

# 3F1: Discrete-Time Signal Processing and Control

Jossy Sayir  
js851@cam.ac.uk

Michaelmas Term 2024



# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Pre-requisites . . . . .	3
1.2	Motivation . . . . .	4
1.3	Revision of continuous time signal processing . . . . .	5
<b>2</b>	<b>Discrete-time signals, linear time-invariant systems (LTIS) and the discrete convolution</b>	<b>10</b>
2.1	Discrete-time signals . . . . .	10
2.2	Difference equations . . . . .	11
2.3	Linear time-invariant systems (LTIS) and the Kronecker delta signal . . . . .	13
2.4	The discrete convolution . . . . .	14
<b>3</b>	<b>The <math>z</math> transform</b>	<b>16</b>
3.1	Motivation . . . . .	16
3.2	The $z$ transform: definition . . . . .	18
3.2.1	Convergence . . . . .	19
3.2.2	A few words about the two-sided $z$ transform . . . . .	20
3.2.3	The $z$ transform and the Discrete-Time Fourier Transform . . . . .	22
3.3	Properties of $z$ transforms and of the DTFT . . . . .	23
3.3.1	Convolution property . . . . .	23
3.3.2	Linearity . . . . .	24
3.3.3	Time shift properties . . . . .	26
3.3.4	Initial Value Theorem . . . . .	27
3.3.5	Conjugate symmetry properties . . . . .	28
3.3.6	Scaling with a geometric sequence . . . . .	29
3.4	Inverting the $z$ transform . . . . .	30
3.5	Relation to other transforms . . . . .	35
<b>4</b>	<b>Linear systems properties and stability</b>	<b>38</b>
4.1	The transfer function of a linear system . . . . .	38
4.2	Poles of the transfer function and BIBO stability . . . . .	42
4.3	Stationary response to sinusoidal inputs and Bode diagrams . . . . .	48
4.3.1	Stationary output of stable systems for oscillatory inputs . . . . .	48

4.3.2	Bode diagram . . . . .	49
4.4	Feedback, closed loop control systems and Nyquist stability criterion . . .	52



# Chapter 1

## Introduction

### 1.1 Pre-requisites

These notes document 16 lectures in 3F1 covering the discrete-time signal processing, control theory part of the course, and (as of 2024) the continuous random processes part of the course as well (the final 3 lectures).

This IIA course assumes that you have the following background knowledge:

Topic	Source
Linear ordinary differential equations (ODEs) with constant coefficients	A-level Further Maths, revised in 1P4 MT
Linear difference equations	1P4 MT
Convolution and impulse responses	1P4 LT
Laplace Transform	1P4 ET
Bode diagrams and Nyquist stability criterion	2P6 MT
Sampling theorem *	2P6 LT
Quantisation *	2P6 LT
Fourier series *	1P4 LT
Fourier transform *	2P6 LT
Discrete Fourier transform (DFT) *	2P6 LT but will be re-taught from scratch in 3F1
Probability	2P7 LT

The items marked with a \* are assumed but not built on extensively in 3F1.

For those transferring from another tripos who have not taken the Part I modules required for this course, we will begin this course by a bird's eye recap of all the material assumed, but I also recommend that you obtain the lecture notes of the relevant modules and refer to them when material taught in 3F1 builds on prior knowledge.

## 1.2 Motivation

With the exception of the sampling theorem and the DFT, everything in the previous table covers continuous-time signal processing and systems. A “signal” in this context is simply a function of a real-valued variable  $t$ , usually time<sup>1</sup>. The aim of this course is for you to learn how the concepts you’ve learned in the context of continuous-time signals transfer to a discrete-time context. The “time” variable is now no longer a continuous quantity but a discrete index that takes values  $0, 1, 2, 3, \dots$ . Signals are no longer “functions” plotted against a continuous axis but sequences of numbers (note that a sequence is still a function in the mathematical sense of the word: a function of the index, which is a natural number.)

Discrete-time signal processing is essential in the modern world because the vast majority of applications are inherently digital, with processing happening after a continuous-time signal has been sampled by an analog-to-digital (ADC) converter and the outcome converted back to an analog signal by a digital-to-analog (DAC) converter. There are very few applications for pure continuous-time signal processing in the modern world. A full understanding of digital signal processing would require us to combine sampling (discretisation of functions in the time domain) and quantisation (discretisation of signal values in the amplitude domain.) In fact, this is not a theoretically well explored area: we know how to quantify the effect of quantisation as added noise (as covered in 2P6) and there is a branch of information theory (rate distortion theory) that has a little more to say about quantisation, but that is the limit of our knowledge about full digital signal processing. In this course, we will ignore the effects of quantisation and assume that amplitudes are continuous. That’s why the title of these notes refer to *discrete-time* signal processing rather than *digital* signal processing.

My aim in this course is for students to realise that discrete-time signal processing is a lot more fun and a lot easier than its continuous-time equivalent. As an undergraduate student, I fell in love with information theory (taught in 3F7), which is still my main area of specialisation today, but digital signal processing was a close second. What fascinated me is that signals and systems become far more tangible for me when they are objects one can calculate by hand, write a programme to simulate, or type a few commands in Python or MATLAB to experiment with. I experimented broadly with sampled music and speech signals and still remember this as one of the highlights of my undergraduate course. When I did my degree, digital signal processing was a relatively new topic taught by professors who had spent most of their life working on analog signal processing and only learned or developed digital processing late in their careers, so I did not find it surprising that it was taught as an advanced 3rd year module after two years of continuous time signal processing. If I were to design an engineering course from scratch today, I would probably teach discrete-time signal processing first and then progress to the mathematically far more intricate world of continuous time signal processing. On the other hand, the benefit of the current practice for you is that you now have a nice and easy module in the third year,

---

<sup>1</sup>There are applications of signal processing where the variable is not time, such as in image processing when you plot the brightness in function of position.

that will consolidate and simplify concepts you've already learned the hard way over the past 2 years, rather than be bombarded with difficult new concepts (take 3F7 if you're looking for new concepts...)

### 1.3 Revision of continuous time signal processing

We now give a brief overview of what you have learned about continuous time signals and systems, mainly to help you make connections you may not have made when you learned the material in disconnected modules.

A signal is a function of a real variable, typically time and denoted  $t$ .

#### Differential Equations

An  $n$ -th order linear differential equation with constant coefficients is an equation of the form

$$y + a_1y' + a_2y'' + a_3y^{(3)} + \dots + a_ny^{(n)} = u + b_1u' + \dots + b_ku^{(k)} \quad (1.1)$$

where  $y$  is a signal to be determined (the “output”),  $u$  is a known signal (the “input”) and  $a_1, \dots, a_n$  are constant coefficients. The notation

$$y^{(k)} = \frac{d^k y}{dt^k} \quad (1.2)$$

denotes the  $k$ -th derivative of  $y$ . The appellation “differential equation” is a misnomer because the equality sign in the equation needs to be fulfilled for every  $t$  and hence such an “equation” is really an infinite set of equations which, technically, should be called an *identity*.

The full set of solutions of an  $n$ -th order linear differential equation with constant coefficients consists of two components:

1. the “complementary function”  $y_{CF}$  which is the solution of the homogeneous equation, where the input  $u$  is replaced by 0. This complementary function can have up to  $n$  terms and has a number of undetermined parameters or factors. Another term for “complementary function” that we will prefer to use in this course is “transient response”.
2. the “particular integral”  $y_{PI}$  is any specific solution of the differential equation. Other terms for particular integral that we will prefer to use in this course, are “stationary response” or “steady state response”.

For any  $y_{CF}$  and  $y_{PI}$ ,  $y_{CF} + y_{PI}$  is a solution of the differential equation since, plugging it into the left of 1.1 will yield  $0 + u = u$  on the right.

As mentioned, a differential equation has a set of solutions rather than a specific solution. Boundary / initial conditions, when present, determine the specific solution. A

linear differential equation with constant coefficients describes a *linear time-invariant system* with input  $u$  and output  $y$ . The complementary function does not depend on the system's input. For a well-behaved system, it is desirable for the complementary function to decay so any initial effects are forgotten and the system converges to a mode of operation where its output depends only on its input signal. Such a system is said to be *stable*.

## LTIS, impulse response and convolution

A linear time-invariant system (LTIS) satisfies the superposition property. If we denote as  $y = \mathcal{L}(u)$  the stationary solution for the equation with input  $u$ , then

$$\mathcal{L}(c_1u_1 + c_2u_2) = c_1\mathcal{L}(u_1) + c_2\mathcal{L}(u_2) \text{ (linearity)} \quad (1.3)$$

and

$$\mathcal{L}[u(t - \tau)] = y(t - \tau) \text{ (time-invariance)}. \quad (1.4)$$

Since any signal  $u$  can be written as a superposition of impulse functions

$$u(t) = \int_{-\infty}^{\infty} u(\tau)\delta(t - \tau)d\tau, \quad (1.5)$$

the response of a system to any input  $u$  can be computed as a convolution of  $u$  with the system's impulse response  $h$ ,

$$y(t) = \int_{-\infty}^{\infty} u(\tau)h(t - \tau)d\tau. \quad (1.6)$$

The convolution operation is also denoted  $y = u * h$ .

## Transforms

The convolution operation is an integral which is not always easy to work out and not suitable when working out solutions of coupled differential equations. Transforms that satisfy the “convolution property” can be applied to signals, i.e., a convolution of signals is equivalent in the transform domain to a multiplication of their transforms. Transforms in the continuous domain are a bit of a “magic trick” that is hard to motivate on intuitive grounds. As we will soon see, one of the benefits of discrete-time signal processing is that transforms are a lot easier to motivate and understand.

There are three principal transforms for continuous signals that you've learned about so far:

- Fourier series apply to periodic functions or functions defined over a finite interval. Their transform is a set of coefficients and, though I don't believe this has been taught in 1P4, it also satisfies the convolution property in that the convolution of two periodic functions has as its Fourier series the product of the coefficients of the two functions;

- Fourier transforms are the tool preferred by signal processing engineers and in circuit analysis. They are the reason why you've been taught that the complex impedance of a capacitor is  $1/(j\omega C)$  rather than  $1/(sC)$ . They satisfy the convolution property and can be used to investigate and design the stationary response of a system but *not* its transient response;
- Laplace transforms are the tool preferred by control engineers. They satisfy the convolution property and can be used to investigate the transient response (and hence the stability of a system) as well as the stationary response (it is identical to the Fourier transform on the imaginary axis for signals that are zero at negative times.)

The Fourier transform, for a signal  $f(t)$ , is

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt, \quad (1.7)$$

where  $\omega$  is a real-valued variable. It is beautifully symmetric in that its inverse transform<sup>2</sup> can be computed as

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{j\omega t} d\omega. \quad (1.8)$$

The Laplace transform is

$$F(s) = \int_0^{\infty} f(t)e^{-st} dt, \quad (1.9)$$

where  $s$  is a complex variable. The only method to invert a Laplace transform taught in Part I was by inspection. An inverse formula for the Laplace transform exists, but understanding it requires a background in complex calculus which is not taught in the Cambridge Engineering tripis.

An LTIS can hence be described in 3 fully equivalent manners shown in Figure 1.1, and you've learned to convert between them (except perhaps from impulse response to

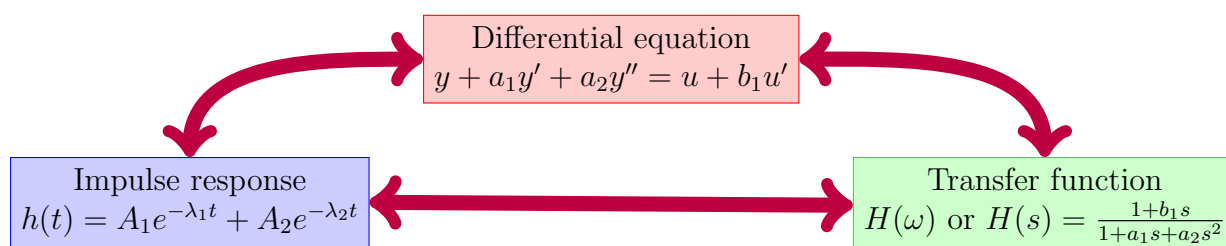


Figure 1.1: The 3 equivalent descriptions of a continuous-time LTIS

<sup>2</sup>Some texts define both the Fourier transform and its inverse with a factor of  $1/\sqrt{2\pi}$ , which is in many ways better: in signal space terms, the transform is then an inner product preserving transform and Parseval's theorem holds with no factors, i.e., the inner product  $\int f(\tau)g^*(t-\tau)d\tau$  of two signals equals the inner product of their transforms, and the energy  $\int |f(t)|^2 dt$  of a signal is preserved by the transform.

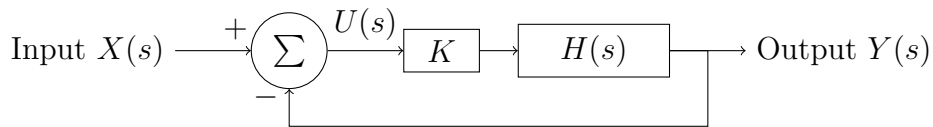


Figure 1.2: The closed-loop system corresponding to the open-loop system  $H(s)$

differential equation, but that's possible if needed.) Of the 3, the most useful description is the transfer function because it can be multiplied with input signals to obtain corresponding stationary output signals and hence combined into overall transfer functions in a network of interconnected systems.

## System spectral response and stability

A linear system always responds to a sinusoidal input with an output sinusoidal of the same frequency, with an adjusted magnitude and phase. The Bode diagram or magnitude and phase of the Fourier transform  $H(\omega)$  gives the system gain and phase shift for every input frequency  $\omega$ . This is a good time to discuss the subtle difference between Fourier and Laplace transforms: the Fourier transform only gives the behaviour of the stationary output, so the assumption when considering an input sinusoidal is that this sinusoidal has been going on from time  $-\infty$  to  $\infty$ , and the input is just a sinusoidal with the adjusted magnitude and phase. The Laplace transform on the other hand assumes an input sinusoidal that starts at time  $t = 0$  and hence the corresponding output consists in a stationary solution, which is identical to that obtained via the Fourier transform (except that it starts at time  $t = 0$ ) and a transient response that might or might not decay to 0, depending on the system.

A system whose transient response decays to zero is called “stable”. There are two scenarios of interest with regards to stability:

- the stability of a system can be derived directly from its transfer function, specifically from the position of its poles;
- the stability of a closed-loop (feedback) system can be derived from the Nyquist diagram of the open-loop (non-feedback) system.

Note that in the latter scenario we do not care about the stability of the open-loop system. Indeed, feedback is often introduced precisely to stabilise an open-loop system that would otherwise be unstable.

A system is stable if the poles of its transfer function lie on the left hand side of the complex plane. If it has poles on the imaginary axis, it is not stable, as certain input signals (sinusoidals of the frequency corresponding to the pole positions) will cause the transient response to grow unbounded. If it has poles on the right hand side of the plane, most input signals will cause an unbounded response.

The Nyquist diagram (locus of the Laplace transfer function  $H(j\omega)$  on the complex plane) allows one to study the stability of the corresponding closed-loop system shown in Figure 1.2 with transfer function

$$H_{CL}(s) = \frac{KH(s)}{1 + KH(s)} \quad (1.10)$$

which is obtained by writing equations tying  $X(s)$ ,  $U(s)$  and  $Y(s)$  and resolving for  $H_{CL}(s) = Y(s)/X(s)$ . The Nyquist stability criterion determines that the closed loop system is stable if and only if the number of encirclements of the point  $s = -1$  in the Nyquist diagram equals the number of poles of the open-loop system on the right-hand side of the plane.

