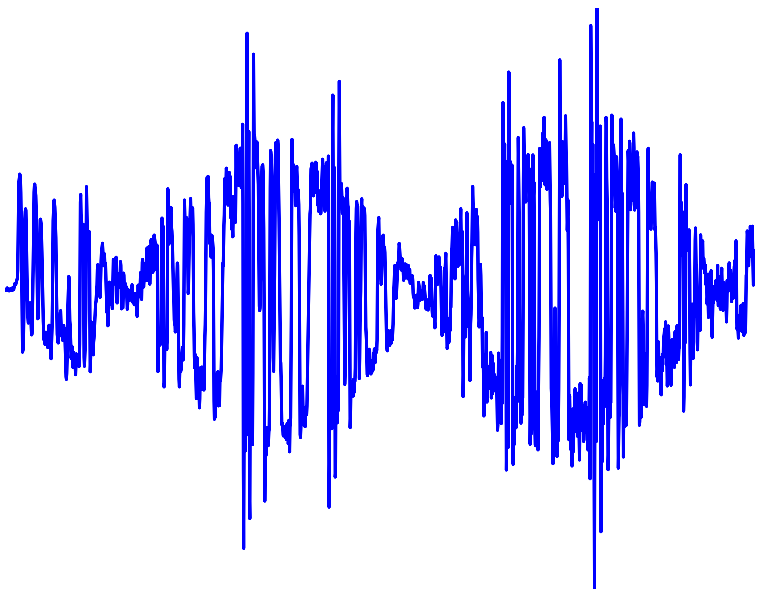
DR. MICHAEL P. HAYES

ENMT301
SIGNAL PROCESSING FOR MECHATRONICS DESIGN
LECTURE NOTES

V51



ELECTRICAL AND COMPUTER ENGINEERING

Contents

Glossary and definitions

1 Notation

x vector

M matrix

$x(t)$ continuous-time signal[1]

$X(f)$ Fourier transform of continuous-time signal, $x(t)$

$X_f(f)$ Fourier transform of continuous-time signal, $x(t)$, when there is possible ambiguity with the Laplace transform, $X(s)$, or z-transform, $X(z)$

$x[n]$ discrete-time signal (equivalent to the sequence $\{x_n\}$)

x_n n^{th} sample of a discrete-time signal

$X[k]$ discrete Fourier transform of $x[n]$

$X_{\frac{1}{\Delta t}}(f)$ discrete-time Fourier transform of $x[n]$ with sampling period Δt

$X(z)$ z-transform of discrete-time signal $x[n]$

$X(s)$ Laplace transform of continuous signal $x(t)$

$|X(f)|$ magnitude of Fourier transform of continuous signal

$\angle X(f)$ phase of Fourier transform of continuous signal

$X^*(f)$ complex conjugate of spectrum $X(f)$

$\{\underline{1}, 2, 0, 3\}$ discrete-time sequence. The underscore marks the time origin. The other samples are zero. This sequence has a length (extent) of 4.

$\{X_n\}$ discrete random process

$\{x_n\}$ single realisation (sample function) of a random process $\{X_n\}$

X_n n^{th} random variable of discrete random process

$f_X(x)$ probability distribution function of random variable X

μ_X, σ_X mean and standard deviation of random variable X

[1] As with any function notation, there is ambiguity over whether $x(t)$ denotes the entire signal or a value at a specific time, t. This needs to be resolved from the context.

2 Glossary

a amplitude

c speed of propagation (m/s)

D transducer dimension (m)

$e(t)$ sonar echo signal

f frequency (Hz)

f_0 centre frequency (Hz)

$H(t)$ Heaviside's unit step

q absorption coefficient

r range (m)

R_u unambiguous range (m)

$s(t)$ sonar transmit signal

t continuous time (s)

T period; sonar repetition period (s)

T_p pulse duration (s)

$\delta(t)$ Dirac delta

$\delta[n]$ unit impulse

λ wavelength (m)

τ time delay (s)

