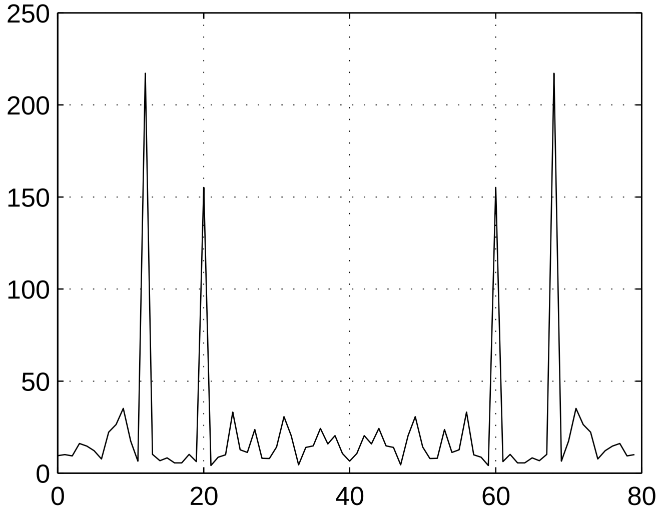
(ii) Is it possible to write an equation for the continuous-time signal based on the information given? If so, write that equation. If not so, explain why.

***P* 3.30.** A continuous-time signal is given by

$$x(t) = A_1 \cos(2\pi f_1 t + \phi_1) + A_2 \cos(2\pi f_2 t + \phi_2) + z(t)$$

where $z(t)$ is noise. The signal $x(t)$ is sampled every 1.25 ms for a total of 80 samples. The figure shows the graph of the DFT of the 80-point signal as a function of the frequency index $k = 0, \ldots, 79$.
Based on the given graph, what are your estimates of f_1 and f_2?

(You may assume that no aliasing has occurred, i.e., the sampling rate of 800 samples/sec is no less than twice each of f_1 and f_2.)



Problem P 3.30

P **3.31.** A continuous-time signal is a sum of two sinusoids at frequencies 164 Hz and 182 Hz. 200 samples of the signal are obtained at the rate of 640 samples/sec, and the DFT of the samples is computed.

(i) What frequencies $\omega_1 = 2\pi f_1$ and $\omega_2 = 2\pi f_2$ are present in the discrete-time signal obtained by sampling at the above rate?

(ii) Of the 200 Fourier frequencies in the DFT, which two are closest to $\omega_1 = 2\pi f_1$ and $\omega_2 = 2\pi f_2$?

(iii) If the 200 samples are padded with M zeros and the $(M + 200)$-point DFT is computed, what would be the least value of M for which both ω_1 and ω_2 are Fourier frequencies?

P **3.32.** A musical note is a pure sinusoid of a specific frequency known as *fundamental* frequency. Played on an instrument, notes are "colored" by the introduction of *harmonics*, namely sinusoids having frequencies which are exact multiples of the fundamental.

An instrument playing a 330 Hz note is recorded digitally over a time interval of duration 50 ms at a sampling rate of 46,200 samples/sec, yielding 2,310 samples. It is assumed that the sampling rate exceeds the Nyquist rate; this means that there are no harmonics at frequencies $k(330)$ Hz for $k \geq 70$.

(i) Does the fundamental frequency of 330 Hz correspond to a Fourier frequency for the 2,310-point sample vector? (Note that if it is a Fourier frequency, its harmonics will be Fourier frequencies also.) If it is not a Fourier frequency, what is the largest value of N less than or equal to 2,310 that would make it a Fourier frequency?

(ii) (MATLAB) The vector `s3` contains the 2,310 samples of the note, where each sample is distorted by a small amount of noise. Take the first N entries of that vector, where N was the answer to part **(i)**, and compute their DFT. Determine the total number of harmonics (positive frequencies only, and excluding the fundamental) which are within 40 dB of the fundamental i.e., the ratio of their amplitude to that of the fundamental is no less than 1%.

***P* 3.33.** The 64-point vector `s4` was generated in MATLAB using

```
n = (0:63).';
s4 = A*cos(2*pi*f1*n+q) + z;
```

The parameters `A`, `f1` (between 0 and 1/2) and `q` were previously specified. The vector `z`, which represents noise, was also generated earlier.

(i) Compute the DFT of `s4` extended by zero-padding to $N = 1000$ points:

```
X = fft(s4,1000)
```

Obtain an estimate of `f1` by locating the maximum value of `abs(X)`. Plot `abs(X)` against cyclic frequency f. (Note: f is related to the Fourier frequency index k by $f = k/N$.)

(ii) Obtain an estimate of the phase `q` using the estimate `f1` and the following script:

```
p = [];
for r = 2*pi*(0:0.002:1)
    s = cos(2*pi*f1*n+r);
    p = [p; s4'*s];
end
```

(The maximum of the inner product `s4'*s` occurs for a value of `r` close to the true value of the phase shift `q`.) Finally, obtain an estimate of the amplitude `A`.

***P* 3.34.** The 128-point vector `s5` consists of three sinusoids of cyclic frequencies f_1, f_2 and f_3 (where $0 \le f_1 < f_2 < f_3 \le 1/2$) plus a small amount of noise. Plot `s5` against time. Use zero-padding to $N = 2048$ points and follow the same technique as in part **(i)** of Problem 3.33 to obtain estimates of f_1, f_2 and f_3.

Chapter 4

Introduction to Linear Filtering

4.1 The Discrete-Time Fourier Transform

4.1.1 From Vectors to Sequences

We will now discuss how the DFT tools developed in the previous chapter can be adapted for the analysis of discrete-time signals of infinite length, also known as *sequences*. The index set (i.e., time axis) for such a sequence $\mathbf{x}$ will consist of all integers:

$$\mathbf{x} = \{x[n],\ n \in \mathbf{Z}\}$$

As in MATLAB, we will use the notation

$$x[n_1 : n_2] = \left[\ x[n_1] \quad \ldots \quad x[n_2]\ \right]^T$$

to denote a *segment* of the sequence $\mathbf{x}$ corresponding to time indices n_1 through n_2. The length of the segment is finite provided both n_1 and n_2 are finite.

In order to express $\mathbf{x}$ as a linear combination of sinusoids of infinite length, it is important to understand how the DFT of a signal vector $\mathbf{s}$ behaves as the length N of $\mathbf{s}$ approaches infinity. To that end, we note the following:

- The analysis equation

  $$S[k] = \sum_{n=0}^{N-1} s[n] e^{-j(2\pi/N)kn}$$

  involves a sum over time indices 0 through $N-1$. This clearly becomes an infinite sum in the limit as $N \to \infty$. A similar expression for the infinite sequence $\mathbf{x}$ would have to include both positive and negative time indices, which would range from $n = -\infty$ to $n = +\infty$.

- The synthesis equation

  $$s[n] = \frac{1}{N} \sum_{k=0}^{N-1} S[k] e^{j(2\pi/N)kn}$$

  is a sum over frequency indices $k = 0$ through $k = N-1$. The corresponding radian frequencies are $\omega = 0$ through $\omega = 2\pi - (2\pi/N)$ in increments of $2\pi/N$. As $N \to \infty$, the spacing between consecutive frequencies shrinks to zero, and thus *every* frequency ω in $[0, 2\pi)$ becomes a Fourier frequency. The continuous-index version of a sum is an integral; thus as $N \to \infty$, one expects the synthesis equation to involve an integral over all frequencies in $[0, 2\pi)$, i.e.,

$$\sum_{k=0}^{N-1} \quad \to \quad \int_{\omega=0}^{\omega=2\pi} d\omega$$

In brief, both the analysis and synthesis equations are expected to take on different forms as $N \to \infty$.

4.1.2 Development of the Discrete-Time Fourier Transform

To develop the exact form of the analysis and synthesis equations for the sequence $\mathbf{x}$, consider the segment

$$\mathbf{x}^{(L)} = x[-L:L] = \begin{bmatrix} x[-L] & \dots & x[L] \end{bmatrix}^T$$

which has length $N = 2L+1$. The infinite sequence $\mathbf{x}$ is obtained by letting $L \to \infty$, as shown in Figure 4.1.

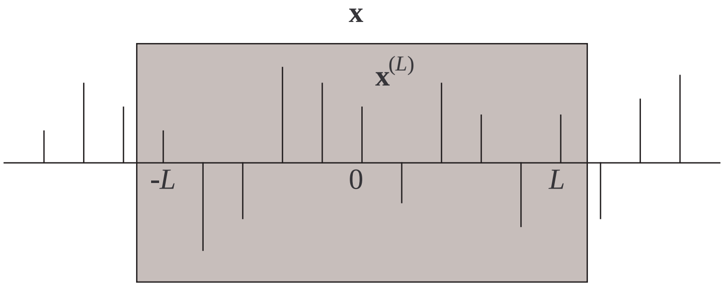


Figure 4.1: The DTFT of a sequence $\mathbf{x}$ is obtained from the DFT of the vector $\mathbf{x}^{(L)}$ by letting $L \to \infty$.

The standard DFT of $\mathbf{x}^{(L)}$ provides us with the coefficients needed for representing $\mathbf{x}^{(L)}$ as a linear combination of N sinusoidal vectors $\mathbf{v}^{(k)}$ of length N, each having a different Fourier frequency $k(2\pi/N)$. The topmost element in each $\mathbf{v}^{(k)}$ (which would correspond to time $n = -L$ in this case) equals $e^{j0} = 1$. For the task at hand, a more suitable choice for the k^{th} Fourier sinusoid is

$$\tilde{\mathbf{v}}^{(k)} = \left[e^{j(2\pi/N)kn} \right]_{n=-L}^{n=L}$$

whose middle entry (corresponding to time $n = 0$) equals 1. This circular shift has no effect on the orthogonality of the Fourier sinusoids, nor does it alter their norm. Thus, as in Subsection 3.1.3,

$$\langle \tilde{\mathbf{v}}^{(k)}, \tilde{\mathbf{v}}^{(\ell)} \rangle = \begin{cases} N, & k = \ell; \\ 0, & k \neq \ell \end{cases}$$

The resulting modified analysis and synthesis equations are

$$X^{(L)}[k] = \sum_{n=-L}^{L} x[n] e^{-j(2\pi/N)kn} \ , \qquad k = 0, \ldots, 2L \tag{4.1}$$

and

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X^{(L)}[k] e^{j(2\pi/N)kn} \ , \qquad n = -L, \ldots, L \tag{4.2}$$

where, again, $N = 2L + 1$. (Incidentally, we note that the frequency index k can also range from $-L$ to L, in which case the Fourier frequencies are taken in $(-\pi, \pi)$.)

As we noted earlier, the set of Fourier frequencies $k(2\pi/N)$ becomes the entire interval $[0, 2\pi)$ as $N \to \infty$. This prompts us to rewrite the sum in (4.1) as a function of ω instead of k, and to also take the limit as $L \to \infty$ (i.e., $N \to \infty$). The resulting expression

$$\sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$$

is a power series in $e^{j\omega}$. Since $e^{j\omega}$ is periodic with period 2π, so is the resulting sum.

Definition 4.1.1. The discrete-time Fourier transform (DTFT) of the sequence $\mathbf{x} = \{x[n],\ n \in \mathbf{Z}\}$ is defined by

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$$

provided the infinite sum converges. □

This definition serves as the *analysis* equation in the case of an infinite-length sequence, and provides us with a continuous set of coefficients $X(e^{j\omega})$ for frequencies in $[0, 2\pi)$. It remains to derive a *synthesis* equation, i.e., to show how the sequence $\mathbf{x}$ can be reconstructed by linearly combining sinusoids $e^{j\omega}$ with the above coefficients. As suggested earlier, this linear combination takes the form of an integral, which is constructed as follows.

For large values of L, we can write an approximation for the sum in (4.2) by replacing each coefficient $X^{(L)}[k]$ by the value of $X(e^{j\omega})$ at the same frequency:

$$x[n] \approx \frac{1}{N} \sum_{k=0}^{N-1} X\left(e^{j(2\pi/N)k}\right) \cdot e^{j(2\pi/N)kn}$$

Multiplying and dividing the right-hand side by 2π results in

$$x[n] \approx \frac{1}{2\pi} \sum_{k=0}^{N-1} X\left(e^{j(2\pi/N)k}\right) \cdot e^{j(2\pi/N)kn} \cdot \left(\frac{2\pi}{N}\right)$$

The sum consists of N equally spaced samples of the function $X(e^{j\omega})e^{j\omega n}$ over the interval $[0, 2\pi)$, each multiplied by the spacing $2\pi/N$ (in frequency). As $N \to \infty$, the sum converges to the integral of the function over the same interval. The approximation ($\approx$) also becomes exact in the limit, and therefore

$$x[n] = \frac{1}{2\pi} \int_0^{2\pi} X(e^{j\omega})e^{j\omega n}\, d\omega\ , \quad n \in \mathbf{Z}$$

In summary, we have obtained the *DTFT pair*

$$x[n] \quad \overset{\text{DTFT}}{\longleftrightarrow} \quad X(e^{j\omega})$$

defined by the equivalent equations:

- *Analysis:*

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \tag{4.3}$$

- *Synthesis:*

$$x[n] = \frac{1}{2\pi} \int_0^{2\pi} X(e^{j\omega})e^{j\omega n}\, d\omega\ , \quad n \in \mathbf{Z} \tag{4.4}$$

The synthesis equation is an example of a linear combination of a *continuously indexed* set of signals (the continuous index being frequency ω in this case). Each signal has its own coefficient (as was the case with the DFT), and the signals are combined at each time instant n by *integrating* over the range of the continuous index.

4.2 Computation of the Discrete-Time Fourier Transform

4.2.1 Signals of Finite Duration

From a computational viewpoint, there are essential differences between the DFT of a finite-length vector and the DTFT of a sequence of infinite length. The DFT is a finite vector of complex values obtained using a finite number of floating-point operations. The DTFT, on the other hand, is an infinite sum computed over a continuum of frequencies, which

- is rarely expressible in a simple closed form; and
- involves, as a rule, an infinite number of floating-point operations at each frequency.

An obvious exception to the second statement is the class of signal sequences which contain only finitely many nonzero values.

Definition 4.2.1. The signal sequence $\mathbf{x}$ has *finite duration* if there exist finite time indices n_1 and n_2 such that

$$n < n_1 \quad \text{or} \quad n > n_2 \qquad \Rightarrow \qquad x[n] = 0$$

It has *infinite duration* otherwise. □

The DTFT of a finite-duration sequence is computed by summing together finitely many terms. We begin by considering three simple examples of such sequences and their DTFT's.

Example 4.2.1. The *unit impulse* sequence (also known as *unit sample* sequence) is defined by

$$x[n] = \delta[n] = \begin{cases} 1, & n = 0; \\ 0, & n \neq 0 \end{cases}$$

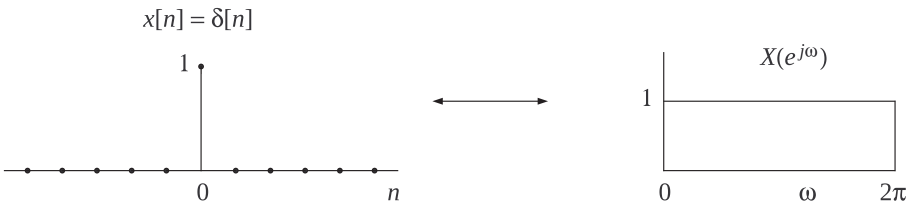
Its DTFT is given by

$$X(e^{j\omega}) = 1 \cdot e^{-j\omega \cdot 0} = 1$$

Using the synthesis equation 4.4, we see that the unit impulse combines *all* sinusoids in the frequency range $[0, 2\pi)$ with *equal* coefficients:

$$\delta[n] = \frac{1}{2\pi} \int_0^{2\pi} e^{j\omega n} \, d\omega$$

□

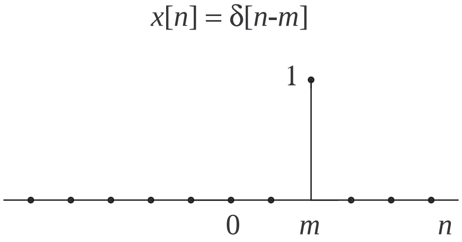


Example 4.2.1

Example 4.2.2. The delayed impulse

$$x[n] = \delta[n-m] = \begin{cases} 1, & n = m; \\ 0, & \text{otherwise} \end{cases}$$

is shown in the figure.



Example 4.2.2

Its DTFT is given by

$$X(e^{j\omega}) = e^{-j\omega m} = \cos \omega m - j \sin \omega m$$

and is complex valued except for $m = 0$. □

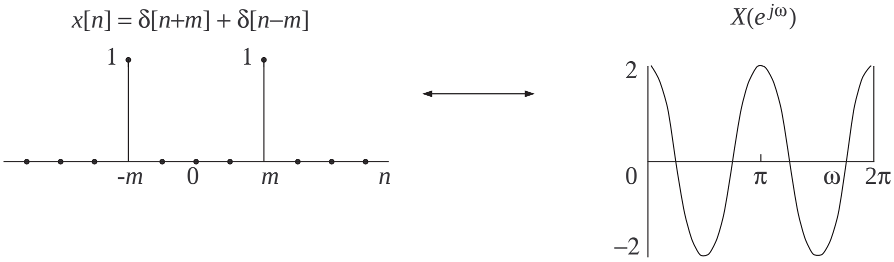
Example 4.2.3. The symmetric impulse pair

$$x[n] = \delta[n+m] + \delta[n-m]$$

has DTFT is given by

$$X(e^{j\omega}) = e^{-j\omega(-m)} + e^{-j\omega m} \ = 2\cos m\omega$$

This is a real sinusoid in ω having period $2\pi/m$. It is plotted in the case $m = 2$. □



Example 4.2.3

In Example 4.2.3, the time-domain signal was real-valued and symmetric about $n = 0$:

$$x[n] = x[-n]$$

The DTFT $X(e^{j\omega})$ was also real-valued and symmetric about $\omega = \pi$. A similar relationship between signals in the time and frequency domains was encountered earlier in our discussion of the DFT. There is, in fact, a direct correspondence between the structural properties of the two transforms, and one set of properties can be derived from the other using standard substitution rules. In the case of the DTFT, symmetry in the time domain is about $n = 0$ (which is the middle index of the modified DFT $X^{(L)}[k]$ introduced in Subsection 4.1.2). Also, time-index reversal (involved in the definition of symmetry) is understood in linear terms, i.e., $n \longrightarrow -n$; circular time reversal is impossible on an infinite time axis. The same is true for time delays of sequences, i.e., they are linear instead of circular.

We know that delaying a vector in time results in multiplying its spectrum by a complex sinusoid. As it turns out, the same is true for sequences:

Fact. *(Time Delay Property of the DTFT) If* $y[n] = x[n-m]$*, then* $Y(e^{j\omega}) = e^{-j\omega m}X(e^{j\omega})$*.*

Proof. We use the analysis equation (4.4) with a change of summation index ($n' = n - m$):

$$\begin{aligned} Y(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} y[n]e^{-j\omega n} \\ &= \sum_{n=-\infty}^{\infty} x[n-m]e^{-j\omega n} \end{aligned}$$

$$\begin{aligned} &= \sum_{n'=-\infty}^{\infty} x[n']e^{-j\omega(n'+m)} \\ &= e^{-j\omega m} \cdot \sum_{n'=-\infty}^{\infty} x[n']e^{-j\omega n'} \\ &= e^{-j\omega m} X(e^{j\omega}) \end{aligned}$$

□

4.2.2 The DFT as a Sampled DTFT

We now turn to an important connection between the DTFT of a finite-duration signal **x** and the DFT of a finite segment which contains all the nonzero values in **x**. Let **x** be given by

$$x[n] = \begin{cases} s[n], & 0 \le n \le L-1; \\ 0, & \text{otherwise} \end{cases}$$

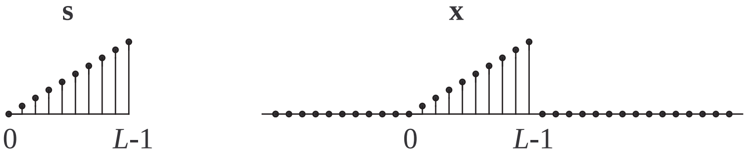


Figure 4.2: Vector **s** and its two-sided infinite zero-padded extension **x**.

We see that **x** is obtained from the L-point vector **s** by padding **s** with infinitely many zeros on both sides (as shown in Figure 4.2). The DFT **S** of **s** is given by

$$S[k] = \sum_{n=0}^{L-1} s[n]e^{-j(2\pi/L)kn} = \sum_{n=0}^{L-1} x[n]e^{-j(2\pi/L)kn} = X(e^{j(2\pi/L)k})$$

i.e., it is obtained by sampling the DTFT $X(e^{j\omega})$ of **x** at the Fourier frequencies for an L-point sample. Similarly, the DFT of the N-point zero-padded extension of **s** is obtained by sampling $X(e^{j\omega})$ at the Fourier frequencies for an N-point sample. As N increases, the set of Fourier frequencies becomes denser (since the spacing equals $2\pi/N$), and the DFT of the zero-padded vector provides uniformly spaced samples of $X(e^{j\omega})$ at a higher resolution.