## Chapter 2

# Discrete-time signals, linear time-invariant systems (LTIS) and the discrete convolution

In Part I, we learned about sampling. If samples of a continuous signal are taken at regular time intervals  $T$ , the sampling theorem states that in theory, the continuous signal can be exactly reconstructed from its samples, provided that its Fourier spectrum is band-limited within frequency band  $[-1/(2T), 1/(2T)]$ . You may have been wondering why one might ever want to sample a continuous signal then reconstruct it exactly from its samples? Wouldn't that be a rather futile, useless exercise? Indeed! In fact, the true meaning of the sampling theorem is that one has not lost any information about the band-limited continuous signal by sampling it, as the samples contain all the information needed to mathematically reconstruct the original signal. In the real world, we sample signals so we can process their digital versions to obtain new signals that may then be converted back to continuous signals. The advantage of doing so is that digital signal processing is far more flexible and powerful than its analog equivalent, and there are also information-theoretical benefits in that signals can be compressed and transmitted with arbitrary reliability once they are in digital format. In 3F1, we are about to learn how to process discrete-time signals to shape their spectrum, stabilise systems, and do many other signal wizarding tricks. Compression and reliable transmission is covered in 3F7.

### 2.1 Discrete-time signals

A discrete-time signal is a function  $s(k)$  of an integer argument  $k$ . It is more commonly denoted using the index notation  $s_k$ .

It will be helpful to distinguish between the following type of signals:

- finite duration signals  $\{s_m, s_{m+1}, \dots, s_{m+\ell-1}\}$  for which  $s_m \neq 0$ ,  $s_{m+\ell-1} \neq 0$ , and  $s_k = 0$  for  $k < m$  and for  $k \geq m + \ell$ .  $\ell$  is the duration of the signal.  $m$  can

be positive, zero, or negative, e.g., a signal of duration  $\ell = 6$  could start at time  $m = -3$  and end at time  $m + \ell - 1 = 2$ , for example

$$(s_{-3}, s_{-2}, s_{-1}, s_0, s_1, s_2) = (5, 3, 0, 0, -1, 3).$$

As is clear from the example, it is not necessary for all signal values in the range from  $m$  to  $m + \ell - 1$  to be non-zero;

- right-sided signals  $\{s_m, s_{m+1}, s_{m+2}, \dots\}$  are zero for all  $k < m$ ,  $s_m \neq 0$ , but there is no upper bound for indices of non-zero values, i.e., for any  $n$ , there exist  $k > n$  for which  $s_k \neq 0$ . Again,  $m$  can be negative, zero, or positive;
- left-sided signals  $\{\dots, s_{m-2}, s_{m-1}, s_m\}$ , by analogy, are zero for all  $k > m$ ,  $s_m \neq 0$ , and there is no lower bound for indices of non-zero values.
- two-sided signals  $\{s_k\}_{-\infty < k < \infty} = \{\dots s_{-2}, s_{-1}, s_0, s_1, s_2 \dots\}$  for which there is no upper *or* lower bound to the indices of non-zero values.

In 3F1, we will concentrate mostly on finite length and right-sided signals that start at  $m = 0$ , but there will be instances where we might consider signals that start at  $m < 0$  or two-sided signals. Right-sided signals that start at  $m = 0$  are also called “semi-infinite sequences”. Left-sided signals are generally excluded for a reason that will become clear later in the course.

The signal values  $s_k$  for any specific  $k$  will mostly be real, but we will sometimes also consider complex-valued signals as these have many applications, for example in communications when considering digital passband modulation.

In some textbooks and in some old exam questions, you may see the notation  $s(kT)$  for a discrete-time signal that makes explicit reference to its corresponding continuous-time signal  $s(t)$  and specifies the sample interval  $T$  and hence sample frequency  $f_s = 1/T$ . This is a reassuring notation when you start off with a background in continuous-time signal processing, but it is not overly helpful: in discrete-time signal processing, we prefer to abstract from the sampling process and consider signals simply as sequences of numbers. You can think of the signal as having a normalised sampling frequency of  $f_s = 1$  Hz or  $\omega_s = 2\pi$  rad s<sup>-1</sup>. Assuming that the signal has been sampled with no aliasing, i.e., its spectrum is fully contained in  $[-\omega_s/2, \omega_s/2]$ , this corresponds to considering only frequencies in the range  $[-\pi, \pi]$  (in rad s<sup>-1</sup>) where the negative frequencies mirror the positive frequencies if the signal is real-valued (but not if it’s complex-valued.) We will come back to this when we discuss the spectrum of discrete-time signals more in detail later in the course.

## 2.2 Difference equations

Difference equations, also known as recurrence relations, were covered in IA Paper 4, MT. A linear difference equation with constant coefficients mirrors the definition of corresponding differential equations, i.e.,

$$y_k + a_1 y_{k-1} + a_2 y_{k-2} + \dots + a_n y_{k-n} = u_k + b_1 u_{k-1} + \dots + b_m u_{k-m} \text{ for } k \geq n \quad (2.1)$$

with possibly some initial conditions to specify  $y_0, y_1, \dots, y_{n-1}$ . Note how the equation is always accompanied by an inequality to specify from which index onwards it applies, i.e., in the equation above the signal values  $y_k$  for  $k < n$  are not required to satisfy the conditions of the difference equation. In the past, IA Paper 4 only covered homogeneous difference equations, for which  $u_k = 0$  for all times  $k$ , but at least as of Michaelmas Term 2022, there are examples of non-homogeneous equations and their solutions in the 1P4 lecture notes. In any case, the solution of linear difference equations follows the solution of linear differential equations so closely that you should be able to figure out the solution of non-homogeneous equations even if you have not learned them formally. Similar to differential equations, the general solution consists of

- a complementary function  $y_k^{CF}$  which is the set of solutions of the homogeneous difference equation (where you set  $u_k = 0$  for all  $k$ ). We will refer to complementary functions as the “transient response”;
- a particular solution  $y_k^{PS}$  which is a solution of the difference equation. We will refer to particular solutions as the “stationary” or “steady-state response”.

For any  $y^{CF}$  and  $y^{PS}$ ,  $y^{CF} + y^{PS}$  is a solution of the difference equation since, plugging it into the left of 2.1 will yield  $0 + u = u$  on the right.

As for continuous systems, the transient response only depends on the system and not on its input, whereas the stationary response depends on the system and its input. Systems whose transient decays over time are stable.

By convention, solutions of homogeneous differential equations are written as linear combinations of exponential functions, i.e.,  $A_1 e^{\lambda_1 t} + \dots + A_n e^{\lambda_n t}$ , whereas solutions of homogeneous difference equations are written as linear combinations of geometric sequences, i.e.,  $A_1 q_1^k + \dots + A_n q_n^k$ . This is merely a convention<sup>1</sup> rather than a fundamental distinction, because we could re-write a geometric sequence as an exponential, i.e.,  $Aq^k = Ae^{k \log q} = Ae^{\lambda k}$  where  $\lambda = \log q$ .

---

<sup>1</sup>Having taught 3F1 for some years now, I have received an incredible number of frustrated complaints from past students as to why everything in discrete-time is different from continuous time, despite it appearing so similar at first glance. As you will discover, we will introduce discrete-time versions of all the tools you are familiar with, transforms, frequency (Bode) plots, Nyquist diagrams, etc., but all of them are subtly different from their discrete time counterparts. I deeply sympathise and commiserate with all my past and current students and share their frustration. The culprit for these differences is simply the convention stated here: in continuous time, people like to use exponentials, whereas in discrete time they like to use geometric sequences. That’s it!! There is *no* essential or fundamental difference. It’s all just due to a silly convention. I wish I could go back and change this (it’d be easy enough) but I’d be deviating from decades of signal processing practice and you’d hate me once you went out to work and had to interact with other engineers who have been taught signal processing in a conventional way.

## 2.3 Linear time-invariant systems (LTIS) and the Kronecker delta signal

A linear difference equation with constant coefficients describes a linear time-invariant system (LTIS). If we denote as  $y^{(i)} = \mathcal{L}(u^{(i)})$  the solution of the difference equation for input signal  $u^{(i)}$ , then

$$\mathcal{L}(c_1 u^{(1)} + c_2 u^{(2)}) = c_1 \mathcal{L}(u^{(1)}) + c_2 \mathcal{L}(u^{(2)}) \text{ (linearity)} \quad (2.2)$$

and

$$\mathcal{L}[u_{k-k_0}] = y_{k-k_0} \text{ (time-invariance)}. \quad (2.3)$$

As in the continuous case, we can write any signal as a superposition of “impulses”. The discrete equivalent of the (Dirac) impulse function goes by the grand name of “Kronecker delta<sup>2</sup>” signal and is much simpler to define and understand than its continuous cousin:

$$\delta_k = \begin{cases} 0 & \text{for } k \neq 0 \\ 1 & \text{for } k = 0. \end{cases} \quad (2.4)$$

Unlike the continuous case, there is no difficulty in having to define a signal that’s infinitely thin and infinitely tall and integrates to 1 (to the great distress of formalists who argue that it isn’t a function at all...). The Kronecker delta signal is a very plain signal that naturally sums to 1.

As mentioned, a signal  $u$  can be written as a superposition of Kronecker delta signals

$$u_k = \sum_{n=-\infty}^{\infty} \delta_n u_{k-n} \quad (2.5)$$

and hence (2.2) and (2.3) imply that the output signal of an LTIS is

$$y_k = \sum_{n=-\infty}^{\infty} h_n u_{k-n} \quad (2.6)$$

where  $\{h_k\}$  is the response of the LTIS to a Kronecker delta input.

$\{h_k\}$  is called the “Kronecker delta response” (or delta response) of the system and (2.6) is the discrete equivalent of the *convolution* integral we are familiar with in continuous time signal processing. More about convolution in the next section.

---

<sup>2</sup>Kronecker delta can also designate a two-dimensional function  $\delta_{ij}$  of integers  $i$  and  $j$  which is zero everywhere except when  $i = j$  but in signal processing we tend to use the one-dimensional version, which is equivalent in the sense that  $\delta_{ij} = \delta_{i-j}$ .

## 2.4 The discrete convolution

The general expression for the discrete convolution between signals  $s$  and  $u$ , denoted  $s * u$ , is

$$y_k = \sum_{n=-\infty}^{\infty} s_n u_{k-n} \text{ for any } k. \quad (2.7)$$

We can apply a variable substitution  $k' = k - n$  in the sum, then re-name the new variable  $k'$  back to  $k$  to show that discrete convolution is commutative, i.e.,  $u * s = s * u$  or

$$y_k = \sum_{n=-\infty}^{\infty} s_n u_{k-n} = \sum_{n=-\infty}^{\infty} s_{k-n} u_n \quad (2.8)$$

Note that the sum goes from  $-\infty$  to  $\infty$ . If, say, one of the signals is the delta response of an LTIS and that impulse response is right-sided starting from  $m = 0$ , i.e., its values for negative indices is zero, then we can re-write the expression for the discrete convolution  $h * u$  as

$$y_k = \sum_{n=0}^{\infty} h_n u_{k-n}. \quad (2.9)$$

An LTIS with a right-sided delta response that starts at  $m = 0$  is called *causal*, because the behaviour of its output signal only depends on present and past values of its input signal but not on future values. Causal systems are an important field of study because some signal processing is implemented in real-time, where samples are handled in sequence as they are received. Such systems must necessarily be causal, as they do not have access to future input values when they produce their output. However, note that causality is far less central in discrete-time signal processing than in continuous-time signal processing, because

- in many applications, we process stored signals that are available in full at the time of processing, and in this case there is absolutely no need to insist on causality (indeed one loses many degrees of freedom by doing so). Think of the signal as a Python list, a NumPy array, or data in an EXCEL file, not necessarily as a voltage evolving at the input of your device as was mostly the case in continuous signal processing;
- in image processing, the index is location rather than time and in most cases inputs to the right and to the left of the current pixel are available in equal measure, so causality is irrelevant;
- even for real-time processing, it is often possible to buffer the input signal and to operate a non-causal LTIS virtually a number of samples delayed with respect to the input signal, so that the input signal is known a few samples in the future with respect to the virtually delayed working position in the signal. Delaying in such a manner is theoretically possible in continuous-time as well, but not realisable in practice as there are no practical devices that can store and replay an interval of a

continuous-time signal within a device: the only continuous signal storage that I am aware of is for long term storage, such as magnetic tape or vinyl records.

Note that the convolution sum is always defined as stated in (2.7) with a sum from  $-\infty$  to  $\infty$ : it is the properties of the convolved signals that make the sum in (2.9) go from 0 to  $\infty$ . There are a few more specialisations of this sum that are of interest:

- when the input signal  $u$  is right-sided starting at  $m = 0$ , as well as the delta response, the convolution becomes a finite sum  $y_k = \sum_{n=0}^k h_n u_{k-n}$
- an LTIS with a finite delta response that is zero at all negative times and at times  $k \geq \ell$  is called a “finite impulse response” (FIR) system. We will use this terminology very frequently in 3F1. For an FIR system applied to a right-sided input that starts at  $m = 0$ , the convolution becomes  $y_k = \sum_{n=0}^{\min\{\ell, k\}} h_n u_{k-n}$ . The difference equation corresponding to an FIR system is

$$y_k = h_0 u_k + h_1 u_{k-1} + \dots + h_\ell u_{k-\ell} \text{ for } k \geq \ell. \quad (2.10)$$

In contrast to an FIR, a causal LTIS with a semi-infinite delta response is called an “infinite impulse response” (IIR) system. The terms FIR and IIR could in principle also be applied to non-causal systems although they are less common in that context.

Recalling the 3 equivalent descriptions of continuous-time systems we saw in Figure 1.1, we have now mapped two of those descriptions to discrete-time signal processing, resulting in Figure 2.1. We are missing the “transfer function” representation that we obtained in

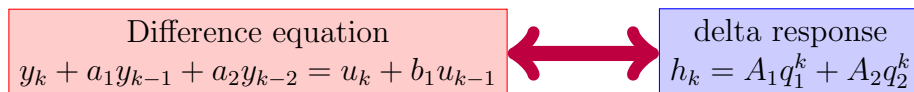


Figure 2.1: Two equivalent descriptions of a discrete-time LTIS

the continuous case by taking transforms. We will introduce this in the next chapter.

# Chapter 3

## The $z$ transform

As we've seen, much of discrete-time signal processing so far mirrors its continuous-time equivalent, with some simplifications. We could now introduce a discrete-time transform by analogy with the Laplace or Fourier transform and that is indeed the approach taken by many textbooks. However, I think doing so would miss a unique opportunity to make sense of transforms. For those who have not had courses in continuous-time signal processing (e.g. those transferring from other triposes), and for those who learned about transforms but saw them as a “magic trick”, never quite understanding why they work (that included myself when I was at your stage in my education), we will begin by giving a motivation that I believe is a natural justification for using transforms. Indeed, I wouldn't be surprised if Laplace and Fourier thought of the discrete case before coming up with their transforms, because transforms are far more natural in the discrete-time world.

### 3.1 Motivation

Let us consider the convolution of two finite-length discrete-time signals  $(u_0, u_1)$  and  $(s_0, s_1, s_2)$ . We will view these signals as infinite length, but equal to zero outside the range  $(0, 1)$  and  $(0, 1, 2)$  for  $u$  and  $s$ , respectively (i.e., asking the value of  $u_{-1}$  is not forbidden or undefined, it's simply  $u_{-1} = 0$ .)