$\mathrm{rect}(t)$ rectangle function

$\mathrm{sinc}(t)$ cardinal sine function

$h[n]$ impulse response

$x[n]$ input to a filter

$u[n]$ discrete Heaviside step

$w[n]$ white Gaussian noise signal

$y[n]$ output from a filter

3 Definitions

Forward Fourier transform:

$$S(f) = \int_{-\infty}^{\infty} s(t) \exp(-\mathrm{j}2\pi f t) \, \mathrm{d}t. \tag{1}$$

Inverse Fourier transform:

$$s(t) = \int_{-\infty}^{\infty} S(f) \exp(\mathrm{j}2\pi f t) \, \mathrm{d}f. \tag{2}$$

Forward z-transform:

$$H(z) = \sum_{n=-\infty}^{\infty} h[n] z^{-n}. \tag{3}$$

Discrete convolution:

$$y[n] = \sum_{m=-\infty}^{\infty} h[m] x[n-m]. \tag{4}$$

If $h[m]$ causal and of finite extent M,

$$y[n] = \sum_{m=0}^{M-1} h[m] x[n-m]. \tag{5}$$

Discrete-time Fourier transform:

$$X_{\frac{1}{\Delta t}}(f) = \sum_{n=-\infty}^{\infty} h[n] \exp(-\mathrm{j}2\pi f n \Delta t). \tag{6}$$

Discrete Fourier transform:

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} h[n] \exp\left(-\mathrm{j}2\pi \frac{nk}{N}\right). \tag{7}$$

1
Sonar I

Sonar[1] is an acronym for sound navigation and ranging. It operates by transmitting a pulse of acoustic energy[2] and then timing how long it takes for an echo to be received.

[1] Radar is similar to sonar but uses much higher frequencies since the speed of propagation is so much greater.
[2] Usually ultrasonic.

1 Pulse echo operation

Pulse echo sonars require transducers for converting a voltage signal to a pressure signal and vice-versa.

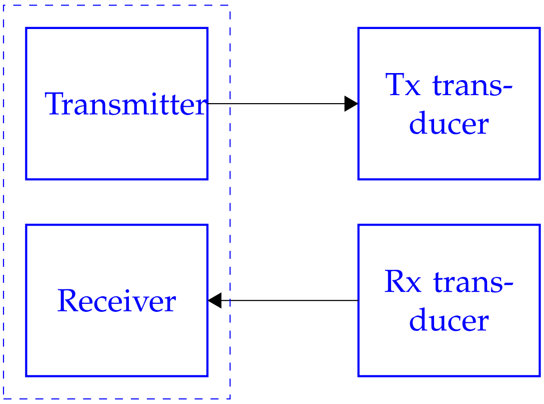


Figure 1.1: Sonar with separate transmitting and receiving transducers (antennas). Sometimes a single transducer is used with a transmit/receive switch.

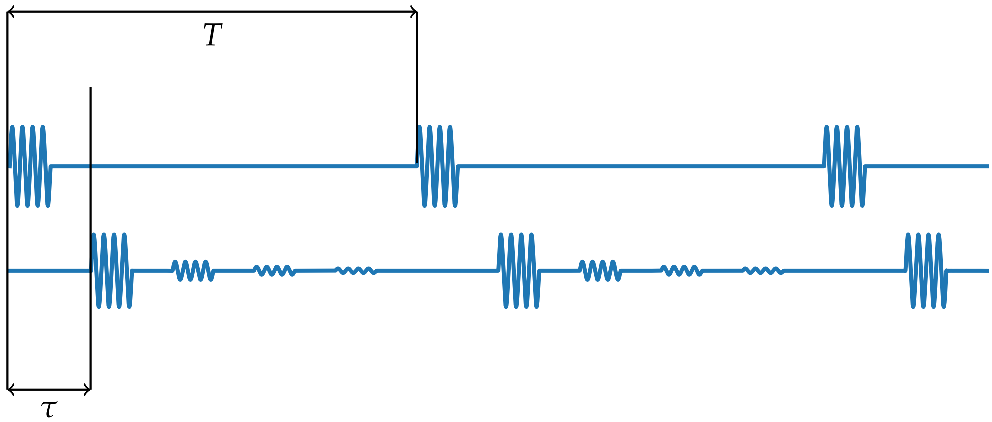


Figure 1.2: Periodic toneburst pulse with a repetition period T and its echoes. The first echo occurs at a delay τ.