If the vector $\mathbf{s}$ appears in the sequence $\mathbf{x}$ with a delay of m time units, i.e.,

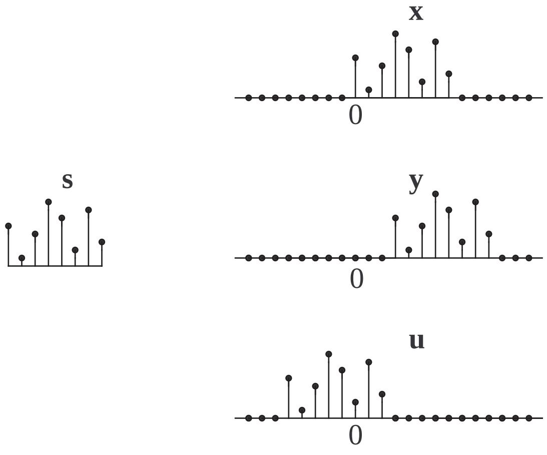
$$x[n] = \begin{cases} s[n-m], & m \le n \le m+L-1; \\ 0, & \text{otherwise,} \end{cases}$$

then by the time delay property established earlier, the DFT $\mathbf{S}$ is obtained by sampling $e^{j\omega m}X(e^{j\omega})$ at frequencies $\omega = k(2\pi/L)$. Also, a dense plot of the DTFT $X(e^{j\omega})$ can be obtained from the DFT of a zero-padded extension of $\mathbf{s}$ by multiplying each entry (of the DFT vector) by the corresponding sample of $e^{-j\omega m}$.

Example 4.2.4. Each of the finite-duration sequences $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{u}$ is a two-sided zero-padded extension of an eight-point vector $\mathbf{s}$ delayed by a different amount. Specifically, let

$$x[0:7] = y[3:10] = u[-5:2] = \mathbf{s}$$

as illustrated in the figure.



Example 4.2.4

Suppose we are interested in computing the DTFT's $X(e^{j\omega})$, $Y(e^{j\omega})$ and $U(e^{j\omega})$ with frequency resolution of 0.01 cycle/sample, i.e., for ω equal to multiples of $2\pi/100$. For that purpose, it suffices to compute 100-point DFT's. Using the standard MATLAB syntax `fft(s,N)` for the DFT of the N-point zero-padded extension of $\mathbf{s}$, we have:

```
k = (0:99).' ;          % s also assumed column vector
```

```
v = exp(j*2*pi*k/100) ;
X = fft(s,100) ;
Y = X.*(v.^(-3)) ;
U = X.*(v.^5) ;
```

(Alternatively, each of `Y` and `U` is the DFT of a circular shift on the zero-padded extension of **s**.) □

4.2.3 Signals of Infinite Duration

An infinite-duration signal sequence **x** has the property that $x[n] \neq 0$ for infinitely many values of n. As a result, the analysis equation

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

involves an infinite sum of complex-valued terms. The expression is meaningful only when the sum *converges absolutely*; in other words, when the magnitudes of the summands add up to a finite value:

$$\sum_{n=-\infty}^{\infty} |x[n]| \cdot |e^{-j\omega n}| = \sum_{n=-\infty}^{\infty} |x[n]| < \infty \tag{4.5}$$

Two important types of infinite-duration signals which *violate* the above condition are:

- sinusoids (real or complex) of any frequency; and
- periodic signals.

Fortunately, the analysis equation is not needed in order to derive the spectrum of either type of signal.

A real or complex sinusoid of requires no further frequency analysis, since every ω in $[0, 2\pi)$ is a valid Fourier frequency for the purpose of representing a sequence. Thus, for example,

$$x[n] = A\cos(\omega_0 n + \phi) = \frac{A}{2}e^{j\phi}e^{j\omega_0 n} + \frac{A}{2}e^{-j\phi}e^{-j\omega_0 n}$$

is a sum of two complex sinusoids at frequencies ω_0 and $2\pi - \omega_0$. We say that $x[n]$ has a *discrete spectrum* with components, or *lines*, at these two frequencies. To plot a discrete spectrum against frequency, we use a vertical line for each sinusoidal component, e.g., as shown in Figure 4.3.

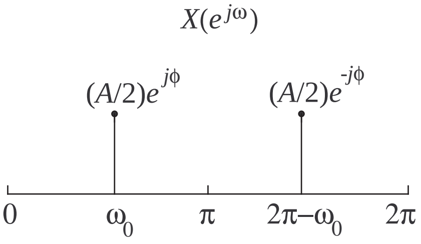


Figure 4.3: The spectrum of the real-valued sinusoid $x[n] = A\cos(\omega_0 n + \phi)$ is shown for $\omega \in [0, 2\pi)$. Note that $(A/2)e^{j\phi}$ and $(A/2)e^{-j\phi}$ are real-valued only when $\phi = 0$ or $\phi = \pi$.

Remark. Even though the sum in the *analysis* equation fails to converge to a valid function $X(e^{j\omega})$, it is still possible to express a discrete spectrum mathematically using a special type of function (of ω), so that the integral in the *synthesis* equation provides the correct time-domain signal. This representation is beyond the scope of the present discussion.

Periodic sequences can be also represented in terms of sinusoids without recourse to the analysis equation. This is because a sequence $\mathbf{x}$ of period L is the infinite two-sided periodic extension of the L-point vector

$$\mathbf{s} = x[0 : L-1] .$$

This is illustrated in Figure 4.4.

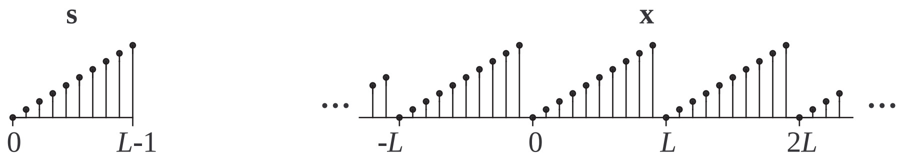


Figure 4.4: An infinite periodic sequence as the two-sided periodic extension of its first period.

From our discussion in Subsection 3.7.2, we know that if $\mathbf{S}$ is the (L-point) DFT of $\mathbf{s}$, then the synthesis equation

$$s[n] = \frac{1}{L}\sum_{k=0}^{L-1} S[k]e^{j(2\pi/L)kn} , \qquad n = 0, \ldots, L-1$$

also returns $x[n]$ for values of n outside the range $0 : L-1$. Thus

$$x[n] = \frac{1}{L}\sum_{k=0}^{L-1} S[k]e^{j(2\pi/L)kn} , \qquad n \in \mathbf{Z}$$

and we have established the following.

Fact. *A periodic sequence* $\mathbf{x}$ *of period* L *is the sum of* L *sinusoidal components at frequencies which are multiples of* $2\pi/L$. *The coefficients of these sinusoids are given by the DFT of* $x[0 : L-1]$ *scaled by* $1/L$. *The spectrum* $X(e^{j\omega})$ *also consists of* L *lines at the above-mentioned frequencies.* □

The spectrum of a periodic sequence is illustrated in Figure 4.5.

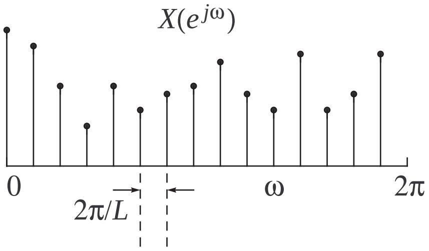


Figure 4.5: The spectrum of a periodic signal of period L, shown for $\omega \in [0, 2\pi)$.

We conclude our discussion with an example of an infinite-duration signal which satisfies the convergence condition (4.5).

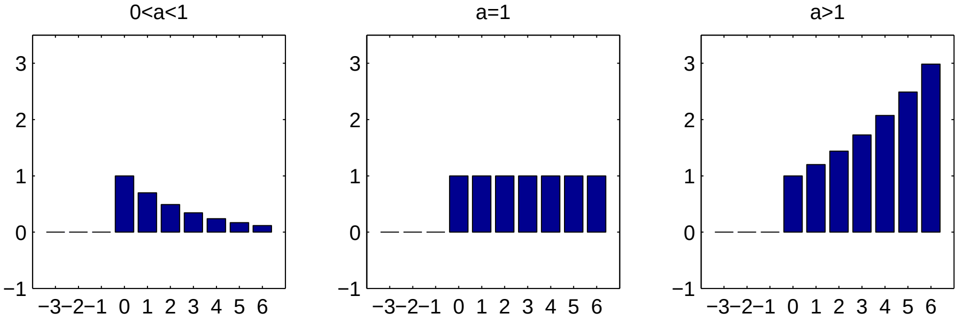
Example 4.2.5. Let $\mathbf{x}$ be a decaying exponential in positive time:

$$x[n] = \begin{cases} a^n, & n \geq 0; \\ 0, & n < 0 \end{cases}$$

where $|a| < 1$. This signal is shown (for $0 < a < 1$) together with the cases $a = 1$ and $a > 1$, both of which violate the convergence condition (4.5).

For $|a| < 1$, we have

$$\begin{aligned} X(e^{j\omega}) &= \sum_{n=0}^{\infty} a^n e^{-j\omega n} \\ &= \sum_{n=0}^{\infty} (ae^{-j\omega})^n \\ &= \frac{1}{1-ae^{-j\omega}} \end{aligned}$$



Example 4.2.5

where the last expression is the formula for the infinite geometric sum from Subsection 3.1.3. The condition $|ae^{-j\omega}| = |a| < 1$ guarantees convergence of the infinite sum. □

4.3 Introduction to Linear Filters

4.3.1 Linearity, Time-Invariance and Causality

In our earlier discussion of the DFT and its applications, we saw how the main sinusoidal components of a signal manifest themselves in the DFT of a suitably long segment of that signal. Identifying these components in the DFT allows us to reconstitute the signal (via an inverse DFT) in a more selective manner, e.g., by retaining desirable frequencies while eliminating undesirable ones. This type of signal transformation is known as *frequency-selective filtering* and can be applied to any segment of the signal.

Frequency-selective filtering can be performed more directly and efficiently without use of DFT's, once the frequencies of interest have been determined. In many applications, these frequencies are known ahead of time, e.g., radar return signals in a particular frequency band, or interference signals from a known source (e.g., a power supply at 50 Hz). In such cases, it is possible to use a so-called *linear filter* to generate the filtered signal in *real time*—i.e., at the rate at which the original signal is being sampled or recorded, and with minimal delay. The remainder of this chapter is devoted to basic concepts needed for the analysis of linear filters and their operation.

A linear filter is a special type of a linear transformation of an *input* signal sequence $\mathbf{x}$ to an *output* sequence $\mathbf{y}$. We call $\mathbf{y}$ the *response* of the filter to the input $\mathbf{x}$. Since the time index for a signal sequence varies from $n = -\infty$ to $n = +\infty$, we implicitly assume that the linear filter is active at all times—this is despite the fact that all signals encountered in practice have finite duration. The term *linear system*, used earlier for a linear transformation, is also applicable to a linear filter. Thus a linear filter $\mathcal{H}$ is a linear system whose input and output are related by

$$\mathbf{y} = \mathcal{H}(\mathbf{x})$$

x
Linear Filter
H
y

Figure 4.6: A linear filter.

Since the objective of filtering is to retain certain sinusoidal components while eliminating others, an essential property of the transformation $\mathcal{H}$ is the relationship between the spectra (i.e., DTFT's) of the sequences $\mathbf{x}$ and $\mathbf{y}$. We begin our discussion of filters by exploring that relationship in a very simple example.

Consider the transformation $\mathcal{H}$ described by the *input-output relationship*

$$y[n] = x[n] - x[n-1] + x[n-2] \ , \qquad n \in \mathbf{Z} \tag{4.6}$$

Recall our earlier definition of a linear transformation:

$$\left.\begin{array}{l} \mathbf{y}^{(1)} = \mathcal{H}(\mathbf{x}^{(1)}) \\ \mathbf{y}^{(2)} = \mathcal{H}(\mathbf{x}^{(2)}) \end{array}\right\} \Rightarrow \ c_1\mathbf{y}^{(1)} + c_2\mathbf{y}^{(2)} = \mathcal{H}(c_1\mathbf{x}^{(1)} + c_2\mathbf{x}^{(2)})$$

It is straightforward to show that the transformation $\mathcal{H}$ defined by (4.6) satisfies the above definition of linearity. In addition, $\mathcal{H}$ has the following properties:

- *Time-Invariance.* In the context of the above example, this means that the relationship between the output sample $y[n]$ and the input samples $x[n]$, $x[n-1]$ and $x[n-2]$ does not change with the time index n, i.e., the same coefficients (1, -1 and 1) are used to combine the three most recent input samples at any time n. Writing (4.6) in the form

  $$y[\cdot] = x[\cdot] - x[\cdot - 1] + x[\cdot - 2]$$

  better illustrates this feature.

- *Causality.* This means that the current output $y[n]$ depends *only* on *present* and *past* values of the input sequence—in this particular case, $x[n]$, $x[n-1]$ and $x[n-2]$. This is an essential constraint on all systems operating in real time, where future values of the input are unavailable. If, on the other hand, n represents a parameter other than time (e.g., a spatial index in an image or a pointer in an array of recorded data), causality is not a crucial constraint.

The defining properties of a linear filter are linearity and time invariance. Causality is an option which becomes necessary for filters operating in real time.

Before proceeding with the analysis of the filter described by (4.6), we illustrate the operation of the filter in Figure 4.7. The output $y[n]$ is a linear combination of $x[n]$, $x[n-1]$ and $x[n-2]$ with coefficients 1, -1 and 1, respectively. As n increases, the *time window* containing the three most recent values of the input slides to the right, while the values of the coefficients remain the same.

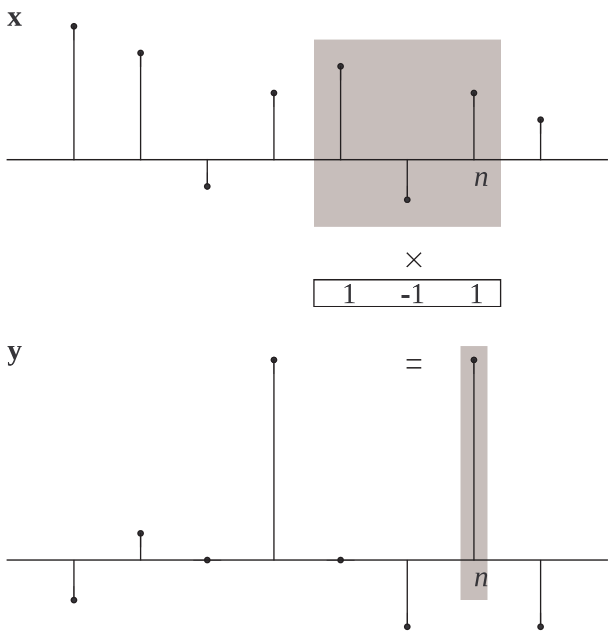


Figure 4.7: Illustration of the operation of the filter $y[n] = x[n] - x[n-1] + x[n-2]$.

4.3.2 Sinusoidal Inputs and Frequency Response

In order to understand how the time-domain equation (4.6) determines the relationship between the spectra of the input and output sequences **x** and **y**, consider a purely sinusoidal input at frequency ω:

$$x[n] = e^{j\omega n}\ , \qquad n \in \mathbf{Z}$$

The output sequence is then given by

$$\begin{aligned} y[n] &= e^{j\omega n} - e^{j\omega(n-1)} + e^{j\omega(n-2)} \\ &= (1 - e^{-j\omega} + e^{-j2\omega}) \cdot e^{j\omega n} \\ &= (1 - e^{-j\omega} + e^{-j2\omega}) \cdot x[n] \end{aligned}$$

for every value of n. Note that **y** is obtained by scaling each sample in **x** by the same (i.e., independent of time) amount. Since a complex scaling factor is equivalent to a change of amplitude and a shift in phase, we have illustrated the following important property of linear filters.

Fact. *When processed by a linear filter, pure sinusoids undergo no distortion other than a change in amplitude and a shift in phase.* □

Remark. Nonlinear filters distort pure sinusoids by introducing components at other frequencies, notably multiples of the input frequency; this effect is known as *harmonic distortion.*

The fact stated above can be proved for (almost) any linear filter using a similar factorization of the output $y[n]$ in terms of the input $x[n] = e^{j\omega n}$ and a a frequency-dependent complex scaling factor $H(e^{j\omega})$. The function $H(e^{j\omega})$ is known as the *frequency response* of the filter. In this case,

$$H(e^{j\omega}) = 1 - e^{-j\omega} + e^{-j2\omega}$$

We can thus write

$$x[n] = e^{j\omega n},\ n \in \mathbf{Z} \qquad \Rightarrow \qquad y[n] = H(e^{j\omega})e^{j\omega n},\ n \in \mathbf{Z} \tag{4.7}$$

We are now in a position to explore the relationship between an *arbitrary* input sequence $\mathbf{x}$ and the corresponding output sequence $\mathbf{y} = \mathcal{H}(\mathbf{x})$ by shifting our focus to the frequency domain. We may assume that the sequence $\mathbf{x}$ can be expressed as a sum of sinusoidal components, where the summation is either (i) discrete or (ii) continuous.

In the first case, which includes all periodic sequences, we have

$$x[n] = \sum_k X_k e^{j\omega_k n}$$

where the sum is over a discrete set of frequencies ω_k. The coefficients X_k are, in general, complex-valued. Since the filter is linear, the response $\mathbf{y}$ will be the sum of the responses to each of the sinusoids on the right-hand side. From (4.7), we know that the response to $e^{j\omega_k n}$ is given by $H(e^{j\omega_k})e^{j\omega_k n}$. We thus obtain

$$y[n] = \sum_k H(e^{j\omega_k})X_k e^{j\omega_k n}$$

We conclude that the output sequence $\mathbf{y}$ also has a discrete spectrum consisting of lines at the same frequencies (i.e., the ω_k's) as for the input sequence $\mathbf{x}$, but with coefficients scaled by the corresponding value of the frequency response $H(e^{j\omega_k})$. In other words,

$$y[n] = \sum_k Y_k e^{j\omega_k n}$$

where

$$Y_k = H(e^{j\omega_k})X_k$$

In the second case, we have (by the synthesis equation for the inverse DTFT) a representation of the filter input as

$$x[n] = \frac{1}{2\pi}\int_0^{2\pi} X(e^{j\omega})e^{j\omega n}\, d\omega$$

i.e., $\mathbf{x}$ is a linear combination of complex sinusoids whose frequencies range over the continuous interval $[0, 2\pi)$. Each of these sinusoids is scaled by the corresponding value of the frequency response $H(e^{j\omega})$ when processed by the filter, and linearity of the filter implies that

$$y[n] = \frac{1}{2\pi}\int_0^{2\pi} H(e^{j\omega})X(e^{j\omega})e^{j\omega n}\, d\omega$$

Comparing this expression for $y[n]$ with one provided by the synthesis equation

$$y[n] = \frac{1}{2\pi}\int_0^{2\pi} Y(e^{j\omega})e^{j\omega n}\, d\omega$$

we conclude that $Y(e^{j\omega})$ is, in fact, given by $H(e^{j\omega})X(e^{j\omega})$. This important result is stated below.

Fact. *For a linear filter, the complex spectrum of the output signal is given by the product of the complex spectrum of the input signal and the filter frequency response:*

$$Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$$

If the input spectrum is discrete, then so is the output spectrum, and the same relationship holds with $X(e^{j\omega})$ and $Y(e^{j\omega})$ replaced by the coefficients in the two spectra. □

Figure 4.8 illustrates the relationship 4.3.2 between input and output spectra.

4.3.3 Amplitude and Phase Response

We return to our basic example

$$y[n] = x[n] - x[n-1] + x[n-2]\ , \qquad n \in \mathbf{Z}$$

and compute the magnitude and angle of the frequency response

$$H(e^{j\omega}) = 1 - e^{-j\omega} + e^{-j2\omega}$$

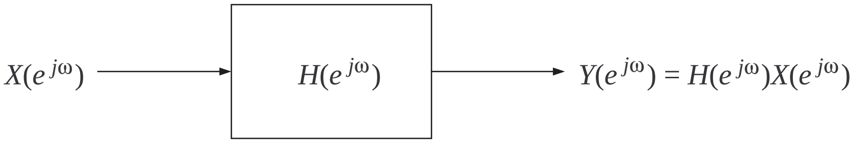


Figure 4.8: Input-output relationship in the frequency domain.