Let us tediously write out the convolution  $y = u * s$  for every  $k$ . If you examine the convolution sum in (2.7), you can picture (as in the continuous case) as overlapping the first signal with the reverted second signal shifted by  $k$ . If  $k < 0$  or  $k > 3$ , the two signals don't overlap in their non-zero ranges. Hence,  $y_k = 0$  for  $k < 0$  and for  $k > 3$ . We only need to compute  $y_k$  for  $0 \leq k \leq 3$ :

$$\begin{cases} y_0 &= u_0 s_0 + u_1 s_{-1} = u_0 s_0 \\ y_1 &= u_0 s_1 + u_1 s_0 \\ y_2 &= u_0 s_2 + u_1 s_1 \\ y_3 &= u_0 s_3 + u_1 s_2 = u_1 s_2 \end{cases} \quad (3.1)$$

Note that the convolved signals had 2 and 3 non-zero values, whereas the convolution has 4 non-zero values. It's easy to see that the full convolution of signals with  $\ell_1$  and  $\ell_2$  non-zero values has  $\ell_1 + \ell_2 - 1$  non-zero values. Examining the outcome of the convolution, observe that the expressions are reminiscent of polynomial multiplication. If we define two polynomials

$$\begin{cases} u(D) &= u_0 + u_1 D \\ s(D) &= s_0 + s_1 D + s_2 D^2 \end{cases} \quad (3.2)$$

and evaluate their multiplication, we obtain

$$y(D) = u(D)s(D) = u_0 s_0 + (u_0 s_1 + u_1 s_0)D + (u_0 s_2 + u_1 s_1)D^2 + u_1 s_2 D^3. \quad (3.3)$$

Observe that the coefficients of  $y(D)$  in (3.3) match the outcome of the convolution in (3.1), i.e., the constant term in  $y(D)$  equals  $y_0$ , the factor of  $D$  equals  $y_1$ , the factor of  $D^2$  equals  $y_2$  and the factor of  $D^3$  equals  $y_3$ . Furthermore, we can interpret  $y(D)$  has having zero factors in front of any other powers of  $D$ , i.e.,  $D^{-1}, D^{-2}, \dots$  and  $D^4, D^5, \dots$ , and that corresponds to our observation that  $y_k = 0$  for  $k < 0$  or  $k > 3$ .

We have used the letter  $D$  for ‘‘Dummy’’ for our polynomial variable at this stage to emphasise the fact that, for the moment at least, we attach no significance to the argument of the polynomial: it's a dummy variable. We are not thinking of the value of  $y(D)$  for a numerical value of  $D$ . Rather, we are mapping finite length sequences to polynomials simply to harness the property of polynomial multiplication to automatically perform convolutions. These are known as ‘‘formal polyomials’’, i.e., polynomials for which the variable is merely a formal symbol and does not stand in for a number.

We can use the same trick for semi-infinite sequences by mapping them to *formal power series*, e.g.,

$$s(D) = s_0 + s_1 D + s_2 D^2 + s_3 D^3 + \dots \quad (3.4)$$

Convolution of semi-infinite sequences is equivalent to multiplication of power series

$$u(D)s(D) = \left( \sum_{k=0}^{\infty} u_k D^k \right) \left( \sum_{m=0}^{\infty} s_m D^m \right) \quad (3.5)$$

$$= \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} u_k s_m D^{k+m} \quad (3.6)$$

$$= \sum_{k=0}^{\infty} \sum_{n=k}^{\infty} u_k s_{n-k} D^n \text{ where we substituted } n = k + m \quad (3.7)$$

$$= \sum_{n=0}^{\infty} \left( \sum_{k=0}^n u_k s_{n-k} \right) D^n \quad (3.8)$$

$$= \sum_{n=0}^{\infty} y_n D^n \text{ where } y_n = \sum_{k=0}^n u_k s_{n-k}, \text{ i.e., } y = u * s. \quad (3.9)$$

Converting sequences to power series is a common trick used across mathematics and engineering. Examples of applications are:

- in combinatorics and other branches of mathematics, formal power series representing a sequence is called a *generating function*;
- in the study of error correction codes (covered in 3F4 and 4F5), the  $D$  transform is used to convert sequences of elements of finite fields (e.g., binary or ternary numbers) into power series;
- in discrete-time signal processing and control, the  $z$  transform converts signals into power series. The  $z$  transform is a bit different in that we do evaluate the polynomial for complex values of  $z$  in some applications, so it is not merely a formal power series.

We call all of these “transforms” because they map a sequence to an object (a polynomial or a power series) and back, where a certain operation is easier to perform in the transform domain. In this case, convolution in the time domain becomes a multiplication in the transform domain.

A final word on formal power series: the relation

$$\frac{1}{1-D} = 1 + D + D^2 + D^3 + \dots \quad (3.10)$$

always holds for a formal power series, whereas it is only true for a power series with a numerical variable if  $|D| < 1$  as the series does not otherwise converge. The reason it always holds for a formal power series is simply due to the calculation rules of polynomial division. If you divide the polynomial  $u(D) = 1$  by  $s(D) = 1 + D$  using long division, you obtain the formal power series on the right of 3.10.

We will define and study the  $z$  transform in the remainder of this chapter. As is clear from this motivation, a formal power series is sufficient to map convolution to the multiplication of power series, and that is our main motivation for introducing a transform. However, the  $z$  transform has other useful properties, some of which rely on the value of the power series for the complex variable  $z$ .

## 3.2 The $z$ transform: definition

**Definition 3.1** *The  $z$  transform of a discrete-time signal  $s$  is defined as*

$$S(z) = \sum_{k=0}^{\infty} s_k z^{-k}. \quad (3.11)$$

The  $z$  transform is an instance of the general transform to polynomials or power series that we discussed in the previous section, with the variable  $D = z^{-1}$ . The name of the variable of course makes no difference whatsoever, and you can easily persuade yourself that

everything we discussed in the previous section works just as well with negative powers<sup>1</sup>. Note that the sum starts from zero, similar to the integral from 0 to  $\infty$  used in the definition of the Laplace transform.

*Example:* Let us compute the  $z$  transform of a geometric sequence

$$s_k = Aq^k \text{ for } k \geq 0. \quad (3.12)$$

Geometric sequences are practically relevant because homogeneous linear difference equations with constant coefficients admit solutions of this form. Their  $z$  transform is

$$S(z) = \sum_{k=0}^{\infty} Aq^k z^{-k} = \frac{A}{1 - qz^{-1}} = \frac{Az}{z - q}. \quad (3.13)$$

Note that in many applications we are restricted to real geometric sequences, i.e.,  $q$  is a real number, but nothing in our derivation above required  $q$  to be real and hence the expression we derived applies just as well to complex geometric sequences where  $q$  is complex. A special case is when  $A = q = 1$  that yields the *unit step* signal

$$u_k = \begin{cases} 0 & \text{for } k < 0 \\ 1 & \text{for } k \geq 0 \end{cases} \quad (3.14)$$

whose  $z$  transform is

$$U(z) = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}. \quad (3.15)$$

### 3.2.1 Convergence

Recall from the previous section that the  $z$  transform isn't simply a formal power series but an actual power series in the complex variable  $z$ . Hence, the next issue we should discuss is the convergence of the power series. For a given signal  $s$ , there may be some complex values of  $z$  for which the expression (3.11) converges to a finite complex value, and some values of  $z$  for which it diverges to infinity. For example, the series in the  $z$  transform of the geometric sequence (3.13) we computed as an example converges only for

---

<sup>1</sup>I don't know why negative powers were chosen for the  $z$  transform. I suspect that it was chosen to mirror the negative power in the Laplace transform. On the other hand, there is no particular reason for the Laplace transform to have a negative sign and the whole theory would work just as well if the Laplace transform were defined with a plus in the exponent (stable systems would need to have their poles in the right hand side of the plane...) The negative powers in the definition of the  $z$  transform can cause headaches when discussing degrees of numerator and denominator polynomials in rational  $z$  transform expressions, as one cannot speak of "negative degrees" and has to factor out terms to have polynomials in positive powers and be able to discuss their degrees. Indeed, that's probably one of the reasons why coding and information theorists prefer to use the  $D$  transform with no negative powers.

$|qz^{-1}| < 1$  or, equivalently,  $|z| > |q|$ . Some textbooks define the  $z$  transform as assigning each discrete-time signal a pair

$$(\text{expression in } z, \text{ region of convergence (ROC) for } z \text{ in } \mathbb{C}) = (S(z), \text{ROC}).$$

This is accurate but tedious. We will refrain from doing so but the reasons why we can avoid this are slightly beyond the scope of the Cambridge Engineering tripos. I'll try to explain this without getting into full technicalities: in complex calculus, one can show that any so-called “analytic function” (that includes all the rational functions that we will be interested in) has a unique continuation in the complex plane as long as it is defined in any region of non-zero surface area. Hence, if you have found an expression  $S(z)$  for the  $z$  transform of a discrete-time signal  $s$  that applies only within a limited region of convergence for  $z$ , this expression  $S(z)$  can unambiguously be extended to the whole complex plane as there is no other way to define a completion over the whole complex plane that maintains the analytic property of the function. Hence, it is fair to refer to  $S(z)$  as the  $z$  transform of  $\{s_k\}$  without restriction on  $z$ , because the expression is an unambiguous continuation of the ROC to the whole of  $\mathbb{C}$ , including regions where the power series doesn't converge.

### 3.2.2 A few words about the two-sided $z$ transform

(This section looks beyond the scope of 3F1. You can skip it if you prefer to.)

While the vast majority of textbooks define the  $z$  transform in the same manner we did in (3.11) as a one-sided sum from 0 to  $\infty$ , there are a few textbooks that define the  $z$  transform as a two-sided sum from  $-\infty$  to  $\infty$ . This deserves a little more thought, because learning a few things about the two-sided  $z$  transform can help us understand the properties of our own one-sided  $z$  transform better.

There are certainly disadvantages to our definition:

- Our definition completely ignores signal values at negative times, so distinct signals with different values at negative times but identical values for times  $k \geq 0$  map to the same  $z$  transform. Our definition is hence not a true “transform” because we'd expect a transform to map signals uniquely to the transform domain and back.
- We could enforce uniqueness by strictly limiting our attention to right-sided signals starting at  $m \geq 0$ , but this isn't always convenient. There are situations for example where you shift signals and end up with signals that have non-zero values at negative times.
- I have found several errors in textbooks and past 3F1 lecture slides, where it was claimed that a system is causal because the  $z$  transform of its delta response does not have coefficients for positive powers of  $z$ : this is wrong, because the  $z$  transform of a non-causal delta response would have only negative powers of  $z$  simply because the definition of the  $z$  transform takes a sum from 0 to  $\infty$  and hence neglects the non-zero values of the response at negative times.

- As we will discover when we discuss time shifting properties of the  $z$  transform, our definition causes those properties to become unnecessarily complicated, whereas they are very simple and natural when using a two-sided definition of the  $z$  transform.

As you can imagine, if there were only drawbacks to our definition and no advantages, we would have taught you the two-sided definition of the  $z$  transform. There are difficulties with the two-sided definition which led us to prefer the definition we gave you in (3.11). This is best illustrated by an example:

*Example:* consider the signals  $u = \{1\}_{k \geq 0}$  and  $v = \{-1\}_{k \leq -1}$ , i.e., the signal that's zero in negative times and 1 in non-negative times vs. the signal that's  $-1$  in negative times and zero in non-negative times. If we apply the two-sided expression  $z$  transform to each, we obtain

$$\tilde{U}(z) = \sum_{k=-\infty}^{\infty} u_k z^{-k} = \sum_{k=0}^{\infty} z^{-k} = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}, \quad (3.16)$$

and

$$\tilde{V}(z) = \sum_{k=-\infty}^{\infty} v_k z^{-k} = - \sum_{k=-\infty}^{-1} z^{-k} = - \sum_{k=1}^{\infty} z^k = z^0 - \sum_{k=0}^{\infty} z^k = 1 - \frac{1}{1 - z} = \frac{z}{z - 1} \quad (3.17)$$

which is the same expression! The sum in (3.16) converges only for  $|z^{-1}| < 1$ , i.e., for  $|z| > 1$ , while the sum in (3.17) converges only for  $|z| < 1$ . Textbooks that map a signal to a pair  $(S(z), \text{ROC})$  avoid this ambiguity but the notation needed to do this is convoluted.

Adopting a one-sided definition of the  $z$  transform allows us to mostly ignore issues related to the region of convergence of power series. However, we won't be militant about this and sometimes when an explanation is rendered easier by using the two-sided  $z$  transform, we may revert to it to simplify a step in a derivation. In particular, note that the two-sided  $z$  transform for finite duration signals is unproblematic as there is no issue of convergence for finite duration signals, even if they start at negative times. Before moving on, it is worth noting two points:

- two-sided  $z$  transforms can be used in the study of stochastic signals when applied to the auto-correlation function of a stationary discrete-time signal, which is always a two-sided function. Textbooks that prefer two-sided  $z$  transforms often deal with stochastic signals<sup>2</sup>;

---

<sup>2</sup>3F3, where discrete-time stochastic signals are taught in Cambridge avoids using  $z$  transforms altogether (mainly so students can choose to take 3F3 without taking 3F1) and in 3F1 we will have 3 lectures on continuous stochastic signals for which we will use the tools we learned in Part I.

- we have already learned about a two-sided transform for discrete-time signals in 2P6: the Discrete-Time Fourier Transform (DTFT). We will revisit this transform in the next section, discuss its relation to the  $z$  transform, and use it extensively in this course.

### 3.2.3 The $z$ transform and the Discrete-Time Fourier Transform

In 2P6, we learned about the Discrete-Time Fourier Transform (DTFT) of a signal, defined as (from 2P6 Signal & Data Analysis, Handout 5, Slide 16, with slightly modified notation to use the normalised frequencies we prefer in 3F1),

$$S(\theta) = \sum_{k=-\infty}^{\infty} s_k e^{-jk\theta} \quad (3.18)$$

where  $\theta = \omega T = \omega/f_s$  is the normalised angular frequency, where  $\omega$  is the frequency in  $\text{rad s}^{-1}$ ,  $T$  is the sampling interval in s and  $f_s$  is the sampling frequency in Hz. Notice that (3.18) is the expression of the two-sided  $z$  transform evaluated on the unit circle. Furthermore, if the signal  $s$  has non-zero values only for non-negative time indices, then the DTFT is also our own 1-sided  $z$  transform evaluated on the unit circle, i.e., for  $\{s_k\}_{k \geq 0}$ ,

$$S(\theta) = S(e^{j\theta}) = S(z) \text{ for } z = e^{j\theta}. \quad (3.19)$$

As we learned in 2P6, the DTFT represents the spectrum of a discrete-time signal, or in other words the (periodic) Fourier spectrum of its associated train of impulses. We now understand that the  $z$  transform evaluated on the unit circle is the *spectrum* of a discrete-time signal. The spectrum represented as a function of the normalised angular frequency  $\theta$  is  $2\pi$  periodic, which is in line with what we learned regarding the sampling theorem (bearing in mind that we are using a normalised sampling frequency  $f_s = 1$  Hz or  $T = 1$  s in this course.) We will learn later in this chapter that the  $z$  transform and the DTFT satisfy symmetry properties, whose consequence is that the spectrum of real-valued signals obeys the relations

$$\begin{cases} |S(-\theta)| = |S(\theta)|, \\ \angle S(-\theta) = -\angle S(\theta). \end{cases} \quad (3.20)$$

In other words, the equivalent of a (Bode) frequency plot for discrete-time systems will only need to plot magnitude and phase for  $\theta$  from 0 to  $\pi$ , because its magnitude plot is even and its phase plot is odd, so there is no extra information in the spectrum for negative frequencies or, equivalently, for  $\theta$  from  $\pi$  to  $2\pi$ . For complex signals however, these relations do not hold and negative frequencies do contain relevant information which is not simply a copy of the information in positive frequencies<sup>3</sup>.