2 *HC-SR04 air sonar*

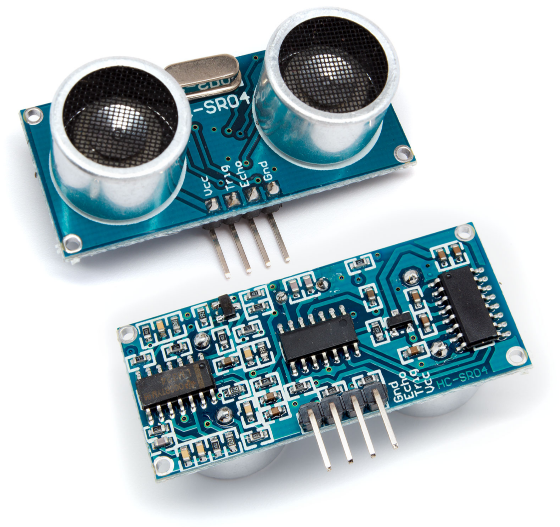


Figure 1.3: Photo of the HC-SR04 air sonar showing both sides of the PCB. The chips from left to right are: quad opamp, MCU, quad inverter with charge-pumps producing ± 10 V from a 5 V supply.

2.1 *Interfacing an HC-SR04 to a microcontroller*

The HC-SR04 requires four connections:

+5V to provide power

0V for a ground reference

TRIG driven from a PIO to trigger a transmit pulse

ECHO input to a PIO for timing of the echo delay

When the HC-SR04 sees the TRIG signal pulsed high for 10 μs, it drives ECHO low and generates a 40 kHz pulse train. The received echoes are compared with a threshold by a comparator. When the HC-SR04 detects the comparator output going high it drives the ECHO signal high.

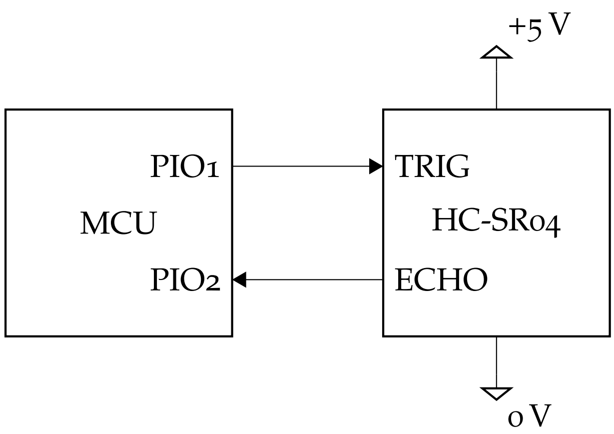


Figure 1.4: HC-SR04 interfaced to an Arduino.

2.2 HC-SR04 electronics

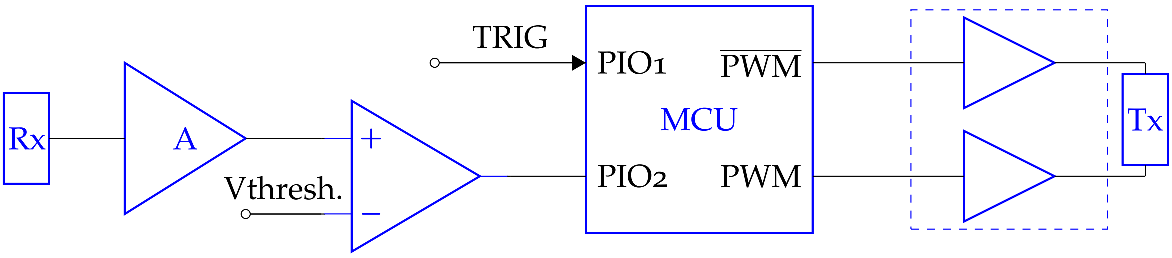


Figure 1.5: HC-SR04 air sonar block diagram.