For the magnitude $|H(e^{j\omega})|$ (also known as the *amplitude response* of the filter), we have

$$\begin{aligned} \left|H(e^{j\omega})\right|^2 &= H^*(e^{j\omega})H(e^{j\omega}) \\ &= (1 - e^{-j\omega} + e^{-j2\omega})(1 - e^{j\omega} + e^{j2\omega}) \\ &= 3 - 2(e^{j\omega} + e^{-j\omega}) + (e^{j2\omega} + e^{-j2\omega}) \\ &= 3 - 4\cos\omega + 2\cos 2\omega \end{aligned}$$

and thus

$$|H(e^{j\omega})| = (3 - 4\cos\omega + 2\cos 2\omega)^{1/2}$$

Since $\cos(n\pi+\theta) = \cos(n\pi-\theta)$, we see that $|H(e^{j\omega})|$ is symmetric about $\omega = \pi$. This is due to the fact that $H(e^{j\omega})$ is the DTFT of a real-valued sequence, namely

$$\delta[n] - \delta[n-1] + \delta[n-2]$$

and as such, it exhibits the same kind of conjugate symmetry as does the DFT of a real-valued vector.

Note that in this case, the amplitude response achieves its maximum at $\omega = \pi$, and has two zeros at $\omega = \pi/3$ and $\omega = 5\pi/3$. This implies that:

- among all possible complex sinusoidal inputs, $(-1)^n$ will undergo the most amplification (or least attenuation);

- sinusoidal inputs such as $e^{j(\pi/3)n}$, $e^{j(5\pi/3)n}$ and $\cos(\pi n/3 + \phi)$ will all be *nulled out*, i.e., will result in $y[n] = 0$ for all n.

The symmetry of the filter coefficients also allows us to obtain $|H(e^{j\omega})|$ in a more direct way:

$$\begin{aligned} H(e^{j\omega}) &= e^{-j\omega}(e^{j\omega} - 1 + e^{-j\omega}) \\ &= e^{-j\omega}(2\cos\omega - 1) \end{aligned}$$

Since $|e^{-j\omega}| = 1$, we obtain

$$|H(e^{j\omega})| = |2\cos\omega - 1|$$

which is equivalent to the earlier expression (as can be shown by using the identity $\cos 2\omega = 2\cos^2\omega - 1$).

The angle $\angle H(e^{j\omega})$ is known as the *phase response* of the filter. In this case, we have

$$\angle H(e^{j\omega}) = -\omega + \angle(2\cos\omega - 1)$$

Since $2\cos\omega - 1$ is a real number, the second term equals 0 (when the number is positive) or π (when the number is negative). We thus have

$$\angle H(e^{j\omega}) = \begin{cases} -\omega, & 0 \le \omega < \pi/3; \\ -\omega + \pi, & \pi/3 \le \omega \le 5\pi/3; \\ -\omega, & 5\pi/3 < \omega < 2\pi. \end{cases}$$

The amplitude and phase responses are plotted in Figure 4.9, against cyclic frequency $\omega/2\pi$. Note that the phase response is antisymmetric about $\omega = \pi$, i.e., $\angle H(e^{j\omega}) = -\angle H(e^{j(\pi-\omega)})$, which is also due to the fact that the filter coefficients are real-valued.

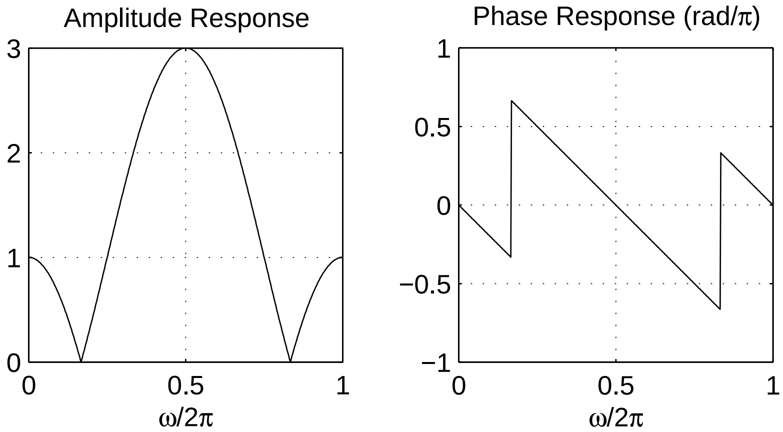


Figure 4.9: Amplitude response (left) and phase response (right) of the filter $y[n] = x[n] - x[n-1] + x[n-2]$. Frequencies and phases are normalized by 2π and π, respectively.

4.4 Linear Filtering in the Frequency Domain

4.4.1 System Function and its Relationship to Frequency Response

In the previous section, we saw that the response of the linear filter $\mathcal{H}$ to the complex sinusoidal input

$$x[n] = e^{j\omega n} , \qquad n \in \mathbf{Z}$$

equals

$$y[n] = H(e^{j\omega})e^{j\omega n} , \qquad n \in \mathbf{Z}$$

In other words, the input and output signal sequences differ by a complex scaling factor which depends on the frequency ω:

$$\mathbf{y} = H(e^{j\omega}) \cdot \mathbf{x}$$

This factor was called the *frequency response* of the filter.

This scaling (or proportionality) relationship between the input and output sequences can be generalized to a class of signals known as *complex exponentials*:

$$x[n] = z^n , \qquad n \in \mathbf{Z}$$

where z is any complex number other than $z = 0$ (which would give $z^n = \infty$ for $n < 0$). Writing z in the standard polar form

$$z = re^{j\omega}$$

we have

$$x[n] = r^n e^{j\omega n}$$

Thus a complex sinusoid is a special case of a complex exponential where $r = 1$, i.e., z lies on the unit circle. Also,

$$|x[n]| = r^n \cdot \left|e^{j\omega n}\right| = r^n$$

and thus the magnitude of $x[n]$

- increases geometrically (in n) if $|z| > 1$;
- decreases geometrically if $|z| < 1$;
- is constant if $|z| = 1$,

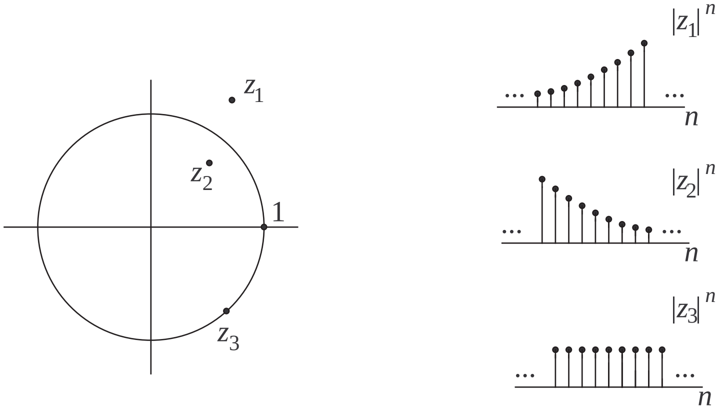


Figure 4.10: Three choices of z (left) and the corresponding complex exponentials in magnitude (right).

as illustrated in Figure 4.10.

Consider the filter introduced in Section 4.3, namely

$$y[n] = x[n] - x[n-1] + x[n-2] \ , \qquad n \in \mathbf{Z}$$

The response of the filter to $x[n] = z^n$ equals

$$\begin{aligned} y[n] &= z^n - z^{n-1} + z^{n-2} \\ &= (1 - z^{-1} + z^{-2}) z^n \\ &= (1 - z^{-1} + z^{-2}) \cdot x[n] \end{aligned}$$

Letting

$$H(z) = 1 - z^{-1} + z^{-2}$$

we can restate the above result as follows:

$$x[n] = z^n, \ n \in \mathbf{Z} \qquad \Rightarrow \qquad y[n] = H(z) z^n, \ n \in \mathbf{Z} \tag{4.8}$$

The fact established in Subsection 4.3.2, namely that

$$x[n] = e^{j\omega n}, \ n \in \mathbf{Z} \qquad \Rightarrow \qquad y[n] = H(e^{j\omega}) e^{j\omega n}, \ n \in \mathbf{Z}$$

is a special case of (4.8) where $z = e^{j\omega n}$.

The function $H(z)$, where z is complex, is known as the *system function*, or *transfer function*, of the filter $\mathcal{H}$. The frequency response of the filter is obtained from $H(z)$ by taking z on the unit circle, i.e., $z = e^{j\omega}$.

4.4.2 Cascaded Filters

The *cascade* (also referred to as *series* or *tandem*) connection of two filters $\mathcal{H}_1$ and $\mathcal{H}_2$ is illustrated in Figure 4.11. At each time instant, the output of $\mathcal{H}_1$ serves as the input to $\mathcal{H}_2$.

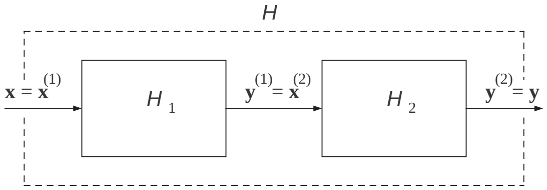


Figure 4.11: Cascade connection of two filters.

In Chapter 2, we showed that the cascade $(A \circ B)(\cdot)$ of two linear transformations $A(\cdot)$ and $B(\cdot)$ is represented by the matrix product $\mathbf{AB}$ (where $\mathbf{A}$ and $\mathbf{B}$ are the matrices corresponding to $A(\cdot)$ and $B(\cdot)$, respectively). A linear filter is a special type of linear transformation which is invariant in time. The cascade of two such filters is also a linear filter $\mathcal{H}$ whose system function $H(z)$ bears a particularly simple relationship to the system functions $H_1(z)$ and $H_2(z)$ of the constituent filters.

From (4.8), we know that $H(z)$ is the scaling factor between the input and output sequences $\mathbf{x}$ and $\mathbf{y}$ when the former (input) is given by $x[n] = z^n$. In this case, $\mathbf{x}$ is also the input to the first filter $\mathcal{H}_1$, which results in an output

$$y^{(1)}[n] = H_1(z)z^n$$

This scaled complex exponential is the input to the second filter $\mathcal{H}_2$, and thus

$$y[n] = y^{(2)}[n] = H_1(z)H_2(z)z^n$$

We have established the following.

Fact. *The cascade connection $\mathcal{H}$ of two filters $\mathcal{H}_1$ and $\mathcal{H}_2$ has system function given by*

$$H(z) = H_1(z)H_2(z)$$

Similarly, the frequency response of $\mathcal{H}$ is given by

$$H(e^{j\omega}) = H_1(e^{j\omega})H_2(e^{j\omega}) \qquad \Box$$

Note that the order in which the two filters are cascaded is immaterial; the same system function is obtained if $\mathcal{H}_2$ is followed by $\mathcal{H}_1$.

Example 4.4.1. Consider a cascade of two identical filters, whose input-output relationship is given by (4.6). Thus

$$\begin{aligned} y^{(1)}[n] &= x[n] - x[n-1] + x[n-2] \\ y[n] &= y^{(1)}[n] - y^{(1)}[n-1] + y^{(1)}[n-2] \end{aligned}$$

In this case, $H_1(z) = H_2(z) = 1 - z^{-1} + z^{-2}$, and thus the system function of the cascade is given by

$$H(z) = H_1(z)H_2(z) = 1 - 2z^{-1} + 3z^{-2} - 2z^{-3} + z^{-4}$$

Similarly,

$$\begin{aligned} H(e^{j\omega}) &= H_1(e^{j\omega})H_2(e^{j\omega}) \\ &= 1 - 2e^{-j\omega} + 3e^{-j2\omega} - 2e^{-j3\omega} + e^{-j4\omega} \\ &= e^{-j2\omega}(3 - 4\cos\omega + 2\cos 2\omega) \\ &= e^{-j2\omega}(1 - 2\cos\omega)^2 \end{aligned}$$

□

4.4.3 The General Finite Impulse Response Filter

We introduced linear filters by considering a particular example where the filter output at any particular instant is formed by linearly combining the three most recent values of the input, with coefficients which are constant in time. There is nothing special about choosing *three* input samples; we can draw the same conclusions about the filter given by the equation

$$y[n] = b_0x[n] + b_1x[n-1] + \cdots + b_Mx[n-M] \ , \qquad n \in \mathbf{Z} \qquad (4.9)$$

where the $M+1$ most recent values of the input are combined to form the output at any given time.

Definition 4.4.1. A filter defined by (4.9) is known as a *finite impulse response* (*FIR*) filter of *order* M. □

The qualifier *finite* in the above definition reflects the fact that a finite number of input samples are combined to form a particular output sample. The term *impulse response* will be defined shortly.

The system function of the FIR filter defined by (4.9) is given by

$$H(z) = b_0 + b_1z^{-1} + \cdots + b_Mz^{-M}$$

and the frequency response is obtained by letting $z = e^{j\omega}$:

$$H(e^{j\omega}) = b_0 + b_1 e^{-j\omega} + \cdots + b_M e^{-j\omega M}$$

Comparing the above equation for $H(e^{j\omega})$ with the definition of the DTFT, we see that $H(e^{j\omega})$ is, in fact, the DTFT of the finite-duration sequence **h** defined by

$$h[n] = \begin{cases} b_n, & 0 \le n \le M; \\ 0, & \text{otherwise.} \end{cases}$$

The sequence **h** is known as the *impulse response* of the filter. As we shall see soon, it is the response of the filter to a unit impulse, i.e., to the signal $x[n] = \delta[n]$.

The coefficient vector **b** and the impulse response **h** for the example of Section 4.3 is shown in Figure 4.12.

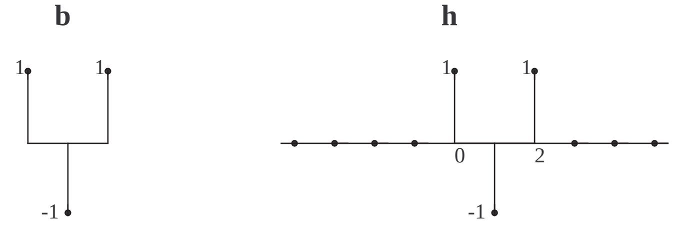


Figure 4.12: The coefficient vector (left) and the impulse response sequence (right) of the FIR filter $y[n] = x[n] - x[n-1] + x[n-2]$.

The relationship between **b** and the frequency response $H(e^{j\omega})$ suggests a convenient way of computing $H(e^{j\omega})$ based on the DFT of **b**. To evaluate $H(e^{j\omega})$ at $N \ge M+1$ evenly spaced frequencies in $[0, 2\pi)$, all we need to do is compute the DFT of the $(M+1)$-point vector **b** zero-padded to length N. Thus for the filter considered in Section 4.3, the frequency response at $N = 256$ evenly spaced frequencies can be computed in MATLAB using the command

```
H = fft([1;-1;1],256)
```

The amplitude and phase responses in Figure 4.9 in the previous section are the graphs of `abs(H)` and `angle(H)`, respectively.

4.4.4 Frequency Domain Approach to Computing Filter Response

The input-output relationship

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_M x[n-M]$$

provides us with a direct way of implementing an FIR filter. At every instant n, a new value of the input $x[n]$ is read into a buffer, an old value (namely $x[n-M-1]$) is shifted out of the same buffer, and the output $y[n]$ is computed using $M+1$ multiplications and M additions.

In applications where a single processor is used to perform many signal processing tasks in parallel, computational efficiency is a key consideration. It therefore pays to examine how a simple component such as an FIR filter (given by the above equation) can be implemented efficiently. In this subsection, we provide a partial answer to this question for two special types of inputs, both of which are readily expressed in terms of complex exponentials or complex sinusoids. A third class of such input signals will be considered in the following subsection.

The first type of input we consider is a real-valued sinusoid, i.e.,

$$x[n] = A\cos(\omega_0 n + \phi) \ , \qquad n \in \mathbf{Z}$$

Using the identity $e^{j\theta} + e^{-j\theta} = 2\cos\theta$, we can write the signal in terms of two complex sinusoids as

$$x[n] = \frac{A}{2} e^{j\phi} e^{j\omega_0 n} + \frac{A}{2} e^{-j\phi} e^{-j\omega_0 n}$$

(In other words, $\mathbf{x}$ has a discrete spectrum with lines at frequencies ω and $2\pi - \omega$.) By linearity of the filter, the output is given by

$$y[n] = \frac{A}{2} e^{j\phi} H(e^{j\omega_0}) e^{j\omega_0 n} + \frac{A}{2} e^{-j\phi} H(e^{-j\omega_0}) e^{-j\omega_0 n}$$

At this point, we note that

$$\begin{aligned} H(e^{-j\omega}) &= b_0 + b_1 e^{j\omega} + \cdots + b_M e^{j\omega M} \\ &= \big(b_0 + b_1 e^{-j\omega} + \cdots + b_M e^{-j\omega M}\big)^* = H^*(e^{j\omega}) \end{aligned}$$

where, for the last equality, we also used the fact the the filter coefficients b_i are real. We thus have

$$y[n] = \frac{A}{2} e^{j\phi} H(e^{j\omega_0}) e^{j\omega_0 n} + \frac{A}{2} e^{-j\phi} H^*(e^{j\omega_0}) e^{-j\omega_0 n}$$

The two terms on the right-hand side are complex conjugates of each other, and thus the identity $z + z^* = 2\Re e\{z\}$ gives

$$y[n] = \Re e\left\{\; AH(e^{j\omega_0})e^{j(\omega_0 n+\phi)}\right\}$$

An alternative expression can be written using the amplitude and phase responses $\left|H(e^{j\omega})\right|$ and $\angle H(e^{j\omega})$:

$$H(e^{j\omega}) = \left|H(e^{j\omega})\right| e^{j\angle H(e^{j\omega})}$$

and thus

$$\begin{aligned} y[n] &= \Re e\left\{\; A\left|H(e^{j\omega_0})\right| e^{j(\omega_0 n+\phi+j\angle H(e^{j\omega_0}))}\right\} \\ &= A\left|H(e^{j\omega_0})\right|\cos\left(\omega_0 n + \phi + \angle H(e^{j\omega_0})\right) \end{aligned}$$

Comparing the resulting expression with $x[n] = A\cos(\omega_0 n + \phi)$, we conclude the following.

Fact. *The response of a linear filter to a real-valued sinusoid of frequency ω_0 is a sinusoid of the same frequency. The **gain** (ratio of output to input amplitude) is given by the filter amplitude response at ω_0, while the **phase shift** (of the output relative to the input) is given by the phase response at the same frequency.* □

Example 4.4.2. Let

$$x[n] = \cos\left(\frac{2\pi n}{3} - \frac{\pi}{4}\right)\;, \qquad n \in \mathbf{Z}$$

be the input to the second-order FIR filter

$$y[n] = x[n] - x[n-1] + x[n-2]\;, \qquad n \in \mathbf{Z}$$

introduced in Section 4.3. We have

$$\begin{aligned} H(e^{j(2\pi/3)}) &= e^{-j(2\pi/3)}(2\cos(2\pi/3) - 1) \\ &= -2e^{-j(2\pi/3)} \\ &= 2e^{j(\pi/3)} \end{aligned}$$

and thus $|H(e^{j(2\pi/3)})| = 2$, $\angle H(e^{j(2\pi/3)}) = \pi/3$. Using the fact established above, we obtain

$$\begin{aligned} y[n] &= 2\cos\left(\frac{2\pi n}{3} - \frac{\pi}{4} + \frac{\pi}{3}\right) \\ &= 2\cos\left(\frac{2\pi n}{3} + \frac{\pi}{12}\right)\;, \qquad n \in \mathbf{Z} \end{aligned}$$ □

The foregoing analysis can be also extended to the real exponential

$$x[n] = Ar^n \ , \qquad n \in \mathbf{Z}$$

and the real *oscillating* exponential

$$x[n] = Ar^n \cos(\omega_0 n + \phi) \ , \qquad n \in \mathbf{Z}$$

where A and r are real-valued. In each case, the output can be computed in terms of the system function $H(z)$ as follows:

$$\begin{array}{rcl} x[n] = Ar^n & \Longrightarrow & y[n] = AH(r)r^n \\ x[n] = Ar^n \cos(\omega_0 n + \phi) & \Longrightarrow & y[n] = A\left|H(re^{j\omega_0})\right| r^n \cos\left(\omega_0 n + \phi + \angle H(re^{j\omega_0})\right) \end{array}$$

Example 4.4.3. Consider the same filter as in Section 4.3, with two different input sequences:

$$\begin{aligned} x^{(1)}[n] &= 3^n \\ x^{(2)}[n] &= 3^n \cos\left(\frac{2\pi n}{3}\right) \end{aligned}$$

The filter system function is given by

$$H(z) = 1 - z^{-1} + z^{-2}$$

and thus

$$\begin{aligned} H(3) &= 7/9 \\ H(3e^{j(2\pi/3)}) &= 2.614 + j10.472 = 10.793 \cdot e^{j1.326} \end{aligned}$$

It follows that

$$\begin{aligned} y^{(1)}[n] &= \frac{7}{9} \cdot 3^n \\ y^{(2)}[n] &= 10.793 \cdot 3^n \cdot \cos\left(\frac{2\pi n}{3} + 1.326\right) \end{aligned}$$

□

4.4.5 Response to a Periodic Input

The third type of input $\mathbf{x}$ for which the filter response $\mathbf{y}$ can be readily computed using a frequency domain-based approach (i.e., based on $H(e^{j\omega})$) is a periodic sequence of period (say) L:

$$x[n + L] = x[n] \ , \qquad n \in \mathbf{Z}$$

Before we explain how $H(e^{j\omega})$ can be used to compute $\mathbf{y}$, we note that

$$\begin{aligned} y[n+L] &= b_0 x[n+L] + b_1 x[n+L-1] + \cdots + b_M x[n+L-M] \\ &= b_0 x[n] + b_1 x[n-1] + \cdots + b_M x[n-M] \\ &= y[n] \end{aligned}$$

and thus *the response is also periodic with period L.* It therefore suffices to compute $y[0 : L-1]$, which will be referred to as the *first period* of $\mathbf{y}$.