Since the DTFT of right-sided signals starting at zero is equivalent to the  $z$  transform on the unit circle, it is clear that the convolution property that we used as a justification

---

<sup>3</sup>You'll remember from 2P6 Communications that this is the property used in digital passband modulation to avoid the "double bandwidth" loss of amplitude modulation.

to derive the  $z$  transform must hold for the DTFT for such signals as well. In essence, the DTFT of right-sided signals starting at zero is an instance of our general  $D$  transform for  $D = e^{-j\theta}$ . For general, possibly two-sided signals, the link between signals and power series that we used to derive the general convolution property of discrete transforms doesn't hold (in mathematics, power series are one-sided by definition.) However, it can be shown that the convolution property still holds and hence holds in general for the DTFT irrespective of whether signals are one-sided or two-sided: the discrete convolution of signals  $u * s$  results in a multiplication of their DTFT  $U(\theta)S(\theta)$ . The DTFT also obeys a reverse convolution property, whereby the multiplication of discrete-time signals  $y_k = u_k s_k$  results in a periodic convolution of their DTFTs,

$$Y(\theta) = (U \circledast S)(\theta) = \frac{1}{2\pi} \int_{-\pi}^{\pi} U(\psi)S(\theta - \psi)d\psi. \quad (3.21)$$

You will prove this in an examples paper question about Fourier series and we will discuss later in this chapter how the DTFT relates to Fourier series. We will use this property later in the course when we discuss filter design.

In 2P6 Examples Paper 7, Question 5, you showed that the DTFT can be inverted using the expression (slightly modified to use our notation and normalised angular frequency  $\theta$ )

$$s_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\theta)e^{jk\theta}d\theta. \quad (3.22)$$

This will be useful when we investigate methods to invert the  $z$  transform.

For the remainder of 3F1, we will only ever use the one-sided  $z$  transform but often revert to the DTFT when handling 2-sided signals, so the rest of this chapter is very much about both transforms.

## 3.3 Properties of $z$ transforms and of the DTFT

### 3.3.1 Convolution property

We have already established that the  $z$  transform satisfies the convolution property, i.e., the  $z$  transform of

$$s_k = \sum_{n=-\infty}^{\infty} a_n b_{k-n}, \text{ equivalently denoted as } s = a * b, \quad (3.23)$$

is

$$S(z) = A(z)B(z). \quad (3.24)$$

*Example:* Just for fun, let's see what happens when we convolve a geometric series  $\{Bq^k\}_{k \geq 0}$  with itself,

$$s_k = \sum_{m=0}^k Bq^m Bq^{k-m} = B^2 q^k \sum_{m=0}^k q^m q^{-m} = B^2 q^k \sum_{m=0}^k 1 = B^2 (k+1) q^k \quad (3.25)$$

whose  $z$  transform, by the convolution property, is

$$S(z) = \frac{Bz}{z-q} \cdot \frac{Bz}{z-q} = \frac{B^2 z^2}{(z-q)^2} \quad (3.26)$$

If we denote  $B^2$  as  $A$ , we have shown that the  $z$  transform of  $\{A(k+1)q^k\}_{k \geq 0}$  is

$$S(z) = \frac{Az^2}{(z-q)^2}. \quad (3.27)$$

Bear this in mind as we explore other ways to obtain similar  $z$  transform pairs in the next few paragraphs, and properties that will allow us to reconcile the slightly different expressions.

As discussed in the previous section, the convolution property also applies to the DTFT. For the rest of this chapter, when we state that a property “also applies to the DTFT”, we mean that it applies even if the signals involved have non-zero values at negative times.

### 3.3.2 Linearity

The  $z$  transform is linear in the sense that, if the signals  $\{s_k^{(1)}\}$  and  $\{s_k^{(2)}\}$  have  $z$  transforms  $S_1(z)$  and  $S_2(z)$ , respectively, then the  $z$  transform of

$$\{A_1 s_k^{(1)} + A_2 s_k^{(2)}\}$$

is

$$A_1 S_1(z) + A_2 S_2(z)$$

for any numbers  $A_1$  and  $A_2$ . Another more subtle application of linearity is when a sequence  $\{s_k\}$  is defined in a manner that depends on a parameter  $q$ . Then the  $z$  transform of

$$\frac{d}{dq} s_k \text{ for all } k$$

is

$$\frac{d}{dq} S(z),$$

i.e., the derivative of a sequence transforms to the derivative of the  $z$  transform. If this is too abstract for you, you will see in the example below how and why it holds.

Both these linearity properties also apply to the DTFT.

*Example:* we can obtain useful  $z$  transforms by taking the derivative of the expression for the  $z$  transform of a geometric sequence (3.13) with respect to  $q$ , to obtain

$$\frac{d}{dq} \left( \sum_{k=0}^{\infty} Aq^k z^{-k} \right) = \sum_{k=0}^{\infty} (Akq^{k-1}) z^{-k} = \frac{d}{dq} \left( \frac{Az}{z-q} \right) = \frac{Az}{(z-q)^2}. \quad (3.28)$$

showing that the  $z$  transform of the sequence  $Akq^{k-1}$  for  $k \geq 0$  is  $Az/(z-q)^2$ . This is what we meant by linearity with respect to the derivative and the property is derived in (3.28) directly from the definition of the  $z$  transform.

We can now apply linearity by multiplying the  $z$  transform pair we obtained by  $q$ , so that the  $z$  transform of

$$s_k = q \cdot Aq^{k-1} = Akq^k \text{ for } k \geq 0 \quad (3.29)$$

is

$$S(z) = \frac{Aqz}{(z-q)^2}. \quad (3.30)$$

Note that this signal is 0 for  $k = 0$ .

You can repeat the derivative trick to obtain expressions for sequences  $Ak^m q^k$  for any  $m$  recursively or inverse transforms of  $1/(z-q)^m$ . Verify in your Information Data Book that you match the expressions given there for the inverse transform of  $z^m/(z-q)^m$  for the special case where  $q = 1$ .

Another signal of interest is the cosine signal  $s_k = \cos(\theta k)$  for  $k \geq 0$ . This can be re-written as

$$s_k = \cos(\theta k) = \frac{e^{j\theta k} + e^{-j\theta k}}{2} \quad (3.31)$$

and hence we use linearity to obtain the  $z$  transform by application of the  $z$  transform (3.13) we obtained for geometric sequences (complex in this case)

$$S(z) = \frac{z/2}{z - e^{j\theta}} + \frac{z/2}{z - e^{-j\theta}} \quad (3.32)$$

$$= \frac{(2z - e^{-j\theta} - e^{j\theta})z/2}{z^2 - (e^{j\theta} + e^{-j\theta})z + 1} \quad (3.33)$$

$$= \frac{z(z - \cos \theta)}{z^2 - 2z \cos \theta + 1} = \frac{1 - z^{-1} \cos \theta}{1 - 2z^{-1} \cos \theta - z^{-2}} \quad (3.34)$$

as listed in the Information Data Book (where they use the notation  $\theta = \omega_0 T$ ). The expression for  $\sin(\theta k)$  is obtained similarly.

### 3.3.3 Time shift properties

The time-shift properties are initially easier to understand using the DTFT. If we shift a sequence  $\{s_k\}_{k \geq m}$  by  $d$  (where  $d$  could be positive or negative) we obtain a new sequence

$$s'_k = s_{k-d}.$$

The shifted sequence's DTFT is

$$S'(\theta) = \sum_{k=-\infty}^{\infty} s'_k e^{-jk\theta} = \sum_{k=-\infty}^{\infty} s_{k-d} e^{-jk\theta} \quad (3.35)$$

$$= \sum_{k'=-\infty}^{\infty} s_{k'} e^{-j(k-d)\theta} = e^{-jd\theta} \sum_{k'=-\infty}^{\infty} s_{k'} e^{-jk'\theta} \quad (3.36)$$

$$= e^{-jd\theta} S(\theta) \quad (3.37)$$

where we used a variable substitution  $k' = k - d$  in the second line. Hence, for the DTFT,

$$\text{time-shift by } d \iff \text{multiplication by } e^{-jd\theta}. \quad (3.38)$$

Note that  $d$  can be positive or negative. In other words,

$$\text{right-shift by } |d| \text{ (delaying)} \iff \text{multiplication by } e^{-j|d|\theta} \quad (3.39)$$

and

$$\text{left-shift by } |d| \text{ (advancing)} \iff \text{multiplication by } e^{+j|d|\theta}. \quad (3.40)$$

It would be tempting to expect this result to extend easily to the  $z$  transform, with a shift by  $d$  resulting in a multiplication of the  $z$  transform by  $z^{-d}$ . This is indeed the case for the 2-sided  $z$  transform but not for the 1-sided  $z$  transform we use in this course, because:

- when right-shifting or delaying a sequence, samples that were in negative times and were not captured by the one-sided  $z$  transform now shift to non-negative times and hence need to be taken into account in the  $z$  transform of the shifted signal. Note that this is only an issue if the sequence had non-zero values at negative time indices.
- when left-shifting or advancing a sequence, samples in non-negative times that were accounted for in the  $z$  transform are now shifted to negative time indices and should no longer be accounted for in the  $z$  transform of the shifted sequence.

The resulting shift rules mirror the rules for the DTFT but they require correction terms to address the appearance or disappearance of samples from or into the negative index range, respectively.

The right-shift (delaying) expression hence becomes

$$S'(z) = z^{-|d|} S(z) + z^{-(|d|-1)} s_{-1} + z^{-(|d|-2)} s_{-2} + \dots + z^{-1} s_{-(|d|-1)} + s_{-|d|}. \quad (3.41)$$

and the left-shift (advancing) expression is

$$S'(z) = z^{|d|}S(z) - z^{|d|}s_0 - z^{|d|-1}s_1 - \dots - zs_{|d|-1}. \quad (3.42)$$

Both are in your Information Data Book (using a slightly different notation.)

The added terms in these expressions have caused confusion to many past generations of 3F1 students. I hope that, by showing you that this is simply an artefact of the 1-sided transform and does not happen for the DTFT, I was able to avert this confusion.

*Example:* consider the  $z$  transform pair  $\{s_k\} = \{A(k+1)q^k\}_{k \geq 0}$ ,  $S(z) = Az^2/(z - q)^2$  we obtained in (3.27) by convolving a geometric sequence with itself. If we shift the time-sequence by 1 to the left, we obtain the sequence

$$s'_k = s_{k-1} = Akq^{k-1} \text{ for } k \geq 1 \quad (3.43)$$

whose  $z$  transform, by the time-shift property, is

$$S'(z) = z^{-1}S(z) + s_{-1} = \frac{Az}{(z - q)^2}. \quad (3.44)$$

Note that we can view the shifted sequence as  $\{s'_k\} = \{Akq^{k-1}\}_{k \geq 0}$ , i.e., starting at zero, because the expression for  $k = 0$  computes to zero anyhow. We now use the linearity property again multiplying the sequence and its  $z$  transform by  $q$  to obtain that the  $z$  transform of  $\{Akq^k\}_{k \geq 0}$  is  $Aqz/(z - q)^2$ , in line with the result we obtained in (3.30) by using the linearity with respect to the derivative property.

### 3.3.4 Initial Value Theorem

The  $z$  transform satisfies the initial value property, in the sense that

$$\lim_{z \rightarrow \infty} S(z) = \lim_{z \rightarrow \infty} \left( s_0 + \frac{s_1}{z} + \frac{s_2}{z^2} + \dots \right) = s_0. \quad (3.45)$$

I took the derivation above straight out of past 3F1 lecture notes, fully aware that it is mathematically sloppy in that it is not completely clear what we mean by “ $z \rightarrow \infty$ ” as there is no such thing as infinity on the complex plane and there are many ways a complex number can grow “large”, e.g., it could grow radially from the origin, keeping its angle constant but growing only its magnitude, or it could spiral out from the center, etc. One could make the derivation more precise by picking one of these trajectories and deriving it in this context, but one would invariably arrive at the same conclusion as long as the “trajectory to infinity” is reasonable.

*Example:* consider the  $z$  transform of the geometric sequence  $\{s_k\}_{k \geq 0} = \{Aq^k\}_{k \geq 0}$  that we derived in (3.13),

$$S(z) = \frac{A}{1 - qz^{-1}}. \quad (3.46)$$

Clearly,  $\lim_{z \rightarrow \infty} S(z) = A = s_0$ .

### 3.3.5 Conjugate symmetry properties

All transforms that you've learned so far have a conjugate symmetry property: if a signal in the time domain is real, then its frequency domain transform obeys some form of symmetry, and vice versa. The  $z$  transform and the DTFT are no different.

If a signal  $\{s_k\}$  in the time domain is real, then its  $z$  transform evaluated at the conjugate  $\bar{z}$  obeys

$$S(\bar{z}) = \sum s_k \bar{z}^{-k} = \sum s_k \overline{z^{-k}} = \sum \overline{s_k z^{-k}} = \overline{S(z)} \quad (3.47)$$

hence the  $z$  transform is conjugate symmetric around the real axis, meaning that its value is conjugated if you negate the imaginary part of its argument. On the unit circle, this property becomes

$$S(e^{-j\theta}) = \overline{S(e^{j\theta})} \quad (3.48)$$

and applies also to the DTFT, for which  $S(-\theta) = \overline{S(\theta)}$  if the signal  $s$  is real. We already mentioned this property when we discussed the interpretation of the DTFT and of the  $z$  transform on the unit circle: for the spectrum of a real signal, the magnitude is an even function and the phase is an odd function.

The DTFT also has a reverse conjugate symmetry property: it's easy to see from the definition of the DTFT that a time-domain signal satisfying the condition

$$s_{-k} = \overline{s_k} \quad (3.49)$$

has a real-valued spectrum  $S(\theta)$ . This property sadly does not extend to our 1-sided  $z$  transform because the transform does not take signal values at negative times into account so the above symmetry property of a time domain signal has no bearing on its  $z$  transform.

The two symmetry properties of the DTFT can be combined to state that the spectrum of a real symmetric signal is real *and* symmetric around the real axis, i.e., an even real-valued function in the angular argument  $\theta$ .

*Example:* consider the signal  $(s_{-1}, s_0, s_1) = (-1, 1, -1)$  which is real and satisfies symmetry and hence also conjugate symmetry in the time domain. Its DTFT is

$$S(\theta) = e^{j\theta} + 1 + e^{-j\theta} = 1 + 2 \cos \theta \quad (3.50)$$

which is real-valued and even in  $\theta$ .