The HC-SR04 air sonar is shown in Figure 1.3. It uses a microcontroller to generate the transmit pulse. A block diagram of its operation is shown in Figure 1.5. Points to note:

1. Complementary PWM outputs from the MCU are used to double the voltage swing for the transmit transducer; this quadruples the output power[3].
2. Power electronics[4] within the transducer driver chip convert the MCU 5 V power supply to ±10 V.
3. Signal conditioning electronics amplifies and bandpass filters the received echo.
4. The amplified echo signal is compared with a threshold voltage using a comparator. The output of the comparator indicates to the MCU when a sufficiently large echo is detected.
5. The comparator threshold voltage is adjusted to compensate for the spreading loss from targets at long range[5].

[3] Since the voltage swing is doubled.
[4] Charge pumps.

[5] The spreading loss could also have been compensated by varying the amplifier gain.

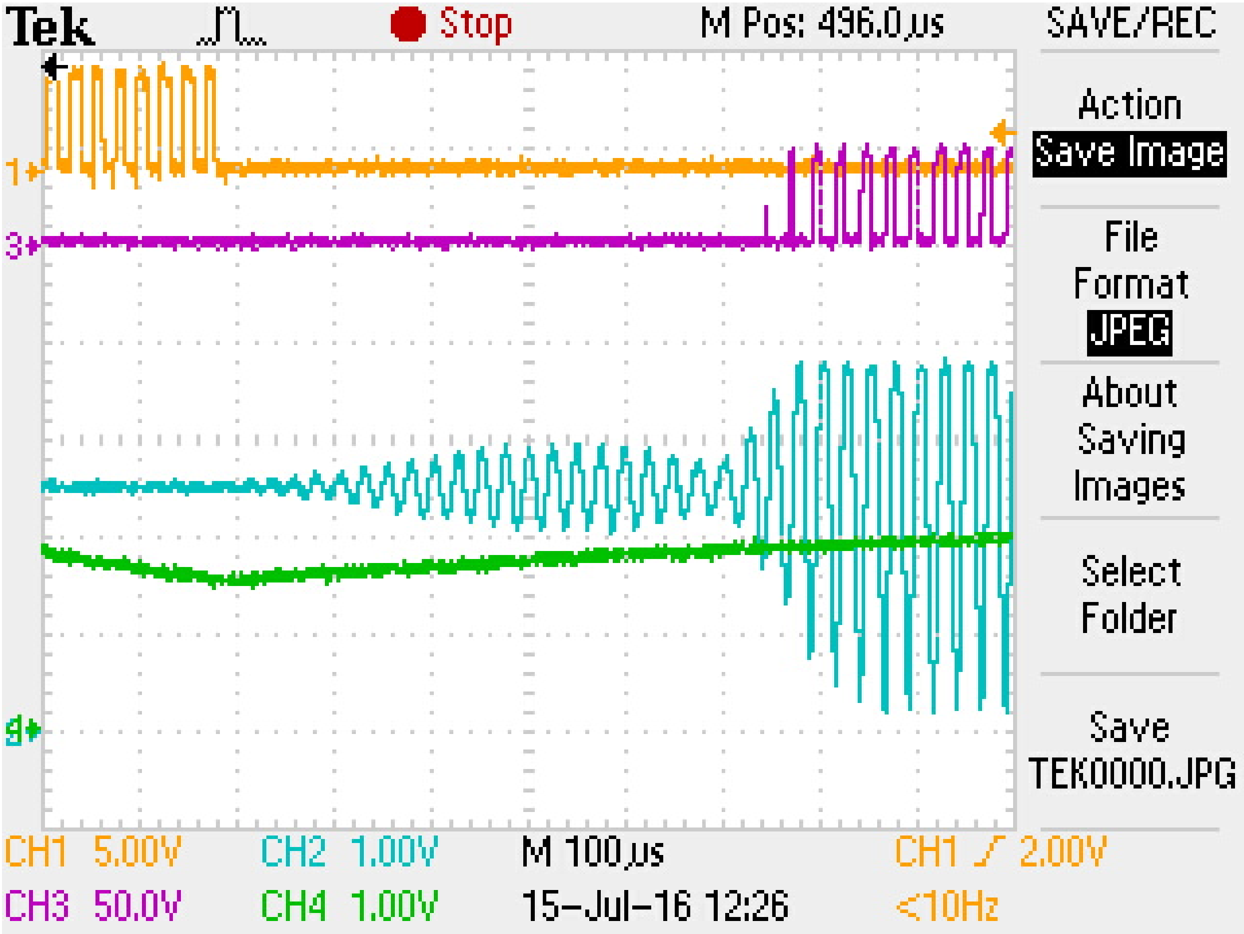


Figure 1.6: HC-SR04 waveforms. The orange trace is the transit signal generated by the MCU, the blue trace is the amplified and filtered echo, the green trace is the threshold, and the purple trace is the output of the comparator. Note, the CH3 probe setting is wrong.

2.3 *Waveforms*

Example signals measured by an oscilloscope from the HC-SR04 are shown in Figure 1.6. There are a number of features to note:

1. The transmit signal is a square wave generated by the MCU (say using PWM).
2. The echo signal is no longer a square wave[6].
3. The threshold signal is not constant[7].
4. The comparator outputs a pulse when the echo signal falls below the threshold.
5. The echo signal has a smaller component prior to the reflection from the target[8].
6. The signals are noisy.

The time delay, τ, is related to the target range, r, by

$$\tau = \frac{2r}{c},$$

where c is the speed of sound propagation.

[6] It is filtered by the transducers and signal conditioning electronics.

[7] This is to compensate for spreading losses.

[8] This is due to acoustic crosstalk—some of the transmitted energy travels directly to the receiver.

2.4 *Example HC-SR04 Arduino code*