The frequency domain approach to computing $y[0 : L-1]$ exploits the fact that the periodic input signal $\mathbf{x}$ is the sum of L sinusoids at frequencies which are multiples of $2\pi/L$. As was discussed in Subsection 4.2.3, if the first period $x[0 : L-1]$ of $\mathbf{x}$ has DFT $\mathcal{X}[\cdot]$, then the equation

$$x[n] = \frac{1}{L} \sum_{k=0}^{L-1} \mathcal{X}[k] e^{j(2\pi/L)kn}$$

holds for every n, i.e., it describes the entire sequence $\mathbf{x}$. By linearity of the filter, the output sequence $\mathbf{y}$ is given by

$$y[n] = \frac{1}{L} \sum_{k=0}^{L-1} H\left(e^{j(2\pi/L)k}\right) \mathcal{X}[k] e^{j(2\pi/L)kn}$$

Since $\mathbf{y}$ is also periodic with period L, we have (similarly to $\mathbf{x}$)

$$y[n] = \frac{1}{L} \sum_{k=0}^{L-1} \mathcal{Y}[k] e^{j(2\pi/L)kn}$$

where $\mathcal{Y}[\cdot]$ is the DFT of $y[0 : L-1]$. Comparing the two expressions for $y[n]$, we see that

$$\mathcal{Y}[k] = H\left(e^{j(2\pi/L)k}\right) \mathcal{X}[k]$$

In other words, the DFT of $y[0 : L-1]$ is obtained by multiplying each entry of the DFT of $x[0 : L-1]$ by the frequency response of the filter at the corresponding Fourier frequency.

This observation leads to the following algorithm for computing $y[0 : L-1]$ given $x[0 : L-1]$ and the coefficient vector $\mathbf{b}$:

- Compute the DFT of $x[0 : L-1]$.
- Compute $H(e^{j\omega})$ at the same Fourier frequencies, e.g., using the DFT of $\mathbf{b}$ after zero-padding to a multiple of L.

- Multiply the two vectors element-by-element.
- Invert the DFT of the resulting vector to obtain $y[0:L-1]$.

Example 4.4.4. Consider the second-order filter introduced in Section 4.3. Suppose that the input sequence $\mathbf{x}$ is periodic with period $L=7$, and such that

$$x[0:6] = \begin{bmatrix} 2 & 5 & 3 & 4 & -7 & -4 & -1 \end{bmatrix}^T$$

The first period of the output sequence is computed in MATLAB as follows:

```
x = [2 5 3 4 -7 -4 -1].' ;
b = [1 -1 1].';              % filter coefficient vector
X = fft(x);
H = fft(b,7);
Y = H.*X;
y = ifft(Y)
```

resulting in

$$y[0:6] = \begin{bmatrix} -1 & 2 & 0 & 6 & -8 & 7 & -4 \end{bmatrix}^T \qquad \square$$

In the example above, $M+1<L$, i.e., the coefficient vector was shorter than the first input period. By zero-padding the coefficient vector $\mathbf{b}$ to length L and taking its DFT, we obtained precisely those values of the frequency response $H(e^{j\omega})$ which were needed in order to compute the product in (4.4.5). Recalling that element-wise multiplication of two DFT's of the same length is equivalent to circular convolution in the time domain, we see that the first period of the output signal is, in effect, computed by circularly convolving the first period of the input signal with the zero-padded (to length L) coefficient vector. This fact can be established independently by examining the time-domain expression

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

in the special case where $\mathbf{x}$ is periodic with period L. For this reason, circular convolution is also known as *periodic convolution.*

If the length of the coefficient vector is greater than the input period L, then the algorithm outlined in Example 4.4.4 needs to be modified—otherwise the command `H = fft(b,L)` would lead to truncation of $\mathbf{b}$, resulting in an error. Two possible modifications, assuming that $M+1$ is satisfies

$$(J-1)L < M+1 \le JL$$

for some integer $J>1$, are:

- Zero-pad $\mathbf{b}$ to length JL, then extract every J^{th} entry of the resulting DFT. Or,

- Use the first L periods of $\mathbf{x}$ (in conjunction with zero-padding $\mathbf{b}$ to length JL), in which case the answer will consist of the first L periods of $\mathbf{y}$. This approach is also equivalent to a circular convolution in the time domain (with vectors of length JL).

4.5 Frequency-Selective Filters

4.5.1 Main Filter Types

We motivated our discussion of linear filters using the concept of *frequency selection.* A frequency-selective filter reconstructs at its output certain desirable sinusoidal components present in the input signal, while attenuating or eliminating others. The range of frequencies which are preserved is known as the filter *passband*, while those that are rejected are referred to as the *stopband*.

For filters with real-valued coefficients operating in discrete time, it is customary to specify frequency bands as subintervals of $[0, \pi]$. This is because the filter frequency response $H(e^{j\omega})$ has conjugate symmetry about $\omega = \pi$:

$$H(e^{j(2\pi-\omega)}) = H(e^{-j\omega}) = H^*(e^{j\omega}) \tag{4.10}$$

(This is equivalent to symmetry in the amplitude response and antisymmetry in the phase response.) Thus it suffices to specify $H(e^{j\omega})$ over $[0, \pi]$, which is also the effective range of frequencies for real-valued sinusoids.

Remark. Any frequency band (i.e., interval) in $[0, \pi]$ has a symmetric band in $[\pi, 2\pi]$. Equation (4.10) also implies that $H(e^{j\omega})$ has conjugate symmetry about $\omega = 0$; thus any frequency band in $[0, \pi]$ has a symmetric band in $[-\pi, 0]$, as well. This is illustrated in Figure 4.13.

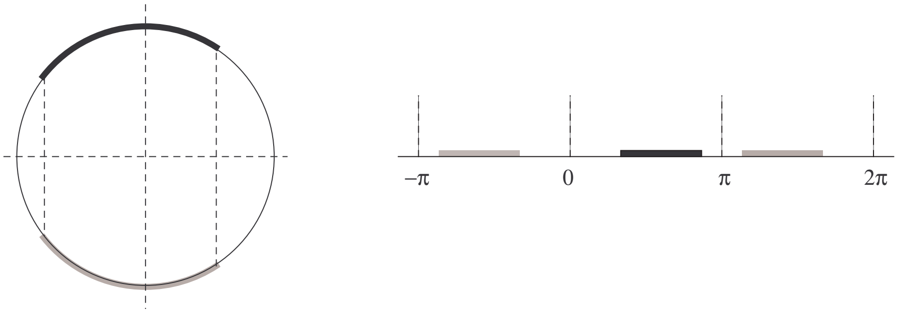


Figure 4.13: A frequency band in $[0, \pi]$ (dark) and its symmetric bands (light), shown on the unit circle (left) and the frequency axis (right).

The four main types of frequency selective filters are: *lowpass*, *highpass*, *bandpass* and *bandstop*. The frequency characteristics (i.e., passband and stopband) of each type are illustrated in Figure 4.14.

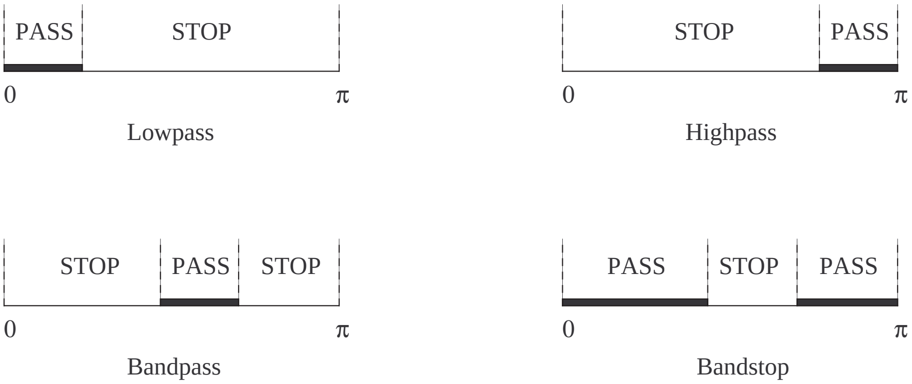


Figure 4.14: Passbands and stopbands of frequency-selective filters.

4.5.2 Ideal Filters

An (zero-delay) *ideal filter* has the following properties:

- the passband and stopband are complements of each other (with respect to $[0, \pi]$);
- the frequency response is constant over the passband; and
- the frequency response is zero over the stopband.

We illustrate some general properties of ideal filters by considering the ideal lowpass filter. The filter frequency response is shown in Figure 4.15, plotted over a full cycle $(-\pi, \pi]$. The edge of the passband is known as the *cutoff* frequency ω_c.

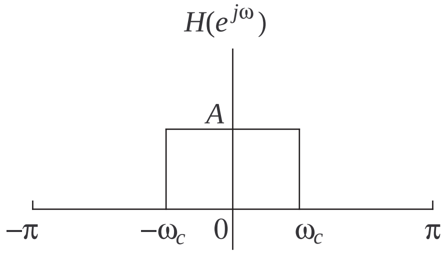


Figure 4.15: Frequency response of an ideal lowpass filter.

The impulse response **h** of the ideal lowpass filter is given by the inverse DTFT of its frequency response $H(e^{j\omega})$. We thus have

$$\begin{aligned} h[n] &= \frac{1}{2\pi}\int_0^{2\pi} H(e^{j\omega})e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\omega})e^{j\omega n}d\omega \\ &= \frac{A}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega n}d\omega \\ &= \frac{A}{2\pi}\left[\frac{e^{j\omega n}}{jn}\right]_{-\omega_c}^{\omega_c} \\ &= \frac{A}{2\pi}\cdot\frac{e^{j\omega_c n}-e^{-j\omega_c n}}{jn} \\ &= \frac{A\sin(n\omega_c)}{\pi n} \end{aligned}$$

The impulse response **h** is plotted in Figure 4.16 for the case $\omega_c = \pi/4$, (with A=1). For any cutoff frequency $\omega_c < \pi$, **h** is a sequence of infinite duration, extending from $n = -\infty$ to $n = +\infty$. By examining the input-output relationship

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] \ ,$$

we see that *infinitely* many *future* values of the input signal **x** are needed in order to compute the output at any time n. This is clearly infeasible in practice (i.e., the filter is “irrecoverably” noncausal). Practical lowpass filters have frequency responses that necessarily deviate from the ideal form shown in Figure 4.15.

Similar conclusions can be drawn for other ideal frequency-selective filters (highpass, bandpass and bandstop). In short, filters with perfectly flat response over the passband, zero response over the stopband, and infinitely steep edges at the cutoff frequencies are not practically realizable.

4.5.3 Amplitude Response of Practical Filters

Figure 4.17 shows the main features of the amplitude response $|H(e^{j\omega})|$ of a lowpass filter used in practice.

- The *passband* is the interval of frequencies over which the amplitude response takes high values. These values range from $A(1-\delta)$ to $A(1+\delta)$

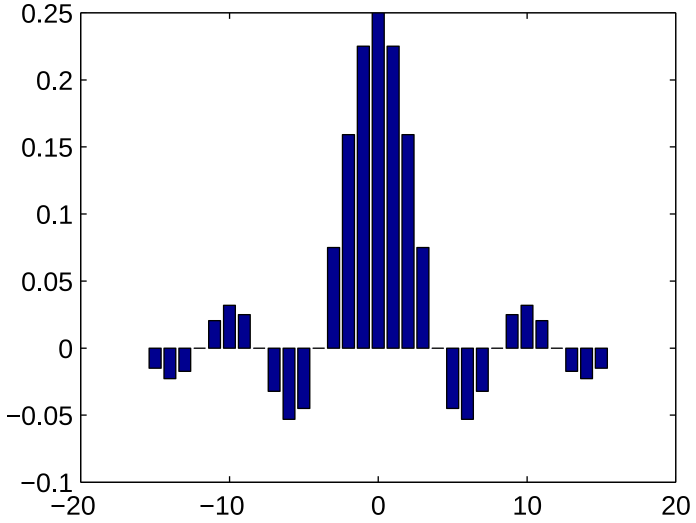


Figure 4.16: Impulse response $h[n]$ of an ideal lowpass filter with cutoff frequency $\omega_c = \pi/4$, plotted for $n = -15$ to $n = 15$.

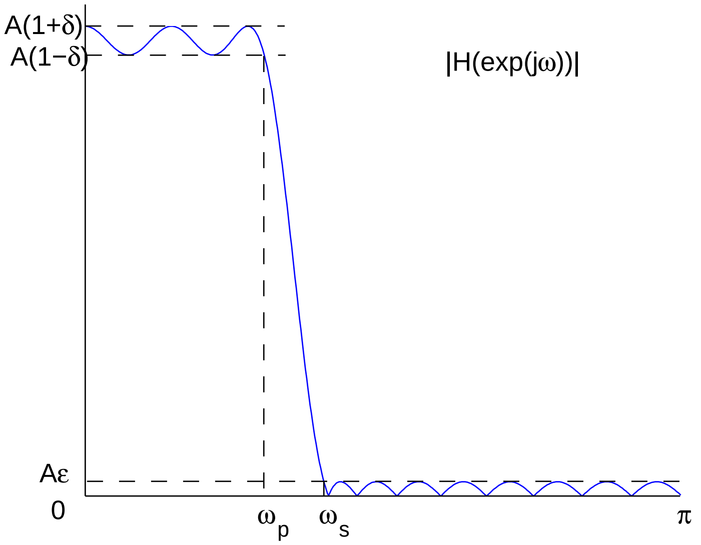


Figure 4.17: Amplitude response of a practical lowpass filter.

(as shown in Figure 4.17), and the fluctucation in value is known as (passband) *ripple*. The edge (endpoint) of the passband ω_p is the highest frequency for which $|H(e^{j\omega})| = A(1 - \delta)$.

- The *stopband* is the interval of frequencies over which the amplitude response takes low values; these range from 0 to ϵ in Figure 4.17. Again, the fluctuation in the value of $|H(e^{j\omega})|$ is referred to as (stopband) ripple. The ratio

$$\frac{\text{midpoint of amplitude range over passband}}{\text{maximum of amplitude over stopband}} = \frac{A}{A\epsilon} = \frac{1}{\epsilon}$$

  is known as the *stopband attenuation.* The edge (endpoint) of the stopband ω_s is the lowest frequency for which $|H(e^{j\omega})| = A\epsilon$.

- The *transition band* is the interval (ω_p, ω_s) separating the passband from the stopband.

(Note that an ideal filter has $\delta = 0$, $\epsilon = 0$, $\omega_p = \omega_s = \omega_c$, and an empty transition band.)

The parameters of other filter types (highpass, bandpass and bandstop) are defined similarly.

4.5.4 Practical FIR Filters

We began our discussion of FIR filters in Section 4.3 by examining the the second-order filter

$$y[n] = x[n] - x[n-1] + x[n-2] \ , \qquad n \in \mathbf{Z}$$

whose frequency response

$$H(e^{j\omega}) = e^{-j\omega}(2\cos\omega - 1)$$

crudely approximates that of a bandstop filter. To obtain filters with good frequency responses, it is necessary to combine more values of the input signal, i.e., use a higher filter order M. There are several design techniques for frequency-selective FIR filters, the details of which are beyond the scope of our discussion. In the case where M is even-valued, the resulting impulse response sequence

$$h[n] = \begin{cases} b_n, & 0 \le n \le M; \\ 0, & \text{otherwise.} \end{cases}$$

resembles the impulse response sequence of an ideal filter over the time interval $n = -M/2$ and $n = M/2$, delayed in time by $M/2$ units. Odd values of M are also used in practice.

The coefficient vectors of FIR filters used for frequency selection typically have symmetry about $n = M/2$, i.e.,

$$b_n = b_{M-n} \tag{4.11}$$

(This feature was also present in the second-order filter introduced in Section 4.3). As a result, their impulse response sequences are also symmetric about $n = M/2$:

$$h[n] = h[M-n]$$

Figure 4.18 shows the impulse response of a lowpass filter of order $M = 17$, which is symmetric about $n = 17/2$.

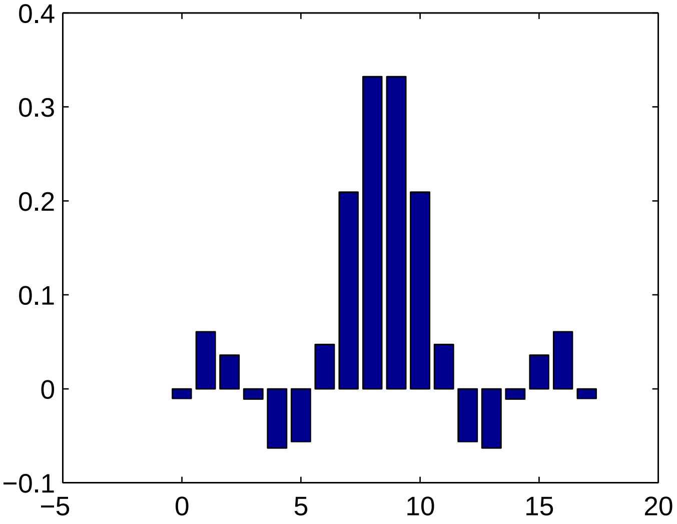


Figure 4.18: Impulse response of a lowpass filter of order $M = 17$.

The frequency response of an FIR filter satisfying (4.11) can be always written in the form

$$H(e^{j\omega}) = e^{-j(M\omega/2)}F(\omega) \tag{4.12}$$

where $F(\omega)$ is a real-valued function expressible as a sum of cosines. We illustrate this property using two examples.

Example 4.5.1. Consider the FIR filter with coefficient vector

$$\mathbf{b} = \begin{bmatrix} 1 & -2 & 3 & -2 & 1 \end{bmatrix}^T$$

Here $M = 4$ (even). The filter frequency response is given by

$$\begin{aligned} H(e^{j\omega}) &= 1 - 2e^{-j\omega} + 3e^{-j2\omega} - 2e^{-j3\omega} + e^{-j4\omega} \\ &= e^{-j2\omega}\left(e^{j2\omega} - 2e^{j\omega} + 3 - 2e^{-j\omega} + e^{-j2\omega}\right) \\ &= e^{-j2\omega}\left(3 - 2(e^{j\omega} + e^{-j\omega}) + (e^{j2\omega} + e^{-j2\omega})\right) \\ &= e^{-j2\omega}(3 - 4\cos\omega + 2\cos 2\omega) \end{aligned}$$

and thus

$$F(\omega) = 3 - 4\cos\omega + 2\cos 2\omega \qquad \square$$

Example 4.5.2. In this case, $M = 5$ (odd). The coefficient vector is given by

$$\mathbf{b} = \begin{bmatrix} 1 & -1 & 2 & 2 & -1 & 1 \end{bmatrix}^T$$

and the filter frequency response is given by

$$\begin{aligned} H(e^{j\omega}) &= 1 - e^{-j\omega} + 2e^{-j2\omega} + 2e^{-j3\omega} - e^{-j4\omega} + e^{-j5\omega} \\ &= e^{-j(5\omega/2)}(e^{j(5\omega/2)} - e^{j(3\omega/2)} + 2e^{j(\omega/2)} \\ &\qquad +2e^{-j(\omega/2)} - e^{-j(3\omega/2)} + e^{-j(5\omega/2)}) \\ &= e^{-j(5\omega/2)}\left(4\cos(\omega/2) - 2\cos(3\omega/2) + 2\cos(5\omega/2)\right) \end{aligned}$$

and thus

$$F(\omega) = 4\cos(\omega/2) - 2\cos(3\omega/2) + 2\cos(5\omega/2) \qquad \square$$

Equation (4.12) allows us to express the amplitude and phase responses as

$$\left|H(e^{j\omega})\right| = |F(\omega)| \qquad \text{and} \qquad \angle H(e^{j\omega}) = -\frac{M\omega}{2} + \begin{cases} 0, & F(\omega) \geq 0; \\ \pi, & F(\omega) < 0. \end{cases}$$

Note that the phase response is a linear function of ω with jumps of π occurring wherever $F(\omega)$ changes sign. Since $F(\omega)$ has no zeros over the passband, no such discontinuities occur there; thus the phase response is exactly linear. By the time delay property of the DTFT (Subsection 4.2.3), adding a linear function of frequency to the phase spectrum is equivalent to a delay in the time domain. Thus *all* frequency components of interest (i.e., those in the passband) are reconstructed at the output with the *same delay*, equal to $M/2$ time units. This is a very appealing property of FIR filters with symmetric coefficients. The other important class of filters (known as IIR, or *infinite impulse response*) exhibits a nonlinear phase response, resulting in sinusoidal components reappearing at the output with variable delays. This effect, known as *phase distortion*, can severely change the shape of a pulse and can be particularly noticeable in images.