*Example:* Consider the complex oscillatory signal  $\{s_k\} = \{e^{j\psi k}\}_{k \geq 0}$ . This has  $z$  transform

$$S(z) = \frac{z}{z - e^{j\psi}} \quad (3.51)$$

which does not obey conjugate symmetry, since the signal  $\{s_k\}$  is not real. Now consider the signal

$$\{w_k\} = \{\cos \psi k\}_{k \geq 0} = \frac{1}{2}\{e^{j\psi k} + e^{-j\psi k}\} = \frac{1}{2}\{s_k\} + \frac{1}{2}\{\overline{s_k}\}. \quad (3.52)$$

This is now a real-valued signal and its  $z$  transform and DTFT should obey the conjugate symmetry property. We will try to verify this. By linearity,

$$W(z) = \frac{1}{2} \left[ \frac{z}{z - e^{j\psi}} + \frac{z}{z - e^{-j\psi}} \right] = \frac{2z(z - \cos \psi)}{z^2 - 2z \cos \psi + 1} \quad (3.53)$$

which matches the expression in the data book. Evaluating the intermediate expression for  $z = e^{j\theta}$ , we obtain

$$W(e^{j\theta}) = \frac{1}{2} \left[ \frac{1}{1 - e^{-j(\theta-\psi)}} + \frac{1}{1 - e^{-j(\theta+\psi)}} \right] \quad (3.54)$$

We also note that

$$W(e^{-j\theta}) = \frac{1}{2} \left[ \frac{1}{1 - e^{j(\theta+\psi)}} + \frac{1}{1 - e^{j(\theta-\psi)}} \right] \quad (3.55)$$

It's easy to verify that

$$\overline{1 - e^{-j(\theta-\psi)}} = 1 - e^{j(\theta-\psi)} \quad \text{and} \quad \overline{1 - e^{-j(\theta+\psi)}} = 1 - e^{j(\theta+\psi)} \quad (3.56)$$

and hence  $W(e^{-j\theta}) = \overline{W(e^{j\theta})}$  which establishes conjugate symmetry, or in terms of the DTFT,  $W(-\theta) = \overline{W(\theta)}$ . A similar result can be derived for the signal  $\{\sin \psi k\}_{k \geq 0}$ .

### 3.3.6 Scaling with a geometric sequence

Let  $\{s_k\}_{k \geq 0}$  be a signal with  $z$  transform  $S(z)$  and let

$$s'_k = q^k s_k \quad (3.57)$$

be the signal  $\{s_k\}_{k \geq 0}$  scaled with a geometric sequence  $\{q^k\}_{k \geq 0}$ . The resulting  $z$  transform is

$$S'(z) = \sum_{k=0}^{\infty} s_k q^k z^{-k} = \sum_{k=0}^{\infty} s_k (q^{-1}z)^{-k} = S(q^{-1}z). \quad (3.58)$$

This property is in the data book and can be very useful when an expression appears to be missing from the data book but can be recovered by combining an existing expression

with this property.

*Example:* The data book gives us (in slightly modified form using  $T = 1$ ) the  $z$  transform pair

$$\{k\}_{k \geq 0} \longleftrightarrow \frac{z}{(z-1)^2}. \quad (3.59)$$

Using the geometric scaling property, we obtain the  $z$  transform pair

$$\{kq^k\}_{k \geq 0} \longleftrightarrow \frac{q^{-1}z}{(q^{-1}z-1)^2} = \frac{zq}{(z-q)^2} \quad (3.60)$$

which is again a similar result as the expression we obtained in 3.30 by other means.

### 3.4 Inverting the $z$ transform

There is more than one approach to inverting the  $z$  transform. Before we proceed to discuss the various approaches, bear in mind that the  $z$  transform just maps discrete-time signals, i.e., sequences, to power series or polynomials. As a result, the  $z$  transform is much easier to invert than continuous-time transforms. If you can think creatively of any way to re-write an expression in the  $z$  domain as a power series or polynomial, then this power series or polynomial is a unique representation of the expression and hence you can just read out the coefficients to recover the time-domain values of your discrete-time signal.

The usual approaches to inverting the  $z$  transform are the following:

1. by inspection (as for the Laplace transform). If your expression in the  $z$  domain is a rational expression, i.e., of the form

$$S(z) = A \frac{1 + a_1 z^{-1} + \dots + a_n z^{-n}}{1 + b_1 z^{-1} + \dots + b_m z^{-m}}, \quad (3.61)$$

you can use *partial fractions*<sup>4</sup> to re-write it as a sum of terms for which we computed the  $z$  transform in the previous section, e.g., geometric sequences and sinusoids, or any signal for which there are  $z$  transform pairs in the Information Data Book;

2. for rational  $z$  transforms of the form (3.61), it is also possible to use polynomial (long) division to obtain a power series;
3. as for the Laplace transform, there is an inverse integral in the complex domain that allows the transform to be inverted based on the “residue theorem” of complex calculus which is beyond the scope of the Cambridge Engineering Tripos;

---

<sup>4</sup>If you need help with partial fractions, I include a detailed revision primer at the end of this section.

4. in some cases, you can use a power series expansion or Taylor expansion of a function in the  $z$  domain to invert the  $z$  transform. We have always used power series expansions as listed in the Mathematics Data Book on complex functions as well as on real functions, even though I don't believe you have ever seen a formal derivation of such series for complex functions. In fact, power series expansions rely on the Taylor expansion in the complex domain, which in itself relies on the "dreaded" residue theorem which remains outside the scope of our course, so you have secretly been using the residue theorem even though we never taught it in the course;
5. you can use the inverse DTFT to invert the  $z$  transform. This is also a hidden way to unknowingly use the residue theorem. The Cambridge engineering course probably deserves an award for teaching 1001 ways to use the residue theorem without actually teaching complex calculus... ☺

We will now study examples of each of the methods listed, skipping (3) the general application of the residue theorem which, as we said, is beyond the scope of 3F1.

*Example:* consider the rational  $z$  transform

$$S(z) = \frac{3z^2 - 3z + 1}{z^2 - 3z + 2}. \quad (3.62)$$

This a so-called "improper" fraction because the degree of the numerator is not strictly smaller than the degree of the denominator. Hence, we first need to factor out a constant term to obtain

$$S(z) = \frac{3z^2 - 9z + 6}{z^2 - 3z + 2} + \frac{6z - 5}{z^2 - 3z + 2} = 3 + \frac{6z - 5}{(z - 1)(z - 2)} \quad (3.63)$$

then apply partial fraction decomposition to the remaining "proper" fraction using our trusted "cover-up" rule to give

$$S(z) = 3 - \frac{1}{z - 1} + \frac{7}{z - 2} = 3 - z^{-1} \frac{1}{1 - z^{-1}} + 7z^{-1} \frac{1}{1 - 2z^{-1}} \quad (3.64)$$

where in the last step we have re-written the terms explicitly as a time shift (product with  $z^{-1}$ ) times the expression for a geometric sequence that we computed in (3.13). We conclude that the signal is

$$s_k = \begin{cases} 3 & \text{for } k = 0, \\ -1 + 7 \cdot 2^{k-1} & \text{for } k > 0 \end{cases} \quad (3.65)$$

and unknown for times  $k < 0$ .

Note that the only "improper" partial fraction we'd ever encounter when inverting a 1-sided  $z$  transform is the one we've seen in this example, where the degrees of

numerator and denominator are the same and you need to factor out a constant term. If the degree of the numerator were larger than that of the denominator, you'd know that whoever asked you to invert this  $z$  transform must have made a mistake because that would result in terms with positive powers of  $z$  which cannot possibly be the outcome of a 1-sided  $z$  transform.

*Example:* we can also tackle the previous example using polynomial long division. There is a wonderful L<sup>A</sup>T<sub>E</sub>X package that displays a polynomial long division, but sadly it cannot cope with negative powers and stops with a remainder when it reaches a polynomial with a constant term. Formatting a long division without this package would be a nightmare, so my solution here is to pre-multiply the numerator by a power, then divide the solution by the same power, i.e.,

$$\frac{3z^2 - 3z + 1}{z^2 - 3z + 2} = \frac{z^p(3z^2 - 3z + 1)}{z^2 - 3z + 2} \cdot z^{-p} \quad (3.66)$$

to get the polynomial division up to  $p$  positions. Here's an example for  $p = 4$ :

$$\begin{array}{r} \phantom{z^2 - 3z + 2) } \phantom{3z^6 - 3z^5} + z^4 \\ \phantom{z^2 - 3z + 2) } \phantom{3z^6 - 3z^5} + 6z^3 + 13z^2 + 27z + 55 \\ \hline z^2 - 3z + 2) \phantom{3z^6 - 3z^5} + z^4 \\ \phantom{z^2 - 3z + 2) } \phantom{3z^6 - 3z^5} - 3z^6 + 9z^5 - 6z^4 \\ \hline \phantom{z^2 - 3z + 2) } \phantom{3z^6 - 3z^5} 6z^5 - 5z^4 \\ \phantom{z^2 - 3z + 2) } \phantom{3z^6 - 3z^5} - 6z^5 + 18z^4 - 12z^3 \\ \hline \phantom{z^2 - 3z + 2) } \phantom{3z^6 - 3z^5} \phantom{6z^5 - 5z^4} 13z^4 - 12z^3 \\ \phantom{z^2 - 3z + 2) } \phantom{3z^6 - 3z^5} \phantom{6z^5 - 5z^4} - 13z^4 + 39z^3 - 26z^2 \\ \hline \phantom{z^2 - 3z + 2) } \phantom{3z^6 - 3z^5} \phantom{6z^5 - 5z^4} \phantom{13z^4 - 12z^3} 27z^3 - 26z^2 \\ \phantom{z^2 - 3z + 2) } \phantom{3z^6 - 3z^5} \phantom{6z^5 - 5z^4} \phantom{13z^4 - 12z^3} - 27z^3 + 81z^2 - 54z \\ \hline \phantom{z^2 - 3z + 2) } \phantom{3z^6 - 3z^5} \phantom{6z^5 - 5z^4} \phantom{13z^4 - 12z^3} \phantom{27z^3 - 26z^2} 55z^2 - 54z \\ \phantom{z^2 - 3z + 2) } \phantom{3z^6 - 3z^5} \phantom{6z^5 - 5z^4} \phantom{13z^4 - 12z^3} \phantom{27z^3 - 26z^2} - 55z^2 + 165z - 110 \\ \hline \phantom{z^2 - 3z + 2) } \phantom{3z^6 - 3z^5} \phantom{6z^5 - 5z^4} \phantom{13z^4 - 12z^3} \phantom{27z^3 - 26z^2} \phantom{55z^2 - 54z} 111z - 110 \end{array} \quad (3.67)$$

We now multiply the outcome  $3z^4 + 6z^3 + 13z^2 + 27z + 55$  by  $z^{-p} = z^{-4}$  to obtain

$$S(z) = 3 + 6z^{-1} + 13z^{-2} + 27z^{-3} + 55z^{-4} + \dots \quad (3.68)$$

corresponding to the signal  $s = (3, 6, 13, 27, 55, \dots)$  which is in line with the expression we obtained in (3.65), i.e.  $s = (3, 7 \cdot 1 - 1, 7 \cdot 2 - 1, 7 \cdot 4 - 1, 7 \cdot 8 - 1, \dots)$ . Note that unlike the partial fractions approach, polynomial division can only give us a few terms of the signal and not an expression for the whole signal.

*Example:* consider the expression in the  $z$  domain

$$S(z) = e^{z^{-1}}. \quad (3.69)$$

This is not a rational function and hence there is no finite linear difference equation for which the above is a homogenous solution, so we are unlikely to encounter such an expression in our study of linear systems. However, we can still use the Power series expansion on page 5 of the Mathematics Data Book to express the exponential as

$$S(z) = 1 + z^{-1} + \frac{z^{-2}}{2} + \frac{z^{-3}}{6} + \frac{z^{-4}}{24} + \dots \quad (3.70)$$

and hence obtain the corresponding time domain signal

$$\{s_k\}_{k \geq 0} = \left(1, 1, \frac{1}{2}, \frac{1}{6}, \frac{1}{24}, \dots\right) = \{1/(k!)\}_{k \geq 0}. \quad (3.71)$$

As mentioned, you have been asked since your first year to take for granted that the power series expansions in the Mathematics Data Book work for complex arguments as well, but the proof of this fact relies on complex Taylor series that rely on the residue theorem of complex calculus that is not covered in the Cambridge Engineering tripos.

*Example:* inverting the  $z$  transform using the DTFT inversion formula is tedious and in general the methods we tried above are far preferable (in particular the partial fractions method.) Later in this course, we will use the DTFT inversion formula to determine the ideal low-pass filter.

For now, we will consider only a very simple signal to understand better how the DTFT inversion formula works and to verify that it gets the right result. Let's consider the signal with  $z$  transform

$$S(z) = Az^{-m} \quad (3.72)$$

whose inverse  $z$  transform is simply the signal whose value is  $A$  at time  $m$  and 0 everywhere else. The DTFT of this signal is obtained by setting  $z = e^{j\theta}$ , hence

$$S(\theta) = Ae^{-jm\theta}. \quad (3.73)$$

Applying the DTFT inversion formula

$$s_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\theta) e^{jk\theta} d\theta = \frac{A}{2\pi} \int_{-\pi}^{\pi} e^{-jm\theta} e^{jk\theta} d\theta = \frac{A}{2\pi} \int_{-\pi}^{\pi} e^{j(k-m)\theta} d\theta \quad (3.74)$$

which clearly gives  $A$  for  $k - m = 0$ , and 0 for  $k - m \neq 0$  because it is the integral of a complex harmonic function over an integer number of periods, which is always zero.

As promised, there follows a revision of partial fractions.

**Partial fractions revision:**

Here's a short summary of what many of you have learned in secondary school about partial fractions:

- An expression

$$f(x) = \frac{b(x)}{a(x)} = \frac{b_mx^m + \dots + b_2x^2 + a_1x + a_0}{a_nx^n + \dots + a_2x^2 + a_1x + a_0} \quad (3.75)$$

is a proper partial fraction if the degree  $m$  of  $b(x)$  is strictly smaller than the degree  $n$  of  $a(x)$ . If  $a(x)$  can be factorised as  $a(x) = (x - p_1)(x - p_2) \cdots (x - p_n)$ , where all the poles  $p_i$  are distinct, then one can obtain an expression of the form

$$f(x) = \frac{\alpha_1}{x - p_1} + \frac{\alpha_2}{x - p_2} + \dots + \frac{\alpha_n}{x - p_n}. \quad (3.76)$$

- The  $\alpha_i$  are constants and can be obtained by the “cover-up method”: write  $f(x)$  with the factorised denominator

$$f(x) = \frac{b(x)}{a(x)} = \frac{b_mx^m + \dots + b_2x^2 + b_1x + a_0}{(x - p_1)(x - p_2) \cdots (x - p_n)} \quad (3.77)$$

and to obtain  $\alpha_i$ , insert the value of  $p_i$  for  $x$  and evaluate the expression while covering up the factor  $(x - p_i)$  in the denominator, i.e.,

$$\alpha_i = \frac{b_mp_i^m + \dots + b_2p_i^2 + a_1p_i + a_0}{(p_i - p_1)(p_i - p_2) \cdots (p_i - p_n)} \quad (3.78)$$

- If you were wondering why the cover-up method works, it is simply because

$$\lim_{x \rightarrow p_i} (x - p_i)f(x) = \alpha_i \quad (3.79)$$

Taking this limit on the partial fraction expansion (3.76) clearly shows why the limit holds, while taking the limit on (3.77) shows why the cover-up method works: multiplying by  $(x - p_i)$  is equivalent to “covering up” or cancelling the pole in  $p_i$ .

- if a pole  $p_i$  has a multiplicity  $k > 1$ , i.e.,  $a(x) = (x - p_i)^k a'(x)$  (where  $a'(x)$  is just the expression with all the other poles) then there are two ways to express the partial fraction expansion, either

$$f(x) = f'(x) + \frac{\alpha_{i1} + \alpha_{i2}x + \dots + \alpha_{ik}x^{k-1}}{(x - p_i)^k}, \quad (3.80)$$

where  $f'(x)$  is the remaining partial fractions terms resulting from the remaining poles, or

$$f(x) = f'(x) + \frac{\alpha_{i1}}{x - p_i} + \frac{\alpha_{i2}}{(x - p_i)^2} + \dots + \frac{\alpha_{ik}}{(x - p_i)^k}. \quad (3.81)$$

The cover-up rule can recover the term  $\alpha_{ik}$  but the remaining numerators must be found manually. Again, to understand why the cover-up rule works for  $\alpha_{ik}$ , think of what happens when you take  $\lim_{x \rightarrow p_i} (x - p_i)^k f(x)$ .