Interfacing the HC-SR04 to a MCU is straightforward. It requires +5 V and ground connection, an output PIO signal to trigger a ping, and a PIO input to measure the time until an echo is detected.

```
// For use with HC-SR04 US sensor

const unsigned int echo_pin = 28;
const unsigned int trig_pin = 26;
const unsigned int repetition_period_ms = 60;
const unsigned int echo_wait_us = 10000;
const unsigned long serial_baud_rate = 115200;

void setup()
{
    pinMode(trig_pin, OUTPUT);
    pinMode(echo_pin, INPUT);
    Serial.begin(serial_baud_rate);
}

void loop()
{
    static unsigned long time_ms_previous = 0;
    unsigned long time_ms;
    unsigned long duration_us;

    time_ms = millis();
    if ((time_ms - time_ms_previous) < repetition_period_ms)
        return;
    time_ms_previous = time_ms;

    // Trigger ping
    digitalWrite(trig_pin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig_pin, LOW);

    // Wait for detected echo or timeout if too long
    duration_us = pulseIn(echo_pin, HIGH, echo_wait_us);

    Serial.print(duration_us);
    Serial.print("\n");
}
```

Listing 1.1: Example Arduino code to drive the HC-SR04 and output the measured delay in microseconds.

3 *Exercises*

1. Describe the purpose of range varying gain amplifier in a time-of-flight sonar system?
2. What is the advantage of using a range varying threshold in a time-of-flight sonar system?
3. What is acoustic crosstalk?
4. Why are the echo signals noisy?
5. If the pulse repetition period is 10 ms what is the maximum unambiguous range?
6. If the maximum unambiguous range needs to be 5 m, what is the minimum pulse repetition period?

2
Signals introduction

The output of sensors are signals. So are the inputs to actuators. Control systems manipulate signals. Signal processing[1] is important for many things, including:

- Communications
- Control systems
- Econometrics
- Imaging systems
- Robotics

[1] Some signal processing applications in a smart-phone include: audio filtering, speech enhancement, image enhancement, image transformation, bluetooth/WiFi/cellular modem modulation and filtering, sensor fusion (IMU, GPS).