4.5.5 Filter Transformations

Multiplication of a sequence by a complex sinusoid in the time domain results in a shift of its DTFT in frequency.

Fact. *(Multiplication by a Complex Sinusoid in Time Domain) If* $y[n] = e^{j\omega_0 n}x[n]$*, then* $Y(e^{j\omega}) = X(e^{j(\omega-\omega_0)})$.

This follows from the analysis equation:

$$\begin{aligned} Y(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} y[n]e^{-jn\omega} \\ &= \sum_{n=-\infty}^{\infty} x[n]e^{j\omega_0 n}e^{-jn\omega} \\ &= \sum_{n=-\infty}^{\infty} x[n]e^{-jn(\omega-\omega_0)} \\ &= X(e^{j(\omega-\omega_0)}) \end{aligned}$$

□

This property, applied to the impulse response sequence **h** and the frequency response $H(e^{j\omega})$, allows us to make certain straightforward transformations between frequency-selective filters of different types (lowpass, highpass, bandpass and bandstop). For example:

- Multiplication of $h[n]$ by $e^{j\pi n} = (-1)^n$ results in $H(e^{j\omega})$ being shifted in frequency by π radians—this transforms a lowpass filter into a highpass one, and vice versa.
- Multiplication of $h[n]$ by

  $$\cos(\omega_0 n) = \frac{e^{j\omega_0 n} + e^{-j\omega_0 n}}{2}$$

  produces a new frequency response equal to

  $$\frac{H(e^{j(\omega-\omega_0)}) + H(e^{j(\omega+\omega_0)})}{2}$$

  If the original filter is lowpass with passband and stopband edges at $\omega = \omega_p$ and $\omega = \omega_s$, respectively, then the above transformation will produce a bandpass filter provided $\omega_s < \omega_0 < \pi - \omega_s$. The center of the passband will be at $\omega = \omega_0$ and its width will be about $2\omega_p$.

These transformations are illustrated in Figure 4.19.

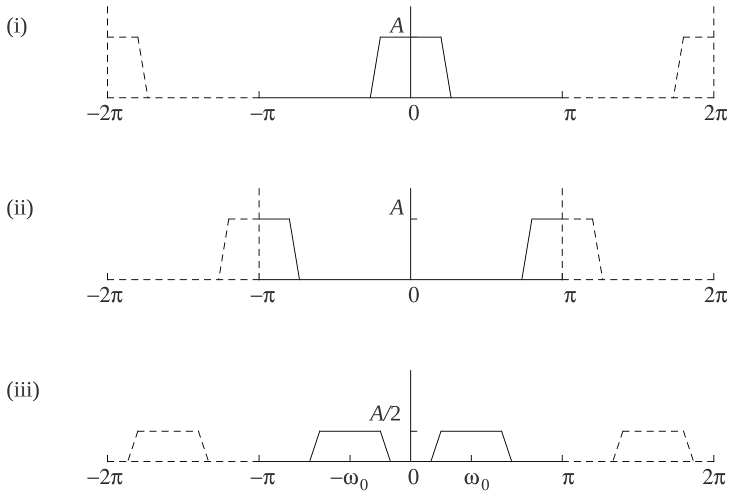


Figure 4.19: Multiplication of the impulse response by a sinusoid results in filter type transformation: lowpass (i) to highpass (ii); and lowpass (i) to bandpass (iii).

4.6 Time Domain Computation of Filter Response

4.6.1 Impulse Response

The input-output relationship

$$y[n] = b_0x[n] + b_1x[n-1] + \cdots + b_Mx[n-M] \tag{4.13}$$

of a finite impulse response filter allows us to directly compute the response $\mathbf{y}$ of the filter to an arbitrary input sequence $\mathbf{x}$. We have seen that for certain classes of input signals (notably sinusoids, exponentials and periodic signals), the output sequence can be computed efficiently using frequency-domain properties of the filter such as the system function $H(z)$ and the frequency response $H(e^{j\omega})$. The frequency domain-based approach is generally recommended for input signals whose spectra are discrete, or else have "nice" closed forms which enable us to invert the equation

$$Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$$

with ease. In most practical applications, the signals involved are not as structured, and it becomes necessary to use the equation (4.13) directly.

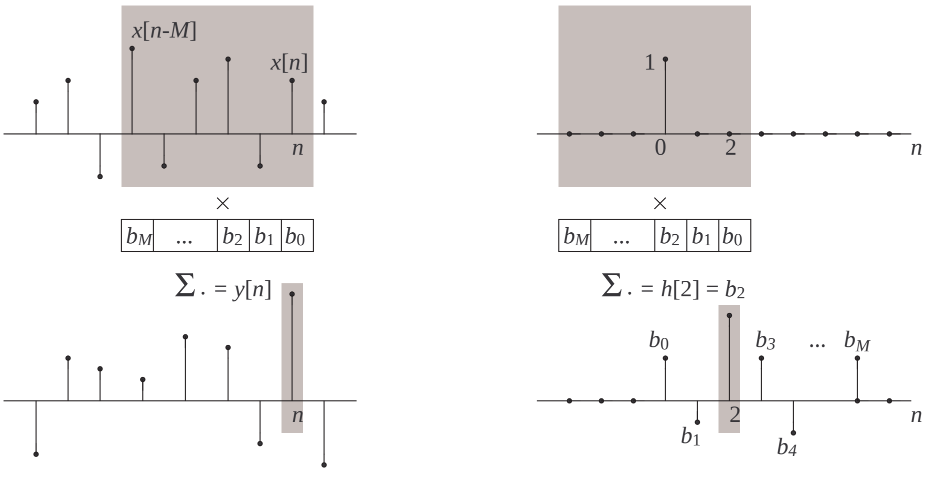
In Section 4.3, we described the operation of the FIR filter defined by (4.13) as follows: At each time n, the time-reverse of the vector $\mathbf{b}$ is aligned with a window of the input signal spanning time indices $n-M$ through n. The products $b_kx[n-k]$ are computed, then added together to produce the output sample $y[n]$. The time-reverse of $\mathbf{b}$ is then shifted to the right by one time index, a new window is formed, and the computation is repeated to yield $y[n+1]$, etc. This procedure is illustrated graphically in Figure 4.20 (i).

In our discussion of the FIR frequency response and system function, we defined the so-called *impulse response* of the filter as the sequence $\mathbf{h}$ formed by padding the vector $\mathbf{b}$ with infinitely many zeros on both sides. To see why that sequence was named so, consider an input signal consisting of a unit impulse at time zero:

$$x[n] = \delta[n]$$

This signal is plotted in the top graph of Figure 4.20 (ii). If $n < 0$, then indices $n-M$ through n are all (strictly) negative, and thus $x[n-M] = \ldots = x[n] = 0$. This implies that

$$y[n] = b_0x[n] + b_1x[n-1] + \cdots + b_Mx[n-M] = 0$$



(i) (ii)

Figure 4.20: (i) Filter response at time n formed by linear combination of input signal values in a sliding time window; (ii) Filter response to a unit impulse.

A zero value for $y[n]$ is also obtained for $n > M$, in which case indices $n - M$ through n are all (strictly) positive and the corresponding values of the input signal are all zero.

For $0 \leq n \leq M$, the window of input samples which are aligned with the (time-reversed) vector $\mathbf{b}$ includes the only nonzero value of the input, namely $x[0] = 1$. Then

$$y[n] = b_n x[0] = b_n$$

as illustrated in the bottom graph of Figure 4.20 (ii).

Thus the response of the filter to a unit impulse is given by

$$h[n] = \begin{cases} b_n, & 0 \leq n \leq M; \\ 0, & \text{otherwise.} \end{cases} \tag{4.14}$$

Clearly, the designation *finite impulse response* (FIR) is appropriate for this filter, since the duration of the impulse response sequence is finite (it begins at time 0 and ends at time M).

4.6.2 Convolution and Linearity

The definition of $\mathbf{h}$ in (4.14) allows us to rewrite the input-output relationship (4.13) as follows:

$$\begin{aligned} y[n] &= \sum_{k=0}^{M} b_k x[n-k] \\ &= \sum_{k=0}^{M} h[k]x[n-k] \\ &= \sum_{k=-\infty}^{\infty} h[k]x[n-k] \qquad (4.15) \end{aligned}$$

where the conversion to an infinite sum was possible because $h[k] = 0$ for $k < 0$ and $k > M$. Although only a finite number (i.e., $M+1$) of summands are involved here, the infinite sum in (4.15) gives us a general form for the input-output relationship of a linear filter which would also hold if the impulse response had infinite duration. (Such filters are beyond the scope of the present discussion.)

Definition 4.6.1. The (linear) *convolution* of sequences $\mathbf{h}$ and $\mathbf{x}$ is the sequence $\mathbf{y}$ denoted by

$$\mathbf{y} = \mathbf{h} * \mathbf{x}$$

and defined by

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] \qquad \square$$

In the above sum, k is a (dummy) variable of summation. At a given time n, the output is formed by summing together all products of the form $h[k]x[n-k]$ (note that the time indices in the two arguments always add up to n).

The concept of convolution was encountered earlier in its circular form, which involved two vectors of the same (finite) length. The relationship between linear and circular convolution will be explored in the next section. In the meantime, we note that linear convolution is *symmetric*:

$$\mathbf{h} * \mathbf{x} = \mathbf{x} * \mathbf{h}$$

or equivalently,

$$\sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

This can be shown using a change of variable from k to $k' = n - k$ in (4.15). As k ranges over all integers (for fixed n), so does k' (in reverse order), and thus

$$\sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k'=-\infty}^{\infty} h[n-k']x[k'] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

The last expression for the convolution sum, namely

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \ , \tag{4.16}$$

can be also obtained directly from the linearity of the filter. We saw earlier that the response of the FIR filter to the input $x[n] = \delta[n]$ is given by $h[n]$, defined in (4.14) above. By a similar argument, the response to a delayed impulse

$$x[n] = \delta[n-k] \qquad n \in \mathbf{Z}$$

is given by $h[n]$ delayed by k time instants, which is the signal $h[n-k]$.

We can express *any* input $\mathbf{x}$ as a linear combination of delayed impulses:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$$

By linearity of the filter, the response $\mathbf{y}$ to input $\mathbf{x}$ will be the linear combination of the responses to each of the delayed impulses in the sum, i.e.,

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

This is the same equation as (4.16) above.

4.6.3 Computation of the Convolution Sum

We now illustrate the computation of (4.15) in its simplest and most frequently encountered form, where both signal sequences $\mathbf{h}$ and $\mathbf{x}$ have finite duration. We will assume that $x[n] = 0$ for $n < 0$ or $n \geq L$; and similarly, $h[n] = 0$ for $n < 0$ or $n \geq P$ (thus the order of the FIR filter equals $M = P - 1$).

As we noted previously, in order to compute $y[n]$, we need to form the products $h[k]x[n-k]$ (where k ranges over all time indices), then sum them together. The easiest way to form $x[n-k]$ is to reverse $\mathbf{x}$ in time:

$$\tilde{x}[k] = x[-k] \ ;$$

then delay $\tilde{\mathbf{x}}$ by n time instants:

$$\tilde{x}[k-n] = x[n-k]$$

This procedure is depicted in Figure 4.21, where $x[n-k]$ is plotted (as a function of k) for different values of n. We note the following:

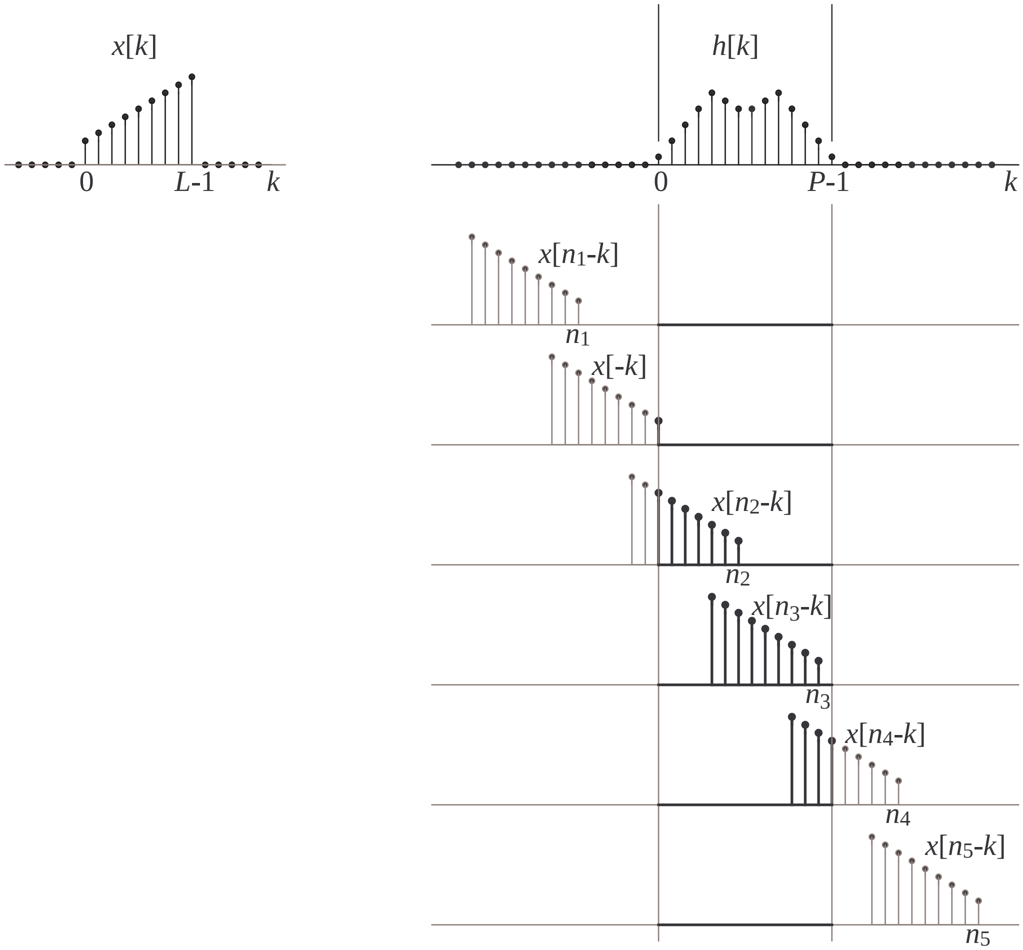


Figure 4.21: Illustration of the convolution of two finite-length sequences $\mathbf{h}$ and $\mathbf{x}$.

- For $n < 0$, the nonzero values of the sequence $x[n-k]$ do not overlap (in time) with those of $h[k]$. As a result, $h[k]x[n-k] = 0$ for all k, and

$y[n] = 0$. (This is expected since the impulse response is causal and the input is zero for negative time indices.)

- Similarly, no overlap occurs when the left edge of $x[n-k]$ follows the right edge of $h[k]$. This happens when $n-(L-1) > P-1$, i.e., when $n > P+L-2$.
- For $0 \le n \le P+L-2$, the nonzero segments of $x[n-k]$ and $h[k]$ overlap in varying degrees, and the resulting value of $y[n]$ is, in general, nonzero.

We summarize our conclusions as follows.

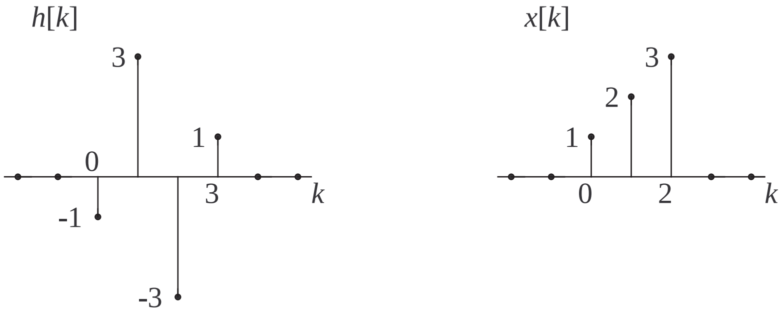
Fact. *The convolution* $\mathbf{y} = \mathbf{h} * \mathbf{x}$ *of two finite-duration sequences* $\mathbf{h}$ *and* $\mathbf{x}$ *is also a finite-duration sequence. If the nonzero segments of* $\mathbf{h}$ *and* $\mathbf{x}$ *begin at time* $n = 0$ *and have length* P *and* L*, respectively, then the nonzero segment of* $\mathbf{y}$ *also begins at time* $n = 0$ *and has length* $P+L-1$*.* □

Example 4.6.1. Consider the two sequences $\mathbf{h}$ and $\mathbf{x}$ given by

$$h[n] = -\delta[n] + 3\delta[n-1] - 3\delta[n-2] + \delta[n-3]$$

and

$$x[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2]$$



Example 4.6.1

Based on our earlier discussion, the nonzero segment of $\mathbf{y} = \mathbf{h} * \mathbf{x}$ begins at time $n = 0$ and has length $4+3-1 = 6$. Thus we only need to compute $y[n]$ for $n = 0, \ldots, 5$.

We follow the technique outlined earlier and illustrated in Figure 4.21. Instead of plotting, we tabulate (in rows) the values of $h[k]$ along with those $x[n-k]$ for each time index n of interest (i.e., $n = 0 : 5$). The products $h[k]x[n-k]$ are then computed and summed together for every value of n, and the resulting output value $y[n]$ is entered in the last column.

k	-2	-1	0	1	2	3	4	5	$y[n]$
$h[k]$			-1	3	-3	1			
$x[-k]$	3	2	1						-1
$x[1-k]$		3	2	1					1
$x[2-k]$			3	2	1				0
$x[3-k]$				3	2	1			4
$x[4-k]$					3	2	1		-7
$x[5-k]$						3	2	1	3

We thus obtain

$$y[0:5] = \begin{bmatrix} -1 & 1 & 0 & 4 & -7 & 3 \end{bmatrix}^T \qquad \Box$$

Remark. Since convolution is symmetric, the same result would have been obtained in the previous example by interchanging the signals $\mathbf{x}$ and $\mathbf{h}$.

4.6.4 Convolution and Polynomial Multiplication

The *z-transform* of a sequence $\mathbf{x}$ is a function of the complex variable $z \in \mathbf{C}$ defined by

$$X(z) = \sum_{k=-\infty}^{\infty} x[k]z^{-k}$$

provided, of course, that the infinite sum converges. If $\mathbf{x}$ has finite duration, then the sum consists of a finite number of nonzero terms and convergence is not an issue (except at $z = 0$). If $x[n] = 0$ for $n < 0$ and $n \geq L$ (as was the case in the previous section), then

$$X(z) = \sum_{k=0}^{L-1} x[k]z^{-k}$$

i.e., $X(z)$ is a polynomial of degree $L-1$ in the variable $z^{-1} = 1/z$.

Note that the z-transform was encountered earlier in Section 4.4: if $\mathbf{h}$ is a FIR filter response of order $M = P - 1$, then

$$H(z) = \sum_{k=0}^{P-1} h[k]z^{-k}$$

is the filter *system function*, which reduces to the filter frequency response when $z = e^{j\omega}$.

The convolution $\mathbf{y} = \mathbf{h} * \mathbf{x}$ mirrors the multiplication of the polynomials $X(z)$ and $H(z)$. To see this, write

$$\begin{aligned} H(z)X(z) &= \left(\sum_{k=0}^{P-1} h[k]z^{-k}\right)\left(\sum_{k=0}^{L-1} x[k]z^{-k}\right) \\ &= \left(h[0] + h[1]z^{-1} + \cdots + h[P-1]z^{-(P-1)}\right) \\ &\quad \cdot \left(x[0] + x[1]z^{-1} + \cdots + x[L-1]z^{-(L-1)}\right) \end{aligned}$$

and consider the z^{-n} term (where $0 \le n \le L + P - 2$) in the resulting product. This term is formed by adding together products of the form

$$h[k]z^{-k} \cdot x[n-k]z^{-(n-k)} = h[k]x[n-k]z^{-n}$$

Its coefficient is thus given by the sum

$$h[0]x[n] + h[1]x[n-1] + \cdots + h[n-1]x[1] + h[n]x[0]$$

which in turn equals

$$(\mathbf{h} * \mathbf{x})[n] = y[n]$$

We therefore have

$$Y(z) = \sum_{n=0}^{L+P-2} y[n]z^{-n} = H(z)X(z)$$

This result can be simply stated as follows.