- if the degree  $m$  of the numerator  $b(x)$  is equal or greater to the degree  $n$  of the denominator  $a(x)$  then there is no proper partial fraction expansion. Consider the expression for the partial fraction expansion (3.76) and you'll immediately see that the largest possible degree for the numerator if you go back to a rational form is  $n - 1$ , because you'll end up multiplying each  $\alpha_i$  by a polynomial of degree  $n - 1$ , and the sum of polynomials of degree  $n - 1$  is  $n - 1$ , or less if the leading coefficients cancel when summing.
- in cases where  $m \geq n$ , you can take a few steps of polynomial division until you obtain a remainder of degree  $n - 1$  or less, and then use a partial fraction expansion on the rest, i.e.,

$$f(x) = \beta_{m-n}x^{m-n} + \dots + \beta_1x + \beta_0 + \sum_{i=1}^n \frac{\alpha_i}{x - p_i}, \quad (3.82)$$

assuming non-repeated poles without loss of generality<sup>5</sup>. In particular, when the degree of the numerator is the same as the degree of the denominator, the partial fraction expansion consists of a constant term followed by a proper partial fraction expansion

$$f(x) = \beta_0 + \sum_{i=1}^n \frac{\alpha_i}{x - p_i}. \quad (3.83)$$

In the next section, we will see that for linear systems, we will only ever deal with expressions where  $m \leq n$ , so we will at most have to use (3.83) and never the more general (3.82).

## 3.5 Relation to other transforms

As discussed, the DTFT can be viewed as one period of the Fourier transform of the train of impulses corresponding to the sampled bandlimited continuous function. This is indeed

---

<sup>5</sup>This means that the assumption is only made to keep the notation manageable but it would be easy, given what we have learned, to extend these statement to repeated poles.

how the DTFT was presented to you in 2P6. The train of impulses is a hypothetical construct that doesn't exist in reality and whose purpose was simply to prove the sampling theorem, showing that the original continuous-time signal can be exactly reconstructed from the discrete-time signal obtained from its samples provided there is no aliasing, i.e., the continuous signal is strictly bandlimited with bandwidth  $B < f_s/2$ , where  $f_s$  is the sampling frequency. The  $z$  transform shares this analogy with the Fourier transform provided the signals considered are zero at negative times<sup>6</sup>.

The relation between  $z$  transform and train of impulses can be extended to the Laplace transform. Consider a continuous signal  $w(t)$  and the corresponding train of impulses

$$w_s(t) = \sum_{k=0}^{\infty} w(kT)\delta(t - kT) \quad (3.84)$$

obtained from its samples at non-negative time instants spaced every  $T$  s. The Laplace transform of  $w_s(t)$  is

$$W_s(s) = \int_0^{\infty} w_s(t)e^{-st} dt = \int_0^{\infty} \sum_{k=0}^{\infty} w(kT)\delta(t - kT)e^{-st} dt \quad (3.85)$$

$$= \sum_{k=0}^{\infty} w(kT) \int_0^{\infty} \delta(t - kT)e^{-st} dt \quad (3.86)$$

$$= \sum_{k=0}^{\infty} w(kT)e^{-skT} = \sum_{k=0}^{\infty} w(kT)z^{-k} \text{ for } z = e^{sT}. \quad (3.87)$$

In other words, the Laplace transform of the train of impulses is the  $z$  transform evaluated at  $s = e^{sT}$ . This mapping of  $s$  to  $z$  will cause us headaches throughout 3F1, making the discrete-time equivalents of Nyquist and similar diagrams subtly different from their analog counterparts, and posing hurdles when converting or modeling analogue filters in the discrete-time domain. Remember that this is all down to a silly convention that we prefer geometric sequences in discrete-time but exponential functions in continuous time, even though they are essentially the same thing.

The final and rather surprising transform that the  $z$  transform and the DTFT relate to is the Fourier series. In 1P4, we never talked about the Fourier series as a “transform”. It certainly is a transform, sharing many of the properties that we ascribe to transforms: Fourier series have a (periodic) convolution property, and time shift property, a conjugate symmetry property, etc. The analogy between Fourier series and DTFT is much closer than you may have imagined:

- complex Fourier series map a continuous function  $f(t)$  defined over an interval (or,

---

<sup>6</sup>most signals that are zero at negative times cannot be strictly bandlimited, but as engineers we can consider the aliasing noise to be negligible for signals of interest, as long as they don't have a significant discontinuity at  $t = 0$ .

equivalently, a periodic function) to a set of coefficients and back

$$c_k = \frac{1}{T} \int_0^T f(t) e^{-j2\pi kt/T} dt \text{ and } f(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi kt/T}. \quad (3.88)$$

The coefficients can be viewed as the “discrete spectrum” of  $f(t)$ .

- the DTFT maps a discrete time-domain signal  $s_k$  to a continuous spectrum  $S(\theta)$  defined over an interval (or, equivalently, a periodic spectrum) and back

$$S(\theta) = \sum_{k=-\infty}^{\infty} s_k e^{-j\theta k} \text{ and } s_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\theta) e^{j\theta k} d\theta. \quad (3.89)$$

You’ll notice that the two transforms are essentially the same, with the time domain and frequency domain swapped: the Fourier series operates on a periodic or interval-bound function and has a discrete spectrum, whereas the DTFT operates on a discrete-time signal and has a periodic or interval-bound spectrum. By setting  $T = 2\pi$  and  $\theta = 2\pi t/T$ , the expressions become identical except for a sign difference in the transform and its inverse, which brings us to a final convention worth noting:

**Minus plus convention:**

All transforms (Fourier series, Laplace, Fourier transforms, the DTFT, the DFT and the  $z$  transform) have a minus sign in the exponent going from time to frequency domain, and a plus sign in the exponent going back from the frequency domain to the time domain.

This again is merely a convention and a matter of preference: the world would have been just as pretty, the sun would have shined just as bright and the birds would have chirped just as sharply at dawn if we had adopted the inverse convention of having a plus to go from time to frequency and a minus to go from frequency to time. . .

# Chapter 4

## Linear systems properties and stability

Having introduced the  $z$  transform in the previous chapter, we now have all the tools needed to replicate the methods you learned in previous years for analogue signal processing. In particular, we can now draw a diagram for discrete-time signal processing like the one we drew in Figure 1.1 for continuous-time systems. The equivalent diagram for discrete-time systems is drawn in Figure 4.1.

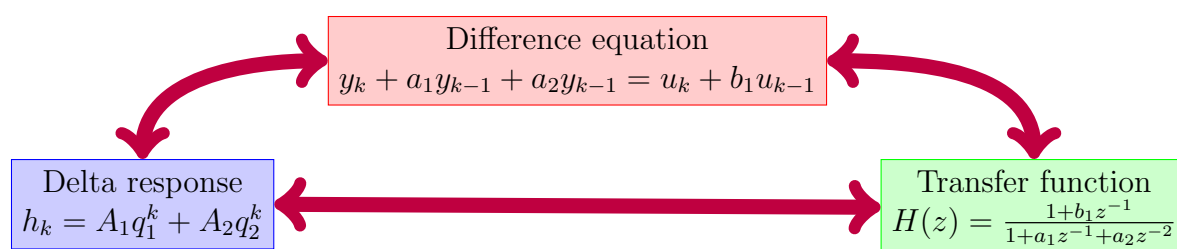


Figure 4.1: The 3 equivalent descriptions of a discrete-time LTIS

We will spend the next few sections investigating properties of linear systems, starting in the next section with a review of how we can circulate on every arrow in every direction in the diagram of Figure 4.1, i.e., going from a difference equation to the transfer function, from the transfer function to the delta response, etc.

### 4.1 The transfer function of a linear system

The transfer function of a linear system is the  $z$  transform of its delta response, just as the transfer function of a continuous linear system is the Laplace or Fourier transform of its impulse response. Obviously, going from the delta response to the transfer function and back simply involves taking the  $z$  transform and its inverse and we've just had a whole chapter on this. Going from the difference equation to the delta response requires solving

a difference equation with a pulse as its input, something you've learned to do in IA Paper 4 Mathematical Methods in Michaelmas term. Going back from a delta response to guess the corresponding difference equation has not been covered but you should be able to guess how to do this without too much difficulty.

In this section, we will discuss how to go directly from a difference equation to a transfer function. Consider the linear difference equation with constant coefficients

$$y_k + a_1 y_{k-1} + \dots + a_n y_{k-n} = b_0 u_k + b_1 u_{k-1} + \dots + b_m u_{k-m} \text{ for } k \geq 0. \quad (4.1)$$

If the signals  $\{y_k\}_{k \geq 0}$  and  $\{u_k\}_{k \geq 0}$  both start at 0, then we can easily apply the linearity properties of the  $z$  transform to transform (4.1) into

$$Y(z) + a_1 z^{-1} Y(z) + \dots + z^{-n} Y(z) = b_0 U(z) + b_1 z^{-1} U(z) + \dots + b_m z^{-m} U(z) \quad (4.2)$$

and hence

$$H(z) = \frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}. \quad (4.3)$$

Not all difference equations are given in their canonical form (4.1). In the example below will demonstrate a conversion from a difference equation to the output of a system in a slightly more challenging scenario.

*Example:* the difference equation

$$y_{k+2} - \frac{3}{2} y_{k+1} + \frac{1}{2} y_k = u_{k+1} - u_k \text{ for } k \geq 0 \quad (4.4)$$

describes a linear system with initial conditions on the output  $y_0 = 1$  and  $y_1 = 0$ . Find the output signal  $\{y_k\}$  when the input is a unit step signal  $\{u_k\} = \{1\}_{k \geq 0}$ .

We transform the difference equation into the  $z$  domain to obtain

$$z^2 Y(z) - z^2 y_0 - z y_1 - \frac{3}{2} (z Y(z) - z y_0) + \frac{1}{2} Y(z) = z U(z) - z u_0 - U(z) \quad (4.5)$$

and hence

$$Y(z) = \frac{z-1}{z^2 - \frac{3}{2}z + \frac{1}{2}} U(z) + \frac{z^2 y_0 + z y_1 - \frac{3}{2} z y_0 - z u_0}{z^2 - \frac{3}{2}z + \frac{1}{2}}. \quad (4.6)$$

The  $z$  transform of the unit step signal is  $U(z) = z/(z-1)$  and  $u_0 = 1$ , so we obtain

$$Y(z) = \frac{z-1}{z^2 - \frac{3}{2}z + \frac{1}{2}} \cdot \frac{z}{z-1} + \frac{z^2 - \frac{3}{2}z - z}{z^2 - \frac{3}{2}z + \frac{1}{2}} \quad (4.7)$$

$$= \frac{z^2 - \frac{3}{2}z}{(z - \frac{1}{2})(z - 1)} = \frac{2z}{z - \frac{1}{2}} - \frac{z}{z - 1} \quad (4.8)$$

and hence finally the system response is

$$y_k = 2 \left( \frac{1}{2} \right)^k - 1 = 2^{1-k} - 1 \text{ for } k \geq 0. \quad (4.9)$$

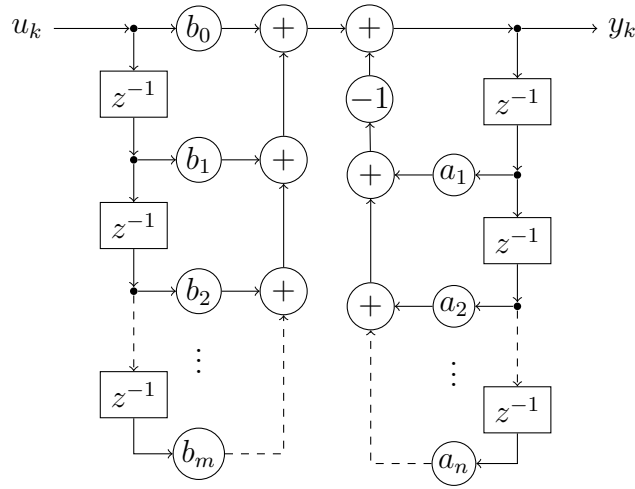


Figure 4.2: The Direct Form I implementation of the canonical difference equation (4.1)

Consider again the canonical difference equation (4.1). Re-writing the equation as

$$y_k = b_0 u_k + b_1 u_{k-1} + \dots + b_m u_{k-m} - a_1 y_{k-1} - \dots - a_n y_{k-n} \text{ for } k \geq 0. \quad (4.10)$$

we can deduce a linear circuit structure that implements the difference equation, as shown in Figure 4.2. In the figure, the circled coefficients (including  $-1$ ) represent multipliers, while the boxed  $z^{-1}$  represent “delay by 1” operators.

Linear circuit theory allows the two halves of the linear circuit in Figure 4.2 to be swapped, thereby giving the so-called “direct form II” that reduces the number of delay elements needed from  $m+n$  to  $\max\{m, n\}$ . I mention this in passing for general interest as it is not my intention in 3F1 to teach you linear circuit theory. The reason I’m asking you to view this circuit diagram is that it shows nicely how the coefficients  $b_0, b_1, \dots, b_m$  are part of a feed-forward circuit, i.e., they feed an adder that lies in front of them in the circuit going from input to output, whereas the coefficients  $a_1, a_2, \dots, a_n$  must be implemented using feedback, i.e., they loop back to an adder that lies before them in the circuit going from output to input.

Let us now return to the transfer function of the canonical difference equation

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}. \quad (4.11)$$

If all the  $a_k$  for  $k = 1, 2, \dots, n$  are zero, then the associated circuit has no feedback, the transfer function becomes

$$H(z) = b_0 + b_1 z^{-1} + \dots + b_m z^{-m}, \quad (4.12)$$

its associated delta response is  $(b_0, b_1, \dots, b_m)$ , a Finite Impulse Response (FIR) system with  $m+1$  coefficients. On the other hand, if the  $a_k$  coefficients are non-zero, then there is feedback in the circuit and the system has an Infinite Impulse Response (IIR), meaning that

the power series emanating from the polynomial division of  $b(z) = b_0 + b_1z^{-1} + \dots + b_mz^{-m}$  by  $a(z) = 1 + a_1z^{-1} + \dots + a_nz^{-n}$  is a true power series that has non-zero coefficients for index values going to infinity.

When dealing with transfer functions of linear systems defined via difference equations in their canonical form (4.1), we will often have to use partial fractions and juggle between expressions in  $z^{-1}$  and expressions in  $z$ , which can be rather confusing. To help you make sense of it, I have prepared a short partial fractions primer with a focus on what happens when going from  $z^{-1}$  to  $z$ .