1 Signal classification

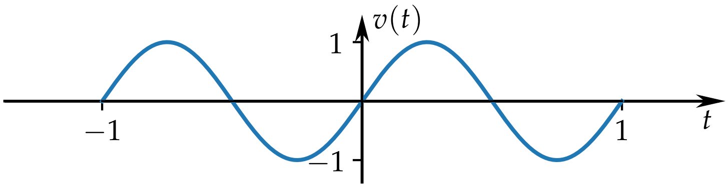
Signals are usually single-valued functions of time[2]. Images and movies can be considered multidimensional signals. The two most common signals are:

[2] Or space in the case of image processing.

Analogue signals These are functions of continuous time with continuous values[3]. For example,

$$v(t) = \sin(2\pi t)\,.$$

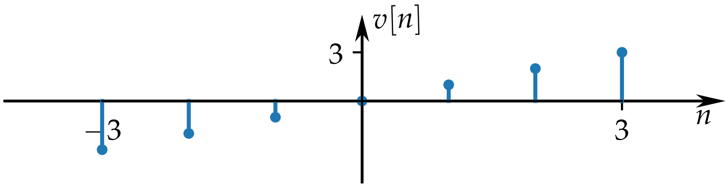
[3] Analogue signals may have discontinuities, for example a square wave, but these are not realisable; they can only be approximated.



Digital signals These are a sequence of integers, e.g.,

$$v[n] = \{-3, -2, -1, \underline{0}, 1, 2, 3\}\,.$$

These are discrete-time/discrete-value signals. They can be shown with a lollipop plot.



1.1 *Four types of signal*

Most signal processing is performed with digital signals but the mathematics is simpler for analogue signals. There are four classifications of signal in terms of continuous/discrete time and continuous/discrete value as shown in Figure 2.1. The discrete-time continuous-amplitude signal is a useful bridge between analogue and digital signals.