Fact. *The z-transform of the filter response is given by the product of the filter system function and the z-transform of the input sequence.* □

A special case of this result is the input-output relationship

$$Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$$

in the frequency domain, which was encountered earlier. We note that

$$Y(z) = H(z)X(z)$$

also holds for infinite-duration sequences, provided the infinite sums involved converge.

Conclusion. Linear convolution of two sequences in the time domain is equivalent to multiplication (of their DTFT's or, more generally, their z-transforms) in the frequency domain. Thus linear convolution of sequences is analogous to circular convolution of vectors (which also corresponds to multiplication of DFT's in the frequency domain).

4.6.5 Impulse Response of Cascaded Filters

Consider two filters with impulse response sequences $\mathbf{h}^{(1)}$ and $\mathbf{h}^{(2)}$, and system functions $H_1(z)$ and $H_2(z)$, respectively. We have seen in Subsection 4.4.2 that the system function $H(z)$ of the cascade of two filters is given by

$$H(z) = H_1(z)H_2(z)$$

Thus we must also have

$$\mathbf{h} = \mathbf{h}^{(1)} * \mathbf{h}^{(2)}$$

i.e., the impulse response sequence of the cascade is given by the convolution of the two impulse response sequences.

The same result can be obtained using a time domain-based argument. Briefly, if the input to the cascade is a unit impulse, then the first filter will produce an output equal to $\mathbf{h}^{(1)}$. The second filter will act on $\mathbf{h}^{(1)}$ to produce a response equal to $\mathbf{h}^{(1)} * \mathbf{h}^{(2)}$, which is also the impulse response of the cascade. This is illustrated graphically in Figure 4.22.

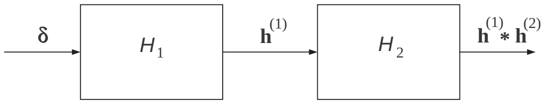


Figure 4.22: Impulse response of a cascade.

Example 4.6.2. Consider two FIR filters with impulse responses

$$h^{(1)}[n] = \delta[n] - 2\delta[n-1] - 2\delta[n-2] + \delta[n-3]$$

and

$$h^{(2)}[n] = \delta[n] + \delta[n-1] - 3\delta[n-2] + \delta[n-3] + \delta[n-4]$$

The impulse response $\mathbf{h}$ of the cascade can be computed directly in the time domain using the formula

$$\mathbf{h} = \mathbf{h}^{(1)} * \mathbf{h}^{(2)}$$

and the convolution technique explained in Subsection 4.6.3. Alternatively, we have

$$\begin{aligned} H_1(e^{j\omega}) &= 1 - 2e^{-j\omega} - 2e^{-j2\omega} + e^{-j3\omega} \\ H_2(e^{j\omega}) &= 1 + e^{-j\omega} - 3e^{-j2\omega} + e^{-j3\omega} + e^{-j4\omega} \end{aligned}$$

and therefore

$$\begin{aligned} H(e^{j\omega}) &= H_1(e^{j\omega})H_2(e^{j\omega}) \\ &= 1 - e^{-j\omega} - 7e^{-j2\omega} + 6e^{-j3\omega} + 6e^{-j4\omega} - 7e^{-j5\omega} - e^{-j6\omega} + e^{-j7\omega} \end{aligned}$$

The impulse response $\mathbf{h}$ of the cascade is read off the coefficients of $H(e^{j\omega})$:

$$\begin{aligned} h[n] &= \delta[n] - \delta[n-1] - 7\delta[n-2] + 6\delta[n-3] \\ &\quad + 6\delta[n-4] - 7\delta[n-5] - \delta[n-6] + \delta[n-7] \end{aligned}$$

In particular, the cascade is an FIR filter of order $M = 7$ with coefficient vector

$$\mathbf{b} = h[0:7] = [\ 1 \quad -1 \quad -7 \quad 6 \quad 6 \quad -7 \quad -1 \quad 1\ ]^T \qquad \square$$

As a closing remark, we note that the time domain and frequency domain formulas for cascaded filters developed here and in Subsection 4.4.2 can be generalized to an arbitrary number of filters connected in cascade.

4.7 Convolution in Practice

4.7.1 Linear Convolution of Two Vectors

In the previous section, we saw that the convolution of two finite-duration sequences also results in a sequence of finite duration. This allows us to extend the concept of (linear) convolution to vectors, by converting them to sequences of finite duration.

Suppose $\mathbf{b}$ and $\mathbf{s}$ are vectors of length P and L, respectively. Let

$$h[n] = \begin{cases} b[n], & 0 \le n \le P-1; \\ 0, & \text{otherwise.} \end{cases} \quad \text{and} \quad x[n] = \begin{cases} s[n], & 0 \le n \le L-1; \\ 0, & \text{otherwise.} \end{cases}$$

be the sequences obtained by padding $\mathbf{b}$ and $\mathbf{s}$ with infinitely many zeros on both sides. As we saw earlier,

$$\mathbf{y} = \mathbf{h} * \mathbf{x}$$

is a sequence of total duration $P + L - 1$ time instants:

$$y[n] = \begin{cases} c[n], & 0 \le n \le P+L-2; \\ 0, & \text{otherwise.} \end{cases}$$

The convolution of b and s is defined by

$$\mathbf{b} * \mathbf{s} = \mathbf{c} = y[0 : P+L-2]$$

Example 4.7.1. In Example 4.6.1,

$$\begin{aligned} \mathbf{b} &= \begin{bmatrix} -1 & 3 & -3 & 1 \end{bmatrix}^T \\ \mathbf{s} &= \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^T \end{aligned}$$

and

$$\mathbf{c} = \mathbf{b} * \mathbf{s} = \begin{bmatrix} -1 & 1 & 0 & 4 & -7 & 3 \end{bmatrix}^T \qquad \square$$

The vectors $\mathbf{b}$ and $\mathbf{s}$ often (though not always) have nonzero first and last entries. In such cases, the first and last entries of $\mathbf{c} = \mathbf{b} * \mathbf{s}$ are also nonzero, since

$$c[0] = b[0]s[0] \qquad \text{and} \qquad c[P+L-2] = b[P-1]s[L-1]$$

If any *single* endpoint (i.e., $b[0]$, $s[0]$, $b[P-1]$ or $s[L-1]$) is replaced by a zero, then the corresponding endpoint of $\mathbf{c}$ (i.e., $c[0]$ or $c[P+L-2]$) also becomes a zero. This leads to the following useful property:

Fact. *If $\mathbf{0}_i$ represents an all-zeros vector of length i, then (using the semicolon notation as in MATLAB)*

$$[\mathbf{0}_i\,;\,\mathbf{b}] * \mathbf{s} = [\mathbf{0}_i\,;\,\mathbf{b} * \mathbf{s}]$$

and

$$\mathbf{b} * [\mathbf{s}\,;\,\mathbf{0}_i] = [\mathbf{b} * \mathbf{s}\,;\,\mathbf{0}_i] \qquad \square$$

4.7.2 Linear Convolution versus Circular Convolution

In our discussion of the DFT, we introduced the *circular* convolution ($\circledast$) of two vectors of the same length N, and showed that circular convolution in the time domain is equivalent to (element-wise) multiplication of DFT's in the frequency domain. It turns out that *linear* convolution of two vectors can be also implemented by circular convolution, after zero-padding the two vectors to the appropriate output length (which is known in advance).

Fact. *If $\mathbf{b} = b[0 : P-1]$ and $\mathbf{s} = s[0 : L-1]$, then*

$$\mathbf{b} * \mathbf{s} = [\mathbf{b}\,;\,\mathbf{0}_{L-1}] \circledast [\mathbf{s}\,;\,\mathbf{0}_{P-1}] \qquad \square$$

This result is demonstrated in the arrays below for the case $P = 6$ and $L = 9$, where the output vector has length $P + L - 1 = 14$. Note that the same product terms $b[k]s[n-k]$ will be generated in corresponding rows of the two arrays. As before, zero entries are omitted.

Linear Convolution:

								b_0	b_1	b_2	b_3	b_4	b_5										
s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0														$\longrightarrow$	c_0
	s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0													$\longrightarrow$	c_1
		s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0												$\longrightarrow$	c_2
			s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0											$\longrightarrow$	c_3
				s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0										$\longrightarrow$	c_4
					s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0									$\longrightarrow$	c_5
						s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0								$\longrightarrow$	c_6
							s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0							$\longrightarrow$	c_7
								s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0						$\longrightarrow$	c_8
									s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0					$\longrightarrow$	c_9
										s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0				$\longrightarrow$	c_{10}
											s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0			$\longrightarrow$	c_{11}
												s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0		$\longrightarrow$	c_{12}
													s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0	$\longrightarrow$	c_{13}

Circular Convolution:

b_0	b_1	b_2	b_3	b_4	b_5										
s_0						s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	$\longrightarrow$	c_0
s_1	s_0						s_8	s_7	s_6	s_5	s_4	s_3	s_2	$\longrightarrow$	c_1
s_2	s_1	s_0						s_8	s_7	s_6	s_5	s_4	s_3	$\longrightarrow$	c_2
s_3	s_2	s_1	s_0						s_8	s_7	s_6	s_5	s_4	$\longrightarrow$	c_3
s_4	s_3	s_2	s_1	s_0						s_8	s_7	s_6	s_5	$\longrightarrow$	c_4
s_5	s_4	s_3	s_2	s_1	s_0						s_8	s_7	s_6	$\longrightarrow$	c_5
s_6	s_5	s_4	s_3	s_2	s_1	s_0						s_8	s_7	$\longrightarrow$	c_6
s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0						s_8	$\longrightarrow$	c_7
s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0						$\longrightarrow$	c_8
	s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0					$\longrightarrow$	c_9
		s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0				$\longrightarrow$	c_{10}
			s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0			$\longrightarrow$	c_{11}
				s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0		$\longrightarrow$	c_{12}
					s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0	$\longrightarrow$	c_{13}

The formal proof of this fact using the time-domain definitions of the linear and circular convolution is neither difficult nor particularly insightful. For a more interesting proof, we turn to the frequency domain. If, as before, **h**, **x** and **y** are the finite-duration sequences obtained by padding **b**, **s** and

$$\mathbf{c} = \mathbf{b} * \mathbf{s}$$

(respectively) with infinitely many zeros, then

$$\mathbf{y} = \mathbf{h} * \mathbf{x}$$

The DTFT's of the three sequences in the above equation are related by

$$Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega}) \tag{4.17}$$

We have

$$\begin{aligned} H(e^{j\omega}) &= \sum_{n=0}^{P-1} b[n]e^{-j\omega n} \\ X(e^{j\omega}) &= \sum_{n=0}^{L-1} s[n]e^{-j\omega n} \\ Y(e^{j\omega}) &= \sum_{n=0}^{P+L-2} c[n]e^{-j\omega n} \end{aligned}$$

In Subsection 4.2.2, we saw that sampling the DTFT of a finite-duration sequence at frequencies

$$\omega_k = \frac{2\pi k}{N} \ , \quad k = 0, \ldots, N-1$$

yields the DFT of $\mathbf{s}$ zero-padded to length N, provided, of course, that N is greater than or equal to the length of $\mathbf{s}$. Taking $N = P + L - 1$ (the size of the longest vector in this case), we have

$$\begin{aligned} H(e^{j(2\pi/N)k}) &= k^{\text{th}} \text{ entry in the DFT of } [\mathbf{b}\,;\,\mathbf{0}_{L-1}] \\ X(e^{j(2\pi/N)k}) &= k^{\text{th}} \text{ entry in the DFT of } [\mathbf{s}\,;\,\mathbf{0}_{P-1}] \\ Y(e^{j(2\pi/N)k}) &= k^{\text{th}} \text{ entry in the DFT of } \mathbf{c} \end{aligned}$$

From (4.17), we conclude that the DFT of $\mathbf{c}$ is given by the element-wise product of the DFT of $[\mathbf{b}\,;\,\mathbf{0}_{L-1}]$ and that of $[\mathbf{s}\,;\,\mathbf{0}_{P-1}]$. From DFT 9, it follows that the time-domain vectors must satisfy

$$\mathbf{c} = [\mathbf{b}\,;\,\mathbf{0}_{L-1}] \circledast [\mathbf{s}\,;\,\mathbf{0}_{P-1}]$$

Remark. If $\mathbf{b}$ and $\mathbf{s}$ are zero-padded to a total length $N > P + L - 1$, then circular convolution will result in $\mathbf{c}$ being padded with $N-(P+L-1)$ zeros.

The equivalence of linear and circular convolution for finite vectors gives us an option of performing this operation in the frequency domain, by computing the DFT's of the two vectors involved (zero-padded to the appropriate length), then inverting the element-wise product of the DFT's. This approach was illustrated in Section 4.4, where we computed the response of an FIR filter to a periodic input using (in effect) a circular convolution implemented via DFT's. As it turns out, implementation via DFT's can be greatly advantageous in practice. This is because the number of floating point operations required to convolve two vectors of length N in the time domain is of the order of N^2; while the DFT and its inverse can be implemented using *fast Fourier transform (FFT)* algorithms, for which the number of such operations is only of the order of $N \log_2 N$. For large values of N, the frequency-domain approach can be much faster.

In MATLAB, the standard command for convolving two vectors `b` and `s` is

```
c = conv(b,s)
```

which is implemented entirely in the time domain. An equivalent frequency-domain implementation using FFT's is

```
N = length(b) + length(s) - 1;
B = fft(b,N);
S = fft(s,N);
C = B.*S;
c = ifft(C)
```

4.7.3 Circular Buffers for Real-Time FIR Filters

In many practical applications, a linear filter will operate over long periods of time, processing input signals which are much longer than the filter's impulse response. If the filter is part of a system operating in real time (e.g., a digital speech encoder used during a telephone conversation), the delay in generating output samples shouldn't—and needn't—be too long. The current output $y[n]$ can be computed as soon as the current input $x[n]$ becomes available using the input-output relationship

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] \tag{4.18}$$

Clearly, the M previous input values must also be available at any given time, since computation of $y[n]$ requires knowledge of the vector $x[n-M:n]$.

The subsequent filter output $y[n+1]$ is computed using the vector $x[n-M+1:n+1]$. Both $x[n-M:n]$ and $x[n-M+1:n+1]$ contain the subvector $x[n-M+1:n]$, and differ only in one element:

$$x[n-M:n] = \begin{bmatrix} x[n-M] & ; & x[n-M+1:n] \end{bmatrix}$$

while

$$x[n-M+1:n+1] = \begin{bmatrix} x[n-M+1:n] & ; & x[n+1] \end{bmatrix}$$

Using signal permutations, $x[n-M+1:n+1]$ can be generated by applying a *left* circular shift on $x[n-M:n]$:

$$\mathbf{P}^{-1}x[n-M:n] = \begin{bmatrix} x[n-M+1:n] & ; & x[n-M] \end{bmatrix}$$

followed by replacing the last entry $x[n-M]$ by $x[n+1]$.

The use of a circular shift in updating the vector of input values suggests a computationally efficient implementation of an FIR filter in terms of a so-called *circular buffer*. The circular buffer is a vector of size $M+1$ which at any particular time n holds $\mathbf{P}^r x[n-M:n]$, i.e., a circularly shifted version of $x[n-M:n]$. A pointer i_{now} gives the position of $x[n]$ in the circular buffer; it is not difficult to see that if the buffer is indexed by $1:M+1$, then $r = i_{\text{now}}$. At time $n+1$, the buffer is updated as follows:

- the pointer i_{now} is incremented circularly by one (position); and
- the input $x[n-M+1]$ at that position is replaced by the current input $x[n+1]$.

After the update, the buffer holds $\mathbf{P}^{r+1}x[n-M+1:n+1]$.

Figure 4.23 illustrates the sequential updating of the circular buffer in the case $M = 6$, assuming all inputs before $n = 0$ were identically equal to zero. The initial position of $x[0]$ (obtained by initialization of i_{now}) is arbitrary.

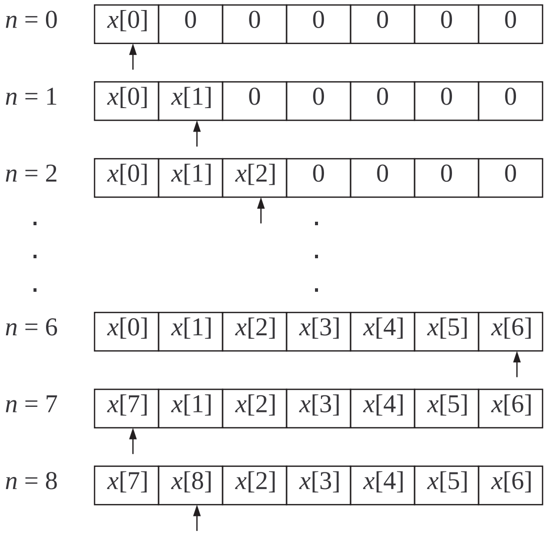

n = 0 | x[0] | 0 | 0 | 0 | 0 | 0 | 0
n = 1 | x[0] | x[1] | 0 | 0 | 0 | 0 | 0
n = 2 | x[0] | x[1] | x[2] | 0 | 0 | 0 | 0
n = 6 | x[0] | x[1] | x[2] | x[3] | x[4] | x[5] | x[6]
n = 7 | x[7] | x[1] | x[2] | x[3] | x[4] | x[5] | x[6]
n = 8 | x[7] | x[8] | x[2] | x[3] | x[4] | x[5] | x[6]

Figure 4.23: A circular buffer of size $P = M + 1 = 7$ used for computing $y[0 : 8]$. For each value of n, the arrow (same as the pointer i_{now}) indicates the current input $x[n]$.

The filter output (4.18) at time n is the (unconjugated) inner product of the coefficient vector $\mathbf{b}$ with the contents of the circular buffer read in *reverse* circular order starting with the current input sample $x[n]$.

4.7.4 Block Convolution

In applications where longer delays between input and output can be tolerated, it may be advantageous to filter the input signal in a block-wise fashion, i.e., by convolving consecutive blocks (segments) of the input signal with the filter impulse response. Convolving long vectors using frequency domain-based algorithms such as the FFT is generally more efficient than following the sliding window approach (in the time domain) discussed in the previous subsection.

To develop a model for the block-wise implementation of convolution, consider an FIR filter of order $M = P-1 > 0$ with coefficient vector $\mathbf{b}$. The filter acts on an input vector $\mathbf{s}$, which has been partitioned into J *blocks* of length L:

$$\mathbf{s} = [\mathbf{s}^{(0)} \ ; \ \mathbf{s}^{(1)} \ ; \ \ldots \ ; \ \mathbf{s}^{(J-1)}]$$

The block length L is a design parameter chosen according to factors such as processing speed and acceptable delay between input and output. Here we will assume that $L \geq P$. Note that we have also taken the length of $\mathbf{s}$ as an exact multiple of L (zero-padding may be used for the last block if necessary).

By linearity and time-invariance of the filter, the output

$$\mathbf{c} = \mathbf{b} * \mathbf{s}$$

is the superposition (i.e., sum) of the responses to each of the input blocks $\mathbf{s}^{(j)}$, where each response is delayed in time by the appropriate index, namely jL. Since the response to $\mathbf{s}^{(j)}$ has a total duration of $L+P-1 > L$ time units, it follows that the responses overlap in time. Specifically, for $0 < j \leq J-1$, the vector $c[jL : jL+P-2]$ is formed by summing together parts of two vectors, namely the time-shifted responses to $\mathbf{s}^{(j-1)}$ and $\mathbf{s}^{(j)}$.

The underlying concept of superposition is illustrated graphically in Figure 4.24 for the case $J = 2$.

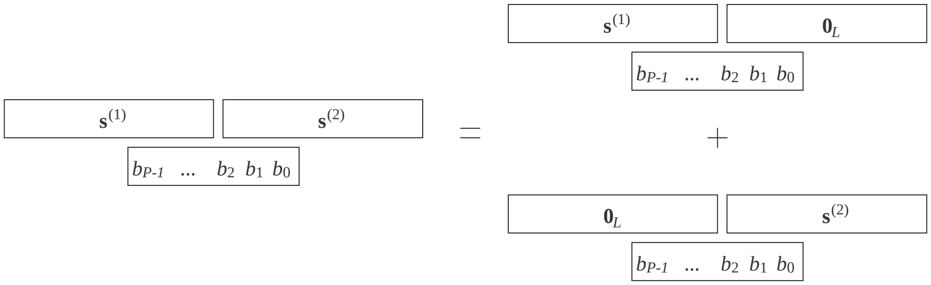


Figure 4.24: Linearity and time-invariance applied to block convolution.

Example 4.7.2. Consider the FIR filter with coefficient vector

$$\mathbf{b} = [\ 1 \quad -3 \quad -3 \quad 1\ ]^T$$

driven by the finite-duration input sequence $\mathbf{x}$, where

$$x[0:11] = \mathbf{s} = [\ 1 \quad -2 \quad 3 \quad 4 \quad 4 \quad -2 \quad 7 \quad 5 \quad 1 \quad 3 \quad -1 \quad -4\ ]^T$$

and $x[n] = 0$ for $n < 0$ and $n > 11$.