### Partial fractions for expressions in $z^{-1}$ :

- Consider the canonical transfer function of a system

$$G(z) = \frac{b(z)}{a(z)} = \frac{b_0 + b_1z^{-1} + \dots + b_mz^{-m}}{1 + a_1z^{-1} + \dots + a_nz^{-n}}. \quad (4.13)$$

Note that  $n$  and  $m$  are *not* the degrees of the numerator and denominator polynomials  $b(z)$  and  $a(z)$ , respectively. To understand this, assume for now that  $m < n$  and multiply the top and bottom of the equation by  $z^n$  to obtain

$$G(z) = \frac{b_0z^n + b_1z^{n-1} + \dots + b_mz^{n-m}}{z^n + a_1z^{n-1} + \dots + a_n}. \quad (4.14)$$

Hence, the degrees of numerator and denominator are equal, and the standard expansion is

$$G(z) = b_0 + \sum_{i=1}^n \frac{\alpha_i}{z - p_i} \quad (4.15)$$

assuming non-repeated poles without loss of generality. This corresponds to a constant term  $b_0$  at  $k = 0$ , followed by a sum of geometric sequences  $z^{-1}\alpha_i z / (z - p_i)$  delayed to start at  $k = 1$ . This isn't very elegant... There is another way: as  $m < n$ , there is no constant term in the numerator, i.e., all terms include a power of  $z$ . Hence, we can re-write  $G(z)$  as

$$G(z) = z \left( \frac{b_0z^{n-1} + b_1z^{n-2} + \dots + b_mz^{n-m-1}}{z^n + a_1z^{n-1} + \dots + a_n} \right). \quad (4.16)$$

Now the expression in brackets has a numerator of degree one less than the degree  $n$  of the denominator, which has a proper partial fractions expansion, so expanding  $G(z)$  assuming no repeated poles gives

$$G(z) = z \left( \sum_{i=1}^n \frac{\alpha_i}{z - p_i} \right) = \sum_{i=1}^n \frac{\alpha_i z}{z - p_i} \quad (4.17)$$

which is a sum of  $z$  transforms of geometric sequences starting at  $k = 0$  without the need for a separate constant term at  $k = 0$ . A lot more elegant...

- If  $m \geq n$ , we cannot use this trick. You can still multiply the numerator and denominator by  $z^m$  to obtain a rational function with equal degree  $m$  in the numerator and denominator, and  $m - n$  repeated poles at  $z = 0$  (those aren't a problem.) This can be expanded as a constant  $b_0$  and partial fractions.

XXXXXXXXXX

Before we end this section, it is worth giving some thought to causality. The canonical difference equation we introduced in (4.1) always results in a causal system because the coefficient in front of  $y_k$  is 1 and coefficients on the right only multiply current and past values of the input. If we relaxed this convention and allowed for an arbitrary coefficient  $a_0$  to multiply  $y_k$ , then this coefficient could be zero and the system would no longer be causal if  $b_0 \neq 0$ , because the output at time  $k - 1$  depends on the input at time  $k$ . The delta response of such a system can still be determined and will include terms at negative times  $k < 0$ . However, the  $z$  transform of this delta response loses its meaning as the transfer function of the system, because it can no longer be used to determine the output from the input by multiplication in the  $z$  domain. As discussed earlier, the 1-sided  $z$  transform is not suited for handling non-causal systems. One would have to work with the 2-sided  $z$  transform which is beyond the scope of 3F1.

## 4.2 Poles of the transfer function and BIBO stability

You already know the concept of stability for continuous-time system and we have touched upon stability for discrete-time systems in Chapter 2. In broad terms, a system is stable if its transient response decays, i.e., the complementary solution of its associated difference equation tends to zero and the system thereby “forgets” its initial conditions. We will make this statement more precise in this section, and study how it relates to properties of a system’s delta response and transfer function.

We begin by stating a formal definition of what we mean by a system that is stable.

**Definition 4.1** *A system is Bounded-Input Bounded-Output (BIBO) stable if, for any bound  $\beta_i$  on its input signals, there exists a bound  $\beta_o$  such that, for any input signal  $\{x_k\}$  satisfying  $|x_k| \leq \beta_i$  for all  $k$ , the corresponding output signal  $\{y_k\}$  satisfies  $\beta_o$  for all  $k$ .*

This definition may appear unnecessarily formal but there is a good reason why such formality is necessary and I will try to explain why:

- a simplistic interpretation of the definition says that if an input signal  $\{x_k\}$  is bounded, i.e., does not tend to  $+\infty$  or  $-\infty$  (or, if its sign keeps changing or if it is complex, its magnitude doesn't tend to infinity,) then the output must also be bounded, i.e., not tend to infinity in magnitude. This interpretation isn't wrong: if you can find a bounded input signal that results in an unbounded output signal, you will have proven that the system is unstable;

- however, there are unstable systems where it isn't possible to find a specific bounded signal that results in a specific unbounded output, i.e., every bounded input gives a bounded output. However, it is possible to find a series of signals all bounded by a common upper bound  $\beta_i$  for which the largest magnitude of the output grows unbounded. In other words, while each output signal is bounded, the maximum magnitude per output signal keeps growing so it isn't possible to find an overall upper bound  $\beta_o$  on all the output signals. A system for which there exists such a series of input signals is unstable.

With this formal definition in hand, we now state the stability theorem that ties stability to properties of a system's delta response and transfer function:

**Theorem 4.1** *Conditions for stability of a discrete-time system* Let  $G$  be a discrete-time linear system with transfer function

$$G(z) = \frac{b_0 z^m + \dots + b_m}{z^n + a_1 z^{n-1} + \dots + a_n} = \frac{b(z)}{a(z)} \quad (4.18)$$

with  $m \leq n$  and no common factors between  $a(z)$  and  $b(z)$ , and let the system's delta response be  $\{g_k\}_{k \geq 0}$ . Then the following statements are equivalent:

1. the system  $G$  is stable;
2. All the roots  $p_i$  of  $a(z)$  satisfy  $|p_i| < 1$ ;
3.  $\sum_{k=0}^{\infty} |g_k|$  is finite

The roots of the denominator polynomial  $a(z)$  of the transfer function are called the "poles" of the system.

The theorem provides a way to verify whether a system is stable from its transfer function (poles inside the unit circle) or from its delta response (sum of magnitudes/absolute values must be finite). We will prove the theorem as follows:

- (i) show that 1. implies 2., i.e., that a stable system has its poles inside the unit circle
- (ii) show that 2. implies 3., i.e., that a transfer function with poles inside the unit circle corresponds to a delta response whose sum of magnitudes/absolute values is finite
- (iii) show that 3. implies 1., i.e., that a system with a delta response whose sum of magnitudes/absolute values is stable

Since not all Engineering students may have been taught logic and formal mathematical proofs, I decided to include this short manual before engaging in the proof:

### A short primer in logic and mathematical proofs:

- Implication:  $A \implies B$  (pronounced “ $A$  implies  $B$ ”) means that if statement  $A$  is true, then statement  $B$  must be true. The reverse doesn’t necessarily hold: it is possible for  $B$  to be true but  $A$  to be false. For example:
  - $x = 2 \implies x^2 = 4$
  - but  $x^2 = 4 \not\implies x = 2$  (because  $x$  could be  $-2$ )
- Implication is transitive:  $A \implies B \implies C$  implies  $A \implies C$ .
- Equivalence:  $A \iff B$  (pronounced “ $A$  if and only if  $B$ ” or “ $A$  equivalent to  $B$ ”) means  $A \implies B$  and  $B \implies A$ .  $A$  is true if and only if  $B$  is true.
- To prove equivalence, you often need to prove the two implications  $A \implies B$  and  $B \implies A$ . Alternatively, if you can prove implications “around the circle”  $A \implies B \implies C \implies A$  as we are proposing to do in the proof of the theorem, then you have proved equivalence of the three statements due to transitivity that automatically guarantees that all implications in the circle are reversible.
- Negation:  $\neg A$  (pronounced “not  $A$ ”) is the statement “ $A$  is false”.
- Contraposition:  $\neg A \implies \neg B$  is equivalent to  $B \implies A$ . In many cases, when trying to prove an implication, it is easier to prove the negated reverse implication or “contraposition”<sup>1</sup>. Negated implications can be confusing so here are a few examples:
  - Let  $x$  be your age.  $x \geq 21 \implies x > 18$  is clearly true and equivalent to  $\neg(x > 18) \implies \neg(x \geq 21)$  or in other words  $x \leq 18 \implies x < 21$ .
  - The statement “if you are in Cambridge, then you are in the United Kingdom” is equivalent to saying “if you are not in the United Kingdom, then you are not in Cambridge”. As it turns out, both these statements are false but would have been true if there weren’t places called Cambridge outside the United Kingdom. But the crux of the matter is that they are equivalent statements, irrespective of whether they are true or false. In other words, they are both either true or false and it’s impossible for one statement to be true and for the other one to be false.

We are now ready to complete the proof of the theorem. *Proof:*

- (i) To prove that stability implies poles inside the unit circle, we will use the contraposi-

---

<sup>1</sup>I only just discovered this term on Wikipedia and I don’t believe that it is widely known by anyone not specifically studying logic.

tion trick and hence prove that one or more poles on or outside the unit circle implies that the system is unstable.

Suppose the transfer function  $G(z) = b(z)/a(z)$  has complex poles  $p_1, p_2, \dots, p_n$  (possibly not all distinct), i.e.,

$$G(z) = \frac{b(z)}{(z - p_1)(z - p_2) \cdots (z - p_n)} \quad (4.19)$$

and assume, without loss of generality, that we have labeled the poles so that  $|p_1| \geq |p_2| \geq \dots \geq |p_n|$ . We need to consider a few cases:

- (a)  $p_1$  is a single pole with  $|p_1| > 1$ , i.e., a pole outside the unit circle. In this case, using the first step of a partial fraction expansion, we can rewrite  $G(z)$  as

$$G(z) = \frac{\alpha}{z - p_1} + \frac{b'(z)}{(z - p_2) \cdots (z - p_n)} = z^{-1} \frac{\alpha}{1 - p_1 z^{-1}} + \frac{b'(z)}{(z - p_2) \cdots (z - p_n)} \quad (4.20)$$

Then the delta response can be written as

$$g_k = \alpha p_1^{k-1} + g'_k \text{ for } k \geq 1 \quad (4.21)$$

where the terms  $g'_k$  result from the inverse  $z$  transform of the residual term of the transfer function in (4.20). Since the magnitude of the first term of  $g_k$  grows unbounded, we conclude the the delta response grows unbounded and hence a pulse input (which is bounded) gives an unbounded output, which proves that the system is unstable.

- (b) if  $p_1$  is a pole of higher multiplicity, e.g.,  $p_1 = p_2 = \dots = p_k$  for multiplicity  $k$ , then a similar argument can still be made but the partial fraction step must use partial fraction for poles with higher multiplicity, but the resulting terms associated with  $p_1$  will still grow unbounded as long as  $|p_1| > 1$ , so our previous argument still holds and the system is unstable.
- (c) now suppose  $|p_1| = 1$ , i.e., the transfer function has its largest magnitude pole on the unit circle. Assume again for now that it's a single pole. We can still perform the partial fraction expansion step as in (4.20) but now the corresponding term  $\alpha p_1^{k-1}$  in the delta response has constant magnitude and hence an input pulse no longer causes an unbounded output so this approach is no longer sufficient to prove that the system is unstable. The trick now is to find another bounded signal that results in an unbounded output. The pole  $p_1 = e^{j\theta_1}$  causes an oscillatory term

$$\alpha p_1^{k-1} = \alpha e^{j\theta_1(k-1)} \quad (4.22)$$

in the sytem's delta response. This suggests that an oscillatory input of the same frequency might create "unstable resonance" and cause an unbounded input.

Hence, we feed the system with an input that we can conveniently express in the  $z$  domain as  $X(z) = z/(z - p_1) = z/(z - e^{j\theta_1})$  to yield an output

$$Y(z) = G(z)X(z) = \frac{\alpha z}{(z - p_1)^2} + \frac{zb'(z)}{(z - p_1)(z - p_2) \cdots (z - p_n)} \quad (4.23)$$

whose inverse  $z$  transform has the form

$$y_k = \alpha k p_1^{k-1} + g'_k = \alpha k e^{j\theta_1 k} + g'_k. \quad (4.24)$$

The oscillatory component of this response now has a linear term caused by the double pole in the response, where the “doubling up” of the pole in the output has been caused by feeding a single pole delta response with an input that has the same pole. This is now an unbounded output for a bounded input  $x_k = p_1^k = e^{j\theta_1 k}$ , proving that the system is unstable.

- (d) If the pole on the unit circle has higher multiplicity, then the delta response is unbounded as it will itself have linear or higher order polynomial terms depending on the multiplicity, so an input pulse causes an unbounded output.
  - (e) We have allowed ourselves to consider a system with a possibly complex pulse response  $e^{j\theta_1(k-1)} + g'_k$  and fed it a complex input signal  $e^{j\theta_1 k}$  to create resonance in our proof. This may be seen as unsatisfactory by control theorists and some other engineers who insist on always considering real-valued signals. It is possible to make a similar (though slightly more tedious) argument for systems with real delta responses and using only real-valued input signals, e.g.,  $\cos \theta_1 k$ , to create resonance. You will be led through such a case in Examples Paper 1.
- (ii) Proving that  $|p_i| < 1$  for all poles  $p_i$  implies  $\sum_{k=0}^{\infty} |g_k|$  is finite follows similar lines as the proof above, re-writing the transfer function using a full partial fraction expansion and showing that the delta response consists of terms that all satisfy the required boundedness and hence it applies to the overall delta response. Multiple poles here don't affect the boundedness of the terms, because any linear or polynomial terms in a component are outweighed by the exponential term that, unlike the previous section where we considered poles on the unit circle that generate sequences of constant magnitude, in this case sequences will be decaying geometrically and hence faster than any polynomial. In the examples paper, you will be asked to elaborate this step of the proof, which I've merely outlined for you here.
- (iii) To prove that the fact  $\sum_{k=0}^{\infty} |g_k|$  is finite implies stability, assume a general bounded input signal  $\{x_k\}$  to the system, such that  $|x_k| \leq \beta_i$  for all  $k$ . Then the output signal

satisfies

$$|y_k| = \left| \sum_{n=0}^{\infty} g_n x_{k-n} \right| \quad (4.25)$$

$$\leq \sum_{n=0}^{\infty} |g_n| |x_{k-n}| \quad (4.26)$$

$$\leq \sum_{n=0}^{\infty} |g_n| \cdot \beta_i = \beta_i \sum_{n=0}^{\infty} |g_n|. \quad (4.27)$$

Since  $\sum_{k=0}^{\infty} |g_k|$  is finite, we've proven that there is an upper bound  $|y_k| \leq \beta_0 = \beta_i \sum_{n=0}^{\infty} |g_n|$  that holds for all output signals corresponding to input signals satisfying  $|x_k| \leq \beta_i$  for all  $k$ , proving that the system is BIBO stable.

□

Bear in mind that stability is a system property, *not* a signal property. If you feed an input signal  $X(z)$  through a stable linear system  $G(z)$ , its output signal  $Y(z)$  must not necessarily be unbounded or have its poles inside the unit circle. It's the transfer function  $G(z)$  that has to fulfil these properties. The input signal  $X(z)$  can be unbounded and cause an unbounded output signal, even for a stable system. The input signal  $X(z)$  can have poles outside the unit circle and cause the output signal  $Y(z)$  to have such poles, since any poles of  $X(z)$  are added to the list of poles of  $G(z)$  when multiplying  $G(z)X(z)$ . One interesting property of a stable system is to observe what happens when the system is fed with a unit step. Recall that the  $z$  transform of a unit step is  $z/(z-1)$ . Hence, feeding the system with a unit step is equivalent to adding a single pole at  $z=1$  to the poles in its transfer function present in the output signal  $Y(z)$ . The limit of the time-domain signal  $y_k$  can be inferred using the following theorem.

**Theorem 4.2 (Final Value Theorem (FVT))** *For a stable system  $G(z)$ , or, equivalently, for any signal  $Y(z) = \frac{z}{z-1}G(z)$  that has all its poles inside the unit circle except for a single pole at  $z=1$ , the following holds*

$$\lim_{k \rightarrow \infty} y_k = \lim_{z \rightarrow 1} (z-1)Y(z). \quad (4.28)$$

The proof of this theorem is simply the proof of the “cover up” rule we provided in our partial fractions primer: if you consider the partial fractions expansion of  $Y(z)$

$$\lim_{z \rightarrow 1} (z-1)Y(z) = \lim_{z \rightarrow 1} \frac{(z-1)zG(z)}{(z-1)} = \lim_{z \rightarrow 1} \left[ (z-1) \frac{\alpha_0 z}{z-1} + (z-1) \sum_{k=1}^n \frac{\alpha_k z}{z-p_k} \right], \quad (4.29)$$

it shows that the limit is  $G(1) = \alpha_0$ . Since  $y_k$  is a sum of geometric sequences that tend to zero except for the term emanating from  $\alpha_0/(z-1)$  in the partial fractions expansion, it tends to  $\alpha_0$ .