Suppose we wish to compute the filter response $\mathbf{y}$ to $\mathbf{x}$ using block convolution with block length $L = 6$. We express $\mathbf{s}$ as

$$\mathbf{s} = [\mathbf{s}^{(1)}\,;\,\mathbf{s}^{(2)}] = [\mathbf{s}^{(1)}\,;\,\mathbf{0}_6] + [\mathbf{0}_6\,;\,\mathbf{s}^{(2)}]$$

where

$$\mathbf{s}^{(1)} = [\ 1 \quad -2 \quad 3 \quad 4 \quad 4 \quad -2\ ]^T$$

and

$$\mathbf{s}^{(2)} = [\ 7 \quad 5 \quad 1 \quad 3 \quad -1 \quad -4\ ]^T$$

Then

$$\begin{aligned}
\mathbf{b} * [\mathbf{s}^{(1)}\,;\,\mathbf{0}_6] &= [\mathbf{b} * \mathbf{s}^{(1)}\,;\,\mathbf{0}_6] \\
&= [\ 1 \quad -5 \quad 6 \quad 2 \quad -19 \quad -23 \quad -2 \quad 10 \quad -2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\ ]^T
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{b} * [\mathbf{0}_6\,;\,\mathbf{s}^{(2)}] &= [\mathbf{0}_6\,;\,\mathbf{b} * \mathbf{s}^{(2)}] \\
&= [\ 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 7 \quad -16 \quad -35 \quad -8 \quad -8 \quad -9 \quad 18 \quad 11 \quad -4\ ]^T
\end{aligned}$$

We thus have

$$\begin{aligned}
\mathbf{b} * \mathbf{s} &= \mathbf{b} * [\mathbf{s}^{(1)}\,;\,\mathbf{0}_6] + \mathbf{b} * [\mathbf{0}_6\,;\,\mathbf{s}^{(2)}] \\
&= [\ 1 \quad -5 \quad 6 \quad 2 \quad -19 \quad -23 \quad 5 \quad -6 \quad -37 \quad -8 \quad -8 \quad -9 \quad 18 \quad 11 \quad -4\ ]^T
\end{aligned}$$

and the filter response is given by

$$y[n] = \begin{cases} (\mathbf{b} * \mathbf{s})[n], & 0 \le n \le 14; \\ 0, & \text{otherwise.} \end{cases}$$

□

Problems

Sections 4.1–4.2

***P* 4.1.** **(i)** Sketch the signal sequence

$$x[n] = \delta[n+2] - \delta[n+1] + 5\delta[n] - \delta[n-1] + \delta[n-2]$$

and write an expression for its discrete-time Fourier transform $X(e^{j\omega})$. Show that $X(e^{j\omega})$ can be written as a sum of cosines with real coefficients.

(ii) Generalize the result of part (i) to a symmetric signal sequence of finite duration ($2L+1$ samples): If

$$x[n] = x[-n] = \begin{cases} \beta_{|n|}, & |n| \le L \\ 0, & |n| > L \end{cases}$$

express $X(e^{j\omega})$ as a sum of L cosines plus a constant.

***P* 4.2.** **(i)** Sketch the signal sequence

$$x[n] = \begin{cases} 1, & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

and express its Fourier transform $X(e^{j\omega})$ in the form

$$e^{jK\omega} F(\omega)$$

where $F(\omega)$ is a real-valued function which is symmetric about $\omega = 0$. (*Hint*: Consult Subsection 3.8.2.)

(ii) Let $M = 23$. How would you define an array x in MATLAB, so that the command

```
X = fft(x,500)
```

computes $X(e^{j\omega})$ at 500 equally spaced frequencies ω ranging from 0 to 1.996π, inclusive?

***P* 4.3.** Consider the signal sequence **x** defined by

$$x[n] = \cos\left(\frac{3\pi n}{14} - 1.8\right) + 2\cos\left(\frac{18\pi n}{35} - 0.7\right) + 6\cos\left(\frac{17\pi n}{24} + 2.0\right)$$

(i) Is the sequence periodic, and if so, what is its period?

(ii) Sketch the amplitude and phase spectra of **x** (both of which are line spectra).

Sections 4.3–4.4

***P* 4.4.** Consider the FIR filter whose input $\mathbf{x}$ and output $\mathbf{y}$ are related by

$$y[n] = x[n] - x[n-1] - x[n-2] + x[n-3]$$

(i) Write out an expression for the system function $H(z)$.

(ii) Express $|H(e^{j\omega})|^2$ in terms of cosines only. Plot $|H(e^{j\omega})|$ as a function of ω.

(iii) Determine the output $y[n]$ when the input sequence $\mathbf{x}$ is given by each of the following expressions (where $n \in \mathbf{Z}$):

- $x[n] = 1$
- $x[n] = (-1)^n$
- $x[n] = e^{j\pi n/4}$
- $x[n] = \cos(\pi n/4 + \phi)$
- $x[n] = 2^{-n}$
- $x[n] = 2^{-n}\cos(\pi n/4)$

(In all cases except the third, your answer should involve real-valued terms only.)

***P* 4.5.** Consider the FIR filter

$$y[n] = x[n] - 3x[n-1] + x[n-2] + x[n-3] - 3x[n-4] + x[n-5]$$

(i) Write MATLAB code which includes the function `fft`, and which computes the amplitude and phase response of the filter at 256 equally spaced frequencies between 0 and $2\pi(1 - 256^{-1})$.

(ii) Express the frequency response of the filter in the form

$$e^{-j\alpha\omega}F(\omega)$$

where $F(\omega)$ is a real-valued sum of cosines.

(iii) Determine the response $y[n]$ of the filter to the exponential input sequence

$$x[n] = \left(\frac{1}{2}\right)^n, \qquad n \in \mathbf{Z}$$

***P* 4.6.** The MATLAB code

```
a = [ 1 -3 5 -3 1 ].’ ;
H = fft(a,500);
A = abs(H);
q = angle(H);
```

computes the amplitude response `A` and phase response `q` of a FIR filter over 500 equally spaced frequencies in the interval $[0, 2\pi)$.

(i) If $\mathbf{x}$ and $\mathbf{y}$ are (respectively) the input and output sequences of that filter, write an expression for $y[n]$ in terms of values of $\mathbf{x}$.

(ii) Determine the output $\mathbf{y}$ of the filter when the input $\mathbf{x}$ is given by

$$x[n] = \left(\frac{1}{3}\right)^n , \qquad n \in \mathbf{Z}$$

(iii) Express the frequency response of the filter in the form

$$e^{-j\alpha\omega} F(\omega)$$

where $F(\omega)$ is a real-valued sum of cosines.

***P* 4.7.** Consider a FIR filter whose input $\mathbf{x}$ and output $\mathbf{y}$ are related by

$$y[n] = \sum_{k=0}^{8} b_k x[n-k] \ ,$$

where the coefficient vector $\mathbf{b}$ is given by

$$\mathbf{b} = [\ 1 \quad 2 \quad -2 \quad -1 \quad 4 \quad -1 \quad -2 \quad 2 \quad 1\ ]^T$$

Let the input $\mathbf{x}$ be an infinite periodic sequence with period $L = 7$, and such that

$$x[0:6] = [\ 1 \quad -1 \quad 0 \quad 3 \quad 1 \quad -2 \quad 0\ ]^T$$

Using DFT’s (and MATLAB), determine the first period $y[0 : 6]$ of the output sequence $\mathbf{y}$.

***P* 4.8.** Consider two FIR filters with coefficient vectors $\mathbf{b}$ and $\mathbf{c}$, where

$$\mathbf{b} = \begin{bmatrix} 3 & 2 & 1 & 2 & 3 \end{bmatrix}^T$$

and

$$\mathbf{c} = \begin{bmatrix} 1 & -2 & 2 & -1 \end{bmatrix}^T$$

(i) Determine the system function $H(z)$ of the cascade. Is the cascade also a FIR filter? If so, determine its coefficient vector.

(ii) Express the amplitude response of the cascade as a sum of sines or cosines (as appropriate) with real-valued coefficients.

***P* 4.9.** Consider the FIR filter with coefficient vector

$$\mathbf{b} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$$

Two copies of this filter are connected in series (cascade).

(i) Determine the system function $H(z)$ of the cascade. Is the cascade also a FIR filter? If so, determine its coefficient vector.

(ii) Determine the response $y[n]$ of the cascade to the sinusoidal input sequence

$$x[n] = \cos\left(\frac{n\pi}{2}\right) , \qquad n \in \mathbf{Z}$$

Section 4.5

***P* 4.10.** The text file `s6.txt` contains the coefficient vector of a FIR filter of order $M = 40$.

(i) Plot the impulse response of the filter (i.e., the vector in `s6.txt`) using the `STEM` or `BAR` functions in MATLAB. What kind of symmetry does the impulse response exhibit?

(ii) Plot $|H(e^{j\omega})|$, i.e., the magnitude of the frequency response of the filter. What type of filter (lowpass, highpass, bandpass or bandstop) do we have here? Determine the edges of the passband and stopband.

(iii) If

$$\begin{aligned} A_1 &= \text{maximum value of } |H(e^{j\omega})| \text{ over the passband} \\ A_2 &= \text{minimum value of } |H(e^{j\omega})| \text{ over the passband} \\ A_3 &= \text{maximum value of } |H(e^{j\omega})| \text{ over the stopband} \end{aligned}$$

compute the ratios A_1/A_2 and A_2/A_3 (using your graphs).

(iv) How would you convert this filter to a bandpass filter whose passband is centered around $\omega = \pi/2$? What would be resulting width of the passband? Verify your answers using MATLAB.

***P* 4.11.** Consider the signal sequence $\mathbf{x}$ given by

$$x[n] = 2\cos(3.0n - 1.7) + \cos(1.4n + 0.8)\ , \qquad n \in \mathbf{Z}$$

(i) Is $\mathbf{x}$ periodic? If so, what is its period?

(ii) If $\mathbf{x}$ is the input to the linear filter with frequency response described by

$$H(e^{j\omega}) = \begin{cases} 2e^{-j3\omega}, & 2\pi/3 \leq \omega \leq 4\pi/3; \\ 0, & \text{all other } \omega \text{ in } [0, 2\pi), \end{cases}$$

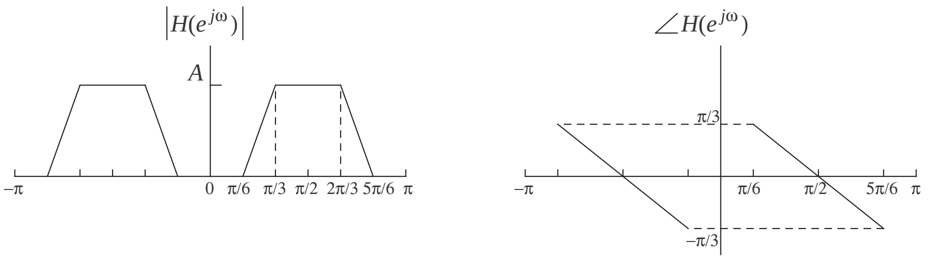
determine the output sequence $\mathbf{y}$.

***P* 4.12.** Consider the signal

$$x[n] = 3\cos\left(\frac{\pi n}{15} - \frac{\pi}{3}\right) + 5\cos\left(\frac{\pi n}{4} + \frac{\pi}{2}\right)$$

(i) Is the signal periodic, and if so, what is its period?

(ii) If $x[n]$ is the input to a filter with amplitude and phase responses as shown in figure, determine the resulting output $y[n]$.



Problem P 4.12

Section 4.6

***P* 4.13.** Consider the FIR filter with impulse response $\mathbf{h}$ given by

$$h[n] = \delta[n] - 2\delta[n-1] + 3\delta[n-3] - 2\delta[n-4]$$

(i) Without using convolution, determine the response $\mathbf{y}$ of the filter to the input signal $\mathbf{x}$ given by

$$x[n] = \delta[n+1] - \delta[n-1]$$

(*Hint:* Express $\mathbf{y}$ in terms of $\mathbf{h}$.)

(ii) Now let the input signal $\mathbf{x}$ be given by

$$x[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2] - \delta[n-3]$$

Using convolution, determine the output signal $\mathbf{y}$.

(iii) Obtain the answer to **(ii)** by direct multiplication of the z-transforms $H(z)$ and $X(z)$.

***P* 4.14.** Consider the signal sequence

$$x[n] = \begin{cases} 1, & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

of Problem P 4.2.

(i) Determine the convolution

$$\mathbf{y} = \mathbf{x} * \mathbf{x}$$

For what values of n is $y[n]$ nonzero?

(ii) Using the results of Problem P 4.2, write an equation for the DTFT $Y(e^{j\omega})$ of $\mathbf{y}$.

(iii) By how many instants should $\mathbf{y}$ be delayed (or advanced) in time so that the resulting signal is symmetric about $n = 0$? What is the DTFT of that signal?

***P* 4.15.** Define $\mathbf{x}$ by

$$x[n] = \delta[n] - \delta[n-1] + \delta[n-2] - \delta[n-3]$$

and let $\mathbf{h}$ be the (*infinite* impulse response) sequence given by

$$h[n] = \begin{cases} \alpha^n, & n \ge 0; \\ 0, & n < 0. \end{cases}$$

Compute $\mathbf{y} = \mathbf{x} * \mathbf{h}$. Sketch your answer in the case $\alpha = 2$.

***P* 4.16.** Consider the filter with impulse response given by

$$h[n] = \begin{cases} 1, & 0 \le n \le 8 \\ 0, & \text{otherwise} \end{cases}$$

(i) Determine the response of the filter to the input

$$x^{(1)}[n] = \cos\left(\frac{n\pi}{2} + \frac{\pi}{8}\right) , \qquad n \in \mathbf{Z}$$

(ii) Determine the output of the filter to the input

$$x^{(2)}[n] = \delta[n] - \delta[n-1] + \delta[n-2]$$

***P* 4.17.** Two FIR filters whose impulse response sequences $\mathbf{h}^{(1)}$ and $\mathbf{h}^{(2)}$ are given by

$$h^{(1)}[n] = \delta[n] + \delta[n-1] + \delta[n-2] + \delta[n-3] , \qquad n \in \mathbf{Z}$$

and

$$h^{(2)}[n] = \delta[n] - \delta[n-2] , \qquad n \in \mathbf{Z}$$

are connected in series (cascade) to form a single filter with impulse response $\mathbf{h}$.

(i) Determine the corresponding system functions $H^{(1)}(z)$, $H^{(2)}(z)$ and $H(z)$.

(ii) Determine $\mathbf{h}$ (i.e., $h[n]$ for every $n \in \mathbf{Z}$).

(iii) Sketch the output $\mathbf{y}$ of the cascade when the input $\mathbf{x}$ is given by

$$x[n] = \delta[n+1] , \qquad n \in \mathbf{Z}$$

Section 4.7

***P* 4.18.** Consider the convolution of $\mathbf{b} = b[0 : P-1]$ and $\mathbf{s} = s[0 : L-1]$, where $\mathbf{b}$ represents a FIR coefficient vector and $\mathbf{s}$ represents the filter input.

(i) Construct a $(P + L - 1) \times L$ matrix $\mathbf{B}$ such that

$$\mathbf{b} * \mathbf{s} = \mathbf{Bs}$$

(In effect, every FIR filter can be viewed as a linear transformation of the input signal sequence. The matrix $\mathbf{B}$ describes this transformation in the case where the input signal has finite duration L. See also Problem P 2.8.)

(ii) Verify that the following MATLAB script generates the required matrix $\mathbf{B}$ from a column vector $\mathbf{b}$:

```
    c = [b ; zeros(L-1,1)];
    r = [b(1) zeros(1,L-1)];
    B = toeplitz(c,r);
```

Compare `B*s` and `conv(b,s)` for random choices of `s`.

***P* 4.19.** A circular buffer for an FIR filter of order $M = 16$ is initialized at time $n = 0$ by placing $x[0]$ in the first (leftmost or topmost) position within the buffer. Express the contents of the buffer at time $n = 139$ in the form

$$\left[\ x[n_1 : n_2] \quad ; \quad x[n_3 : n_4]\ \right]$$

***P* 4.20.** Consider the following incomplete MATLAB code, where `P` is a known (previously defined) integer and `b` is a known column vector or length `P`.

```
    iupdate = [2:P 1];
    ireverse = toeplitz(1:P, [1 P:-1:2]);
    xcirc = zeros(P,1);
    inow = P;
    xvec = [];
    yvec = [];
    while 1
        x = randn(1);
        xvec = [xvec ; x];
        %
        % computation of scalar variable y
        %
        yvec = [yvec ; y];
    end
```

This code simulates the continuous operation (note the `while` loop) of a FIR filter with coefficient vector `b`, driven by a sequence `xvec` of Gaussian noise. The scalar variables `x` and `y` denote the current input and output, and the output sequence is given by `yvec`. Complete the code using three commands (in the commented-out section) that contain no additional variables (i.e., other than previously defined), numbers or loops. Test your code against the results obtained using the function `CONV`.

***P* 4.21.** Consider the FIR filter

$$y[n] = x[n] - 2x[n-1] - x[n-2] + 4x[n-3] - x[n-4] - 2x[n-5] + x[n-6]$$

The response of the filter to the input

$$x[0:3] = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix}^T$$

($x[\cdot] = 0$ otherwise) is given by

$$y[0:9] = \begin{bmatrix} 1 & -1 & 0 & -7 & 8 & 13 & -20 & -1 & 11 & -4 \end{bmatrix}^T$$

($y[\cdot] = 0$ otherwise); while the response to

$$\tilde{x}[0:3] = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix}^T$$

($\tilde{x}[\cdot] = 0$ otherwise) is given by

$$\tilde{y}[0:9] = \begin{bmatrix} 1 & -2 & -2 & 10 & -8 & -10 & 18 & -2 & -9 & 4 \end{bmatrix}^T$$

($\tilde{y}[\cdot] = 0$ otherwise). Determine the response of the filter to

$$\hat{x}[0:7] = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & 2c_0 & 2c_1 & 2c_2 & 2c_3 \end{bmatrix}^T$$

($\hat{x}[\cdot] = 0$ otherwise).

***P* 4.22.** Consider the two vectors

$$\mathbf{b} = \begin{bmatrix} 2 & -3 & 4 & -5 & -1 & -1 \end{bmatrix}^T$$

and

$$\mathbf{s} = \begin{bmatrix} 2 & -3 & 0 & 7 & -3 & 4 & 1 & 5 \end{bmatrix}^T$$

(i) Determine the convolution $\mathbf{b} * \mathbf{s}$ using the function `CONV` in MATLAB.
(ii) Repeat using the functions `FFT` and `IFFT` instead of `CONV`.

***P* 4.23.** Consider the following MATLAB code:

```
a = [ 1 -2 3 -4 ].' ;
b = [ 1 2 -1 2 1].' ;
A = fft(a,8);
B = fft(b,8);
C = A.*B;
c = ifft(C);
```

(i) Without using MATLAB, determine the vector `c`.

(ii) Let $\mathbf{a} = x[0:3]$, where $\mathbf{x}$ is a sequence of period $L = 4$. The signal $\mathbf{x}$ is the input to a FIR filter with coefficient vector $\mathbf{b}$, resulting in an output sequence $\mathbf{y}$. How would you modify the code shown above so that $\mathbf{c} = y[0:3]$?

***P* 4.24.** Consider the following MATLAB code:

```
a = [ 2 -1 0 -1 2 ].' ;
b = [ 3 -2 5 1 -1].' ;
A = fft(a,9);
B = fft(b,9);
C = A.*B;
c = ifft(C);
```

(i) Without using MATLAB, determine the vector `c`.

(ii) Without performing a new convolution, determine the vector `e` obtained by the following code:

```
a = [ 2 -1 0 -1 2 ].' ;
d = [-3 2 -5 -1 1 3 -2 5 1 -1].' ;
A = fft(a,14);
D = fft(d,14);
E = A.*D;
e = ifft(E);
```

***P* 4.25.** The data file `s7.txt` contains a vector $\mathbf{s}$ of length $L = 64$. Let

$$\mathbf{b} = \begin{bmatrix} 2 & -1 & -3 & 1 & 3 & 1 & -3 & -1 & 2 \end{bmatrix}^T$$

(i) Use the `CONV` function in MATLAB to compute the convolution $\mathbf{c} = \mathbf{b} * \mathbf{s}$. What is the length of $\mathbf{c}$?

(ii) Compute $\mathbf{c}$ using three convolutions of output length $2^5 = 32$, followed by a summation. Your answer should be the same as for part **(i)**.

Epilogue

Our overview of signals took us from analog waveforms to discrete-time sequences (obtained by sampling) to finite-dimensional vectors. With the aid of basic tools and concepts from linear algebra—notably linear independence, Gaussian elimination, least-squares approximation (projection) and orthogonality—we developed the representation of an N-dimensional signal vector in terms of N sinusoidal components known as the discrete Fourier transform (DFT). The extension of vectors to sequences (by taking $N \to \infty$) brought the continuum of frequency into play and gave rise to the discrete-time Fourier transform (DTFT). In the second part of this book, we will develop similar tools for the analysis of analog waveforms and examine the implications of sampling from a frequency-domain viewpoint.

The traditional exposition of signal analysis begins with the sinusoidal representation of periodic analog waveforms known as the *Fourier series.* This representation is then extended to aperiodic waveforms (by taking a limit as the signal period tends to infinity), resulting in the so-called *Fourier transform.* Sequences and vectors are treated next, and the DTFT and DFT are developed using some of the ideas and results from Fourier series and the Fourier transform. Our approach in this book clearly follows a different order.

While both the Fourier series and the Fourier transform for analog waveforms will be developed in the second part of this book, the motivated reader might be interested in a preview of that development, which, as it turns out, is quite accessible at this point. It is also possible to provide here a definitive answer to a question raised in Section 1.6, namely on the sufficiency of Nyquist-rate sampling for the faithful reconstruction of bandlimited analog waveforms.

The Fourier Series

A continuous-time signal $\{s(t), t \in \mathbf{R}\}$ is periodic with period T_0 if

$$s(t + T_0) = s(t) \ , \qquad t \in \mathbf{R}$$

Associated with the period T_0 are two familiar parameters, the cyclic frequency $f_0 = 1/T_0$ (Hz) and the angular frequency $\Omega_0 = 2\pi/T_0$ (rad/sec).

The continuous-time sinusoids

$$\cos \Omega_0 t, \qquad \sin \Omega_0 t, \qquad \text{and} \qquad e^{j\Omega_0 t}$$

encountered in Chapter 1 are all periodic with period T_0. It follows that for any nonzero integer k, the sinusoids

$$\cos k\Omega_0 t, \qquad \sin k\Omega_0 t, \qquad \text{and} \qquad e^{jk\Omega_0 t}$$

are periodic with period $T_0/|k|$; and since T_0 is an integer multiple of $T_0/|k|$, they are also periodic with period T_0. These sinusoids are also referred to as *harmonics* of the basic sinusoid (real or complex) of frequency Ω_0.

The key result in Fourier series is that a "well-behaved" periodic signal $s(t)$ with period T_0 can be expressed as a linear combination of the sinusoids $e^{jk\Omega_0 t}$, all of which are periodic with period T_0. In other words,

$$s(t) = \sum_{k=-\infty}^{\infty} S_k e^{jk\Omega_0 t}, \qquad t \in \mathbf{R}$$

The coefficients S_k can be obtained from $s(t)$ by a suitable integral over a time interval of length equal to one period, i.e., T_0. Together with the corresponding frequencies $k\Omega_0$, these coefficients yield a discrete (line) spectrum $\{(k\Omega_0, S_k), \ k \in \mathbf{Z}\}$ for the periodic signal $s(t)$, as shown in Figure E.1. Note that the frequency Ω is in radians *per second* (as opposed to per sample), and that there is no periodicity in the spectrum (as was the case with discrete-time sequences).

As it turns out, this result can be established by considering the usual DTFT analysis and synthesis formulas for *discrete-time* signals. We know that the spectrum of a time-domain sequence is a periodic function of the frequency ω (measured in radians per sample), with period 2π. Thus given a spectrum $X(e^{j\omega})$, the time-domain sequence $\{x[n], \ n \in \mathbf{Z}\}$ can be recovered from $X(e^{j\omega})$ using the integral (synthesis equation)

$$x[n] = \frac{1}{2\pi} \int_0^{2\pi} X(e^{j\omega}) e^{j\omega n} \, d\omega \ ;$$

while $X(e^{j\omega})$ can be obtained from $\{x[n],\ n \in \mathbf{Z}\}$ using the sum (analysis equation)

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$$

The trick here is to associate the continuous frequency parameter ω with continuous *time*. To that end, we define

$$t \stackrel{\text{def}}{=} \omega/\Omega_0$$

(which has the correct units, namely seconds) and use the periodic frequency-domain signal $X(e^{j\omega})$ to create a periodic time-domain signal $s(t)$:

$$s(t) \stackrel{\text{def}}{=} X(e^{j\Omega_0 t}) = X(e^{j\omega})$$

Since $X(e^{j\omega})$ is an arbitrary signal of period 2π (in ω), $s(t)$ is an arbitrary signal of period $2\pi/\Omega_0 = T_0$ (in t). From the synthesis equation, we see that $s(t)$ can be associated with a sequence $\{S_k,\ k \in \mathbf{Z}\}$ defined by

$$S_k \stackrel{\text{def}}{=} x[-k]$$

and such that

$$\begin{aligned} S_k &= \frac{1}{2\pi} \int_0^{2\pi} X(e^{j\omega}) e^{-j\omega k}\, d\omega \\ &= \frac{\Omega_0}{2\pi} \int_0^{2\pi/\Omega_0} s(t) e^{-jk\Omega_0 t}\, dt \\ &= \frac{1}{T_0} \int_0^{T_0} s(t) e^{-jk\Omega_0 t}\, dt \end{aligned}$$

From the analysis equation, we then obtain

$$s(t) = X(e^{j\Omega_0 t}) = \sum_{k=-\infty}^{\infty} x[k] e^{-jk\Omega_0 t} = \sum_{k=-\infty}^{\infty} S_k e^{jk\Omega_0 t}$$

We have therefore established that the periodic signal $s(t)$ can be expressed as a *Fourier series*, i.e., a linear combination of complex sinusoids whose frequencies are multiples of Ω_0. (A Fourier series can be also regarded as a power series in $e^{j\Omega_0 t}$.) The k^{th} coefficient S_k is obtained by integrating, over one period, the product of $s(t)$ and the complex conjugate of the k^{th} sinusoid.

In using the DTFT to derive the Fourier series, we exploited the *duality* of time and frequency, which is a central theme in Fourier analysis. Duality was encountered earlier in the discussion of the DFT, where it was shown that the time and frequency domains had the same dimension (namely vector size N), and that the DFT and IDFT operations were essentially the same (with the exception of a scaling factor and a complex conjugate). In the case of the DTFT, the two domains are quite different: time n is a discrete parameter taking values in $\mathbf{Z}$, while frequency ω is a continuous parameter taking values in an interval of length 2π. Also, the time-domain signal is a sequence, while the frequency-domain signal is continuous-parameter periodic signal. In effect, the Fourier series is the *dual* of the DTFT where the two domains are interchanged: the time-domain signal is a continuous-parameter periodic signal, while the frequency-domain signal is a discrete sequence of coefficients (i.e., a line spectrum). Both the Fourier series and the DTFT use (essentially) the same sum and integral for transitioning between the two domains (discrete-to-continuous and continuous-to-discrete, respectively). Figure E.1 is a graphical illustration of the duality of the Fourier series and the DTFT.

The Fourier Transform

The Fourier transform is a frequency-domain representation for certain classes of aperiodic continuous-time signals. These signals are typically either *absolutely integrable* or *square-integrable*, i.e., satisfy

$$\int_{-\infty}^{\infty} |s(t)| dt < \infty \qquad \text{or} \qquad \int_{-\infty}^{\infty} |s(t)|^2 dt < \infty$$

(or both). Either type of integrability implies that the signal $s(t)$ decays sufficiently fast in t as $t \to \infty$ or $t \to -\infty$; and that if $s(t)$ is locally unbounded, it cannot approach infinity too fast.

Consider, for example, the three signals defined, for all $t \in \mathbf{R}$, by

$$e^{-|t|}, \qquad e^t \qquad \text{and} \qquad \cos t + \cos \pi t$$

The first signal satisfies both integrability conditions and has a Fourier transform, while the second signal violates both integrability conditions and does not have a Fourier transform. The third signal (which is aperiodic) violates both integrability conditions and does not have a Fourier transform in the conventional sense; yet it has an obvious frequency-domain representation in terms of a line spectrum (at frequencies $\Omega = \pm 1$ and $\Omega = \pm\pi$ rad/sec).

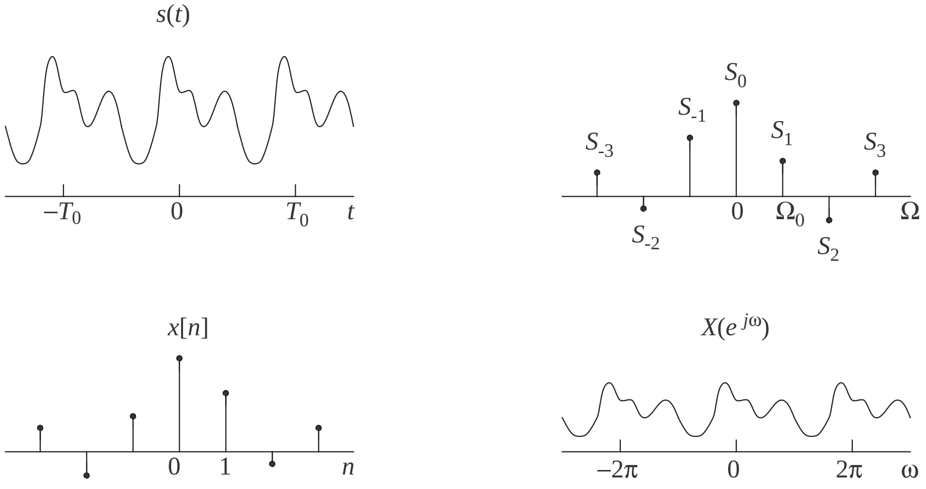


Figure E.1: A periodic waveform (top left) and its Fourier series coefficients (top right); a sample sequence (bottom left) and its discrete-time Fourier transform (bottom right). If the continuous-parameter graphs are scaled versions of each other, then the discrete-parameter graphs are also scaled and time-reversed versions of each other (duality of the Fourier series and the DTFT).

It is possible to obtain the Fourier transform of an aperiodic signal $\{s(t),\ t \in \mathbf{R}\}$ by considering the Fourier series of the periodic extension of the time-truncated signal $\{s(t),\ -T/2 < t < T/2\}$ and then taking a suitable limit as $T \to \infty$. Here we will follow a different approach based on the DTFT of the sampled signal

$$s_\Delta[n] \overset{\text{def}}{=} s(n\Delta)\ , \qquad n \in \mathbf{Z}$$

(where $\Delta > 0$ is the sampling period). The idea behind the proof is as follows: if $s_\Delta[\cdot]$ has a frequency-domain representation which converges, as $\Delta \to 0$, to a single "spectrum-like" function, then that function must be valid as a frequency-domain representation for the continuous-time signal $\{s(t),\ t \in \mathbf{R}\}$ also, since that signal can be asymptotically obtained in this fashion, i.e., by reducing the sampling period to zero.

To simplify the analysis, we will let $\Delta \to 0$ by taking

$$\Delta = T, T/2, T/4, \ldots, T/2^r, \ldots$$

in turn, where $T > 0$ is fixed. This ensures that the sampling instants for different values of Δ are *nested*; in particular, any time $t = (m2^{-r})T$, where $r \in \mathbf{N}$ and $m \in \mathbf{Z}$, is a sampling instant for *all* sufficiently small Δ. By varying m and r, we obtain a dense set of points t on the time axis, which are sufficient for the specification of the entire signal $s(\cdot)$.

If t is such a point, then t/Δ is an integer for sufficiently small Δ, and thus t is the $(t/\Delta)^{\text{th}}$ sampling instant for the sequence $s_\Delta[\cdot]$. The synthesis equation (inverse DTFT) gives

$$\begin{aligned} s(t) &= s_\Delta[t/\Delta] \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} S_\Delta(e^{j\omega})e^{j\omega(t/\Delta)}d\omega \\ &= \frac{1}{2\pi}\int_{-\pi/\Delta}^{\pi/\Delta} \{\Delta S_\Delta(e^{j\Omega\Delta})\}\cdot e^{j\Omega t}d\Omega \end{aligned}$$

where the change of variables $\Omega = \omega/\Delta$ was used for the last integral. Note that Ω is now measured in radians per second, which is appropriate for a continuous-time signal.

Viewed as a function of Ω, the expression $\Delta S_\Delta(e^{j\Omega\Delta})$ is periodic with period $2\pi/\Delta$. As we will soon see, this expression converges (as $\Delta \to 0$) for each $\Omega \in \mathbf{R}$, leading to a new function

$$S(\Omega) \overset{\text{def}}{=} \lim_{\Delta\to 0} \Delta S_\Delta(e^{j\Omega\Delta}) , \qquad \Omega \in \mathbf{R}$$

The function $S(\Omega)$ cannot be periodic. Assuming that it decays sufficiently fast as $\Omega \to \infty$ and $\Omega \to -\infty$ (which it does, by virtue of the integrability conditions on $s(\cdot)$), the last integral will converge, as $\Delta \to 0$, to

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} S(\Omega)e^{j\Omega t}d\Omega$$

Thus also

$$s(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\Omega)e^{j\Omega t}d\Omega$$

Remark. It is interesting to note that the equivalent expression

$$s(t) = \frac{1}{2\pi}\int_{0}^{2\pi/\Delta} \Delta S_\Delta(e^{j\Omega\Delta})e^{j\Omega t}d\Omega$$

will, upon replacing $\Delta S_\Delta(e^{j\Omega\Delta})$ by $S(\Omega)$ and $2\pi/\Delta$ by its limiting value ∞, result in the *incorrect* integral

$$\frac{1}{2\pi}\int_{0}^{\infty} S(\Omega)e^{j\Omega t}d\Omega$$

for $s(t)$. This paradox can be resolved by observing that, no matter how small Δ is, the integral for $s(t)$ is taken over an entire period of the function $\Delta S_\Delta(e^{j\Omega\Delta})$. As $\Delta \to 0$, the period T/Δ approaches infinity, and the limiting function $S(\Omega)$ is aperiodic—in other words, a period of $S(\Omega)$ takes up the entire real axis. Thus the correct limits for the integral should be $-\infty$ to ∞.

To show that $\Delta S_\Delta(e^{j\Omega\Delta})$ indeed converges, we use the analysis equation:

$$\begin{aligned}\Delta S_\Delta(e^{j\Omega\Delta}) &= \Delta \cdot \sum_{n=-\infty}^{\infty} s_\Delta[n] e^{-jn\Omega\Delta} \\ &= \sum_{n=-\infty}^{\infty} s(n\Delta) \cdot e^{-jn\Omega\Delta} \cdot \Delta\end{aligned}$$

The last sum consists of equally spaced samples (over the entire time axis) of the function $s(t)e^{-j\Omega t}$, each multiplied by the spacing, or sampling period, Δ. As $\Delta \to 0$, the sum converges to an integral, and thus

$$S(\Omega) = \lim_{\Delta \to 0} \Delta S_\Delta(e^{j\Omega\Delta}) = \int_{-\infty}^{\infty} s(t) e^{-j\Omega t} dt$$

The function $S(\Omega)$ is the *Fourier transform,* or spectrum, of the continuous-time signal $s(t)$. Here, Ω (in rad/sec) is a continuous frequency parameter ranging from $-\infty$ to ∞. Similarly, $s(t)$ is the *inverse Fourier transform* of $S(\Omega)$. Note that in this case, the two domains, time and frequency, have the same dimension (that of the continuum). This gives rise to interesting duality properties, which are apparent from the similarities between the analysis and synthesis equations:

$$\begin{aligned}S(\Omega) &= \int_{-\infty}^{\infty} s(t) e^{-j\Omega t} dt \\ s(t) &= \frac{1}{2\pi} \int_{0}^{\infty} S(\Omega) e^{j\Omega t} d\Omega\end{aligned}$$

Nyquist Sampling

In Section 1.6, we saw that an analog signal made up of a continuum of sinusoidal components with frequencies in the range $[0, f_B)$ (Hz) *cannot* be sampled at a rate less than $2f_B$ without loss of information due to aliasing. We will show how sampling at a rate greater than, or equal to, $2f_B$

(samples/second) allows for perfect reconstruction of the continuous-time signal.

Let us assume then that the continuous-time signal $\{s(t),\ t \in \mathbf{R}\}$ has a Fourier transform $S(\Omega)$ which vanishes outside the interval $(-\Omega_B,\, \Omega_B)$ (where $\Omega_B = 2\pi f_B$). We thus have

$$s(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\Omega)e^{j\Omega t}d\Omega = \frac{1}{2\pi}\int_{-\Omega_B}^{\Omega_B} S(\Omega)e^{j\Omega t}d\Omega \tag{E.1}$$

Sampling at the Nyquist rate yields a sequence of samples at times $t = nT_s$, where

$$T_s \overset{\text{def}}{=} \frac{1}{2f_B} = \frac{\pi}{\Omega_B}$$

From (E.1), we obtain an expression for each of these samples:

$$s(nT_s) = \frac{1}{2\pi}\int_{-\Omega_B}^{\Omega_B} S(\Omega)e^{j\pi(\Omega/\Omega_B)}d\Omega \tag{E.2}$$

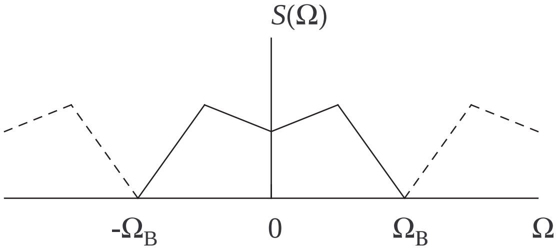


Figure E.2: Shown in bold is the Fourier transform (spectrum) $S(\Omega)$ of a bandlimited continuous-time signal.

The signal $S(\Omega)$ has finite duration in the frequency domain. As such, it can be periodically extended outside the interval $(-\Omega_B,\, \Omega_B)$ (see Figure E.2) and the resulting periodic extension can be expressed in terms of a Fourier series in Ω. Clearly, the Fourier series will return $S(\Omega)$ for Ω in the interval $(-\Omega_B,\, \Omega_B)$. Using the Fourier series formulas developed earlier (with the period parameter equal to $2\Omega_B$), we have

$$S(\Omega) = \sum_{n=-\infty}^{\infty} A_n e^{jn\pi(\Omega/\Omega_B)} \tag{E.3}$$

where

$$A_n = \frac{1}{2\Omega_B}\int_{-\Omega_B}^{\Omega_B} S(\Omega)e^{-jn\pi(\Omega/\Omega_B)}d\Omega \tag{E.4}$$

Comparing (E.2) and (E.3), we immediately obtain

$$A_{-n} = \frac{\pi}{\Omega_B}s(nT_s) = T_s s(nT_s) \tag{E.5}$$

Thus the sequence of samples of $s(\cdot)$ obtained at the Nyquist rate $2f_B$ completely determine the spectrum $S(\Omega)$ through its Fourier series expansion (E.3). Since $S(\Omega)$ uniquely determines $s(t)$ (via the analysis equation (E.1)), it follows that sampling at the Nyquist rate is sufficient for the faithful reconstruction of the bandlimited signal $s(t)$.

As it turns out, we can obtain an explicit formula for $s(t)$ at any time t in terms of the samples $\{s(nT_s)\}$ by combining (E.1), (E.3) and (E.5) as follows:

$$\begin{aligned} s(t) &= \frac{1}{2\pi}\int_{-\Omega_B}^{\Omega_B}\left(\sum_{n=-\infty}^{\infty} A_{-n}e^{-jn\pi(\Omega/\Omega_B)}\right)\cdot e^{j\Omega t}d\Omega \\ &= \frac{1}{2\Omega_B}\cdot\sum_{n=-\infty}^{\infty} s(nT_S)\cdot\int_{-\Omega_B}^{\Omega_B} e^{j\Omega(t-nT_s)}d\Omega \end{aligned}$$

The inner integral equals

$$\frac{e^{j\Omega_B(t-nT_s)} - e^{-j\Omega_B(t-nT_s)}}{j(t-nT_s)} = \frac{2\sin(\Omega_B(t-nT_s))}{t-nT_s}$$

and thus the final expression for $s(t)$ is

$$s(t) = \sum_{n=-\infty}^{\infty} s(nT_s)\cdot\frac{\sin(\Omega_B(t-nT_s))}{\Omega_B(t-nT_s)}$$

Clearly, $s(t)$ as a weighted sum of all samples $s(nT_s)$ with coefficients provided by the *interpolation* function

$$I_{\Omega_B}(\tau) \overset{\text{def}}{=} \frac{\sin(\Omega_B\tau)}{\Omega_B\tau}$$

evaluated for $\tau = t - nT_s$. Writing

$$s(t) = \sum_{n=-\infty}^{\infty} s(nT_s)\cdot I_{\Omega_B}(t-nT_s)$$

we can also see that $s(\cdot)$ is a linear combination of the signals $I_{\Omega_B}(\cdot - nT_s)$, which are time-delayed versions of $I_{\Omega_B}(\cdot)$ centered at the sampling instants nT_s.