## 4.3 Stationary response to sinusoidal inputs and Bode diagrams

As discussed earlier, the difference equation associated with stable systems has a complementary solution (or transient) that tends to zero and hence an overall solution that tends towards its particular solution (or stationary / steady-state solution). In particular, similarly to continuous systems, when the input signal is oscillatory (of the form  $Ae^{j(\theta k + \Phi)}$  or  $A \cos(\theta k + \Phi)$ ) then the stationary output is also oscillatory with the same frequency, but with an adjusted magnitude and phase.

In this section, we will first prove what we've just stated, namely that oscillatory inputs give oscillatory outputs of the same frequency. We will then discuss Bode diagrams and the operational meaning of the spectrum of the transfer function of a linear system.

### 4.3.1 Stationary output of stable systems for oscillatory inputs

Consider a system with a stable transfer function

$$G(z) = \frac{b(z)}{(z - p_1)(z - p_2) \cdots (z - p_n)} \quad (4.30)$$

with its poles  $p_1, p_2, \dots, p_n$  strictly inside the unit circle. If we input a complex sinusoidal signal  $\{x_k\}_{k \geq 0} = \{Ae^{j(\theta k + \Phi)}\}$  into the system, its output is

$$Y(z) = G(z)X(z) = \frac{b(z)}{(z - p_1) \cdots (z - p_n)} \cdot \frac{Ae^{j\Phi}z}{z - e^{j\theta}}. \quad (4.31)$$

Using partial fractions, we can re-write this as

$$Y(z) = \frac{G(e^{j\theta})Ae^{j\Phi}z}{z - e^{j\theta}} + \sum_{k=1}^n \frac{\alpha_k z}{z - p_k} \quad (4.32)$$

As previously, we have assumed single poles in the expansion in order to keep things tidy, but you can easily persuade yourself that everything we derive here would work equally well if you had poles with higher multiplicity. The inverse transform of the output is of the form

$$y_k = Ae^{j\Phi}G(e^{j\theta})e^{j\theta k} + g'_k \quad (4.33)$$

where  $g'_k$  are the terms emanating from the inverse transform of the residual term on the right of (4.32): since the poles of this expression are all inside the unit circle, these terms decay as  $k$  goes to  $\infty$ . Hence the stationary output signal is

$$\tilde{y}_k = Ae^{j\Phi}G(e^{j\theta})e^{j\theta k} = A|G(e^{j\theta})|e^{j(\theta k + \Phi + \angle G(e^{j\theta}))}. \quad (4.34)$$

The output is a complex sinusoidal of the same frequency as the input, scaled by  $|G(e^{j\theta})|$  and phase-shifted by  $\angle G(e^{j\theta})$ . This argument for a complex input sinusoidal can be extended to real sinusoids, for example a cosine, by writing  $A \cos(\theta k + \Phi) = \frac{A}{2}(e^{j(\theta k + \Phi)} +$

$e^{-j(\theta k + \Phi)}$ ) and using the superposition principle, to show that the stationary output is of the form

$$\tilde{y}_k = A|G(e^{j\theta})| \cos(\theta k + \Phi + \angle G(e^{j\theta})) \quad (4.35)$$

This suggests that the spectrum of DTFT  $G(\theta)$  of a discrete-time system fully determines the stationary response of a linear system to oscillatory inputs.

Note that the stationary response of a stable system to an oscillatory input only depends on the transfer function evaluated on the unit circle and hence on the DTFT of the system's delta response. We can write its spectrum alternatively as  $G(e^{j\theta})$  or  $G(\theta)$  depending on whether we use the  $z$  transform or the DTFT.

*Warning:* the derivation above applies to non-stable systems as well but, unlike for stable systems, the terms  $g'_k$  do not necessarily decay as  $k$  goes to  $\infty$ , hence the stationary output may be drowned out by a growing transient and become insignificant in comparison. Furthermore, it should be noted that the DTFT sum may not converge for a non-stable system, and the  $z$  transform is only defined by virtue of the analytic continuation theorem that we touched upon in Section 3.2.1.

### 4.3.2 Bode diagram

The Bode diagram plots the magnitude and phase of the spectrum  $G(\theta)$  of the delta response of a stable system as a function of the normalised frequency  $\theta$ . By convention, the magnitude is plotted in dB. An example Bode diagram for the transfer function

$$G(z) = \frac{\sqrt{181}/100}{z^2 - 0.9 \times 2 \times \cos(\pi/4)z + 0.9^2} \quad (4.36)$$

is given in Figure 4.3. Note a few details about this diagram:

- the  $y$  axis on the magnitude plot is effectively logarithmic because the unit is dB.
- the  $y$  axis on the phase plot is in rad. When plotting the phase, one can either restrict the  $y$  axis to the interval  $[-\pi, \pi]$  but the example plot would have a jump from  $-\pi$  to  $+\pi$  at  $\theta = \pi/4$  if one did this. To plot a phase over an extended phase domain avoiding such jumps, use the command `numpy.unwrap()` in Python or simply `unwrap()` in MATLAB.
- the  $x$  axis for both plots is linear from 0 to  $\pi$ , which is the significant domain of the normalised frequency for systems with a real delta response. Should you ever need to plot a Bode diagram for a system with a complex delta response, you would need to expand these plots either to the interval  $[-\pi, \pi]$  or to the interval  $[0, 2\pi]$ . The former is probably preferable. As we will see when studying the Discrete Fourier Transform (DFT) later in this course, a DFT spectrum always comes up as a two-sided discrete spectrum effectively from 0 to just short of  $2\pi$ . The functions `numpy.fft.fftshift()` in Python or just `fftshift()` in MATLAB rotate such a spectrum so as to put the 0 frequency in the middle, effectively changing the domain to discrete frequency points from  $-\pi$  to just short of  $\pi$ .

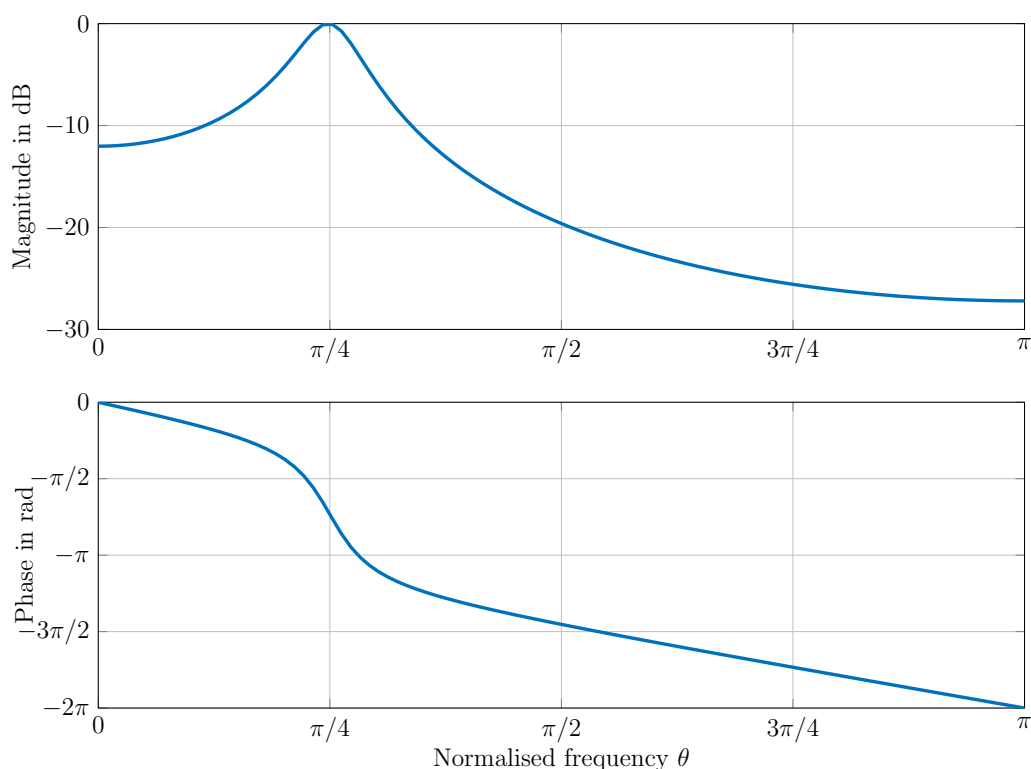


Figure 4.3: an example Bode diagram

- you may be surprised that we plotted Bode diagrams against a linear frequency axis. Bode diagrams in continuous systems are typically double-logarithmic, where logarithmic magnitude in dB is plotted against the logarithm of the frequency  $\omega$ . In discrete time systems, it's really a matter of preference and depends on the type of application you are considering. Frequencies are naturally bounded to half the sampling frequency and don't go to infinity as in continuous systems. If the system you are considering has its salient features well spread over the finite frequency axis (as is the case for the Bode diagram in Figure 4.3, where the resonance frequency is at  $\pi/4$ ), then it makes sense to display its Bode diagrams against a linear frequency axis. If on the other hand you are using an extremely high sampling frequency, using discrete time signal processing effectively in an “almost continuous time” regime, i.e., if the salient features of your spectrum occur at tiny fractions of  $\pi$ , then it still makes sense to plot the Bode diagram against a logarithmic frequency. Both are acceptable and there is no right or wrong.
- in general, when plotting a Bode diagram, you are interested in actual frequencies rather than normalised frequencies. Knowing that your signal or system has a notch at 10 kHz says a lot more to most engineers than knowing that it occurs at a normalised frequency of  $\theta = \pi/4$ . Figure 4.4 re-draws the Bode diagram of Figure 4.3 against the frequency in kHz for a sampling frequency of  $f_s = 40$  kHz.

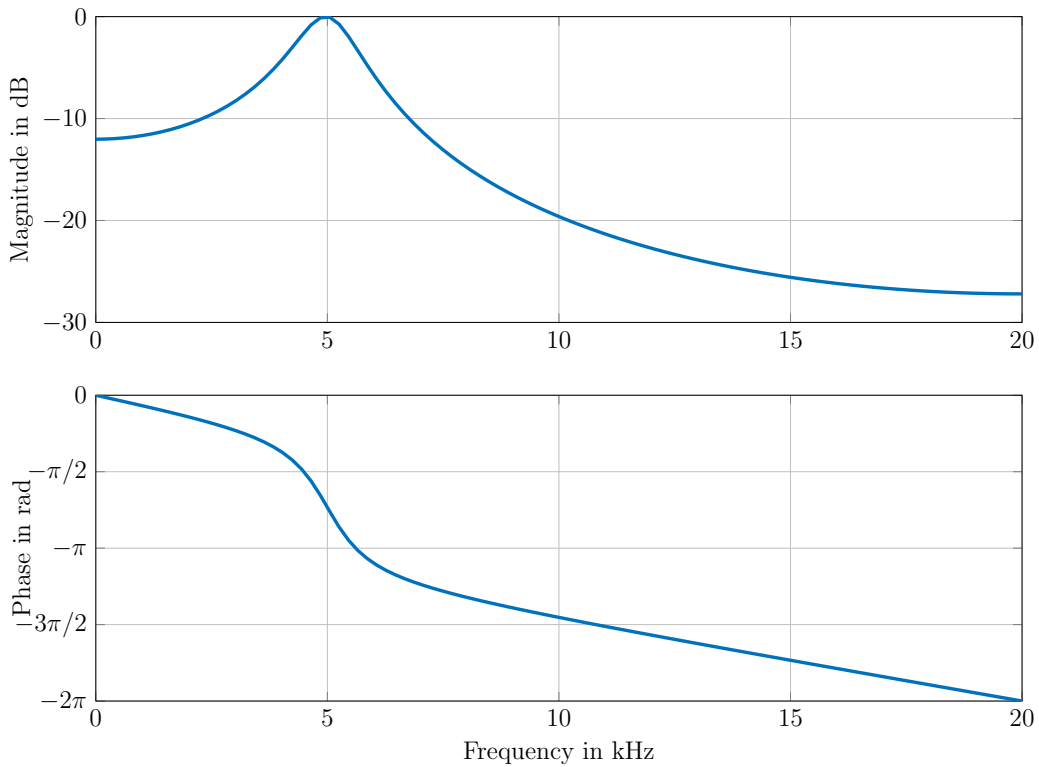


Figure 4.4: an example Bode diagram

We must warn again against inadvertently using Bode diagrams for non-stable systems. The system whose Bode diagram we plotted in Figure 4.3 has its poles at  $0.9e^{j\pi/4}$  and  $0.9e^{-j\pi/4}$ , i.e., inside the unit circle and hence it is a stable system, as per the rules we derived in Theorem 4.1. If we had instead considered a transfer function

$$G(z) = \frac{\sqrt{222}/100}{z^2 - 1.1 \times 2 \times \cos(\pi/4)z + 1.1^2} \quad (4.37)$$

with poles at  $1.1e^{j\pi/4}$  and  $1.1e^{-j\pi/4}$ , i.e., just outside the unit circle, we would have plotted a Bode diagram almost identical to the one we plotted in Figure 4.3, but this Bode diagram would have been useless in determining the output for a given input signal because the transient of such a system would grow to be far larger than its stationary output. The reason we are warning you again is that we have not learned methods to determine stability from a Bode diagram, and hence one has to be careful before plotting such a diagram to check whether the system is stable.

For a stable system with transfer function  $G(z)$  with input signal  $\{x_k\}_{k \geq 0}$  with  $z$  transform  $X(z)$ , the convolution property dictates that the output is  $Y(z) = G(z)X(z)$  and hence also on the unit circle  $Y(e^{j\theta}) = G(e^{j\theta})X(e^{j\theta})$  or alternatively directly using the convolution property of the DTFT,  $Y(\theta) = G(\theta)X(\theta)$ . Hence we see that the Bode diagram

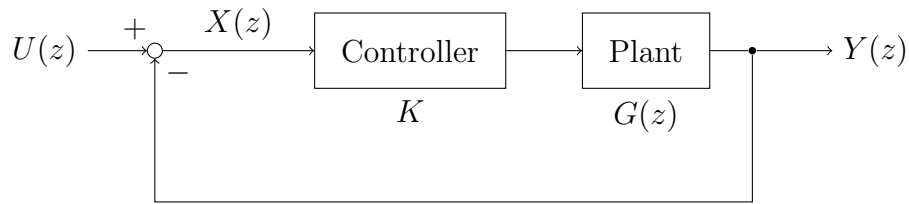


Figure 4.5: A closed loop control system

gives us an additive mask applied to an input signal as

$$20 \log_{10} |G(\theta)| + 20 \log_{10} |X(\theta)| \longrightarrow 20 \log_{10} |Y(\theta)| \quad \text{and,} \quad (4.38)$$

$$\angle G(\theta) + \angle X(\theta) \longrightarrow \angle Y(\theta) \quad (4.39)$$

So for the Bode diagrams in Figure 4.4, the input signal would retain its frequency components around 5 kHz, and incur an attenuation of all other frequency components, where the attenuation is at least 10 dB at frequencies outside the range 2.5 kHz to 7.5 kHz.

## 4.4 Feedback, closed loop control systems and Nyquist stability criterion

In this section, we will consider closed loop control systems as exemplified in Figure 4.5.