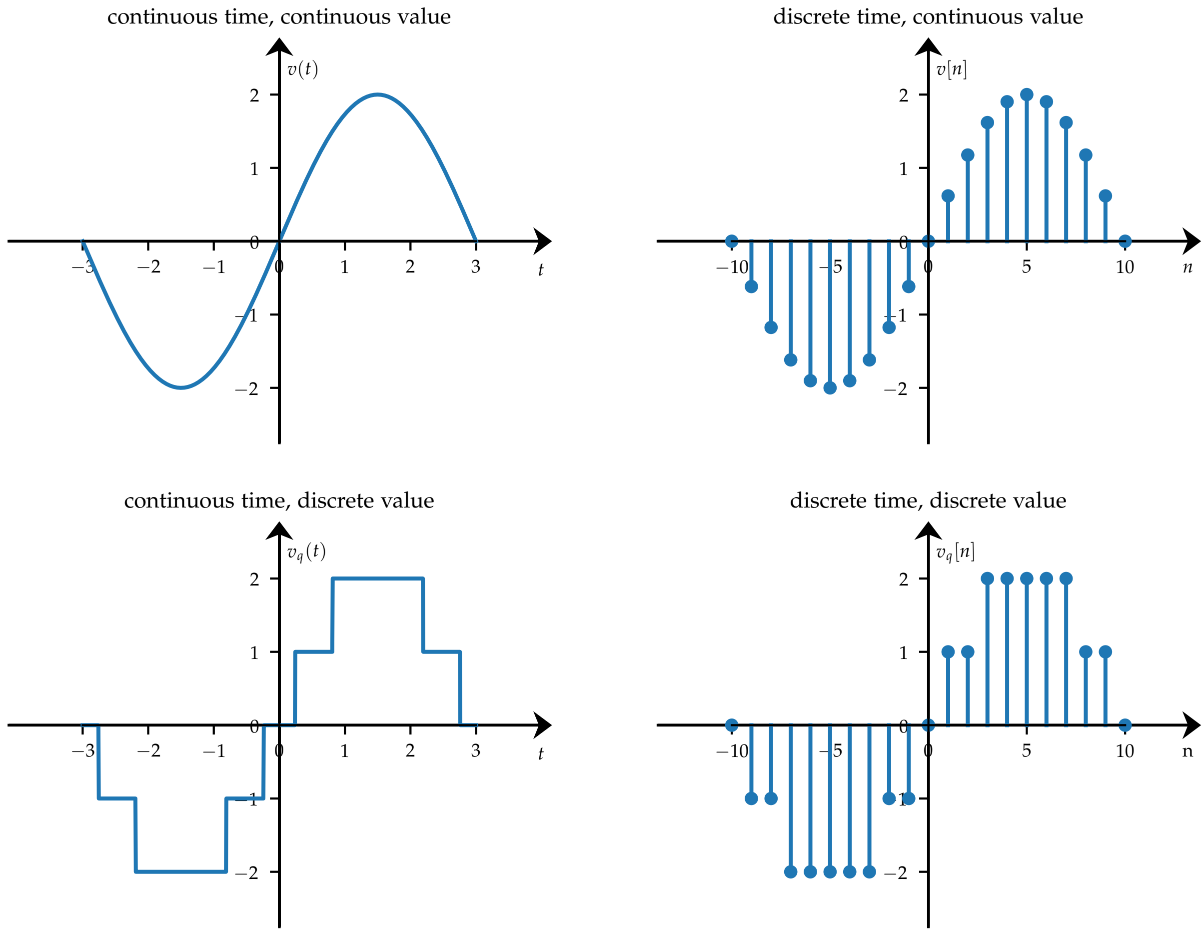


Figure 2.1: Comparison of signals in terms of continuous/discrete time and continuous/discrete value.

$$
\begin{aligned}
t &\in \mathcal{R} \quad \text{continuous time}\\
n &\in \mathcal{I} \quad \text{discrete time}\\
v(t) &\in \mathcal{R} \quad \text{continuous value}\\
v_q(t) &\in \mathcal{I} \quad \text{discrete value}
\end{aligned}
$$

2 *Continuous time signals*

Continuous time signals are denoted with a lowercase letter and t as an argument[4]. For example, $v(t)$ can denote the voltage signal from an analogue sensor.

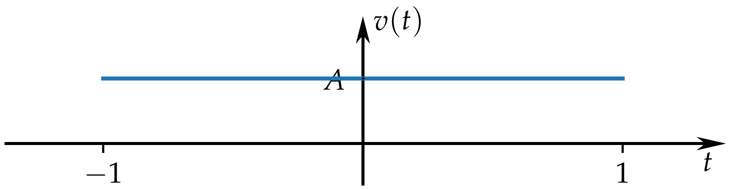
[4] There can be confusion between the notation for the entire signal or the value at a particular time. The latter is denoted $v(0)$ or sometimes $v(t_0)$.

2.1 *Common continuous time signals*

Continuous time signals are usually parameterised. For example:

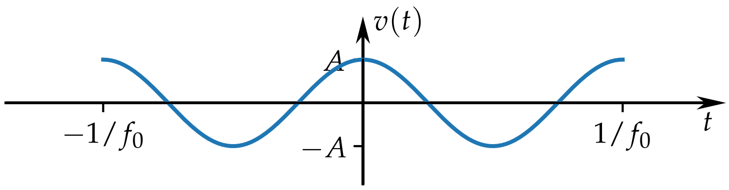
DC

$$v(t) = A \tag{2.1}$$



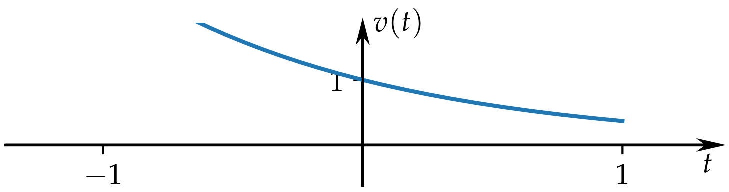
AC

$$v(t) = A\cos\left(2\pi f_0 t + \phi\right). \tag{2.2}$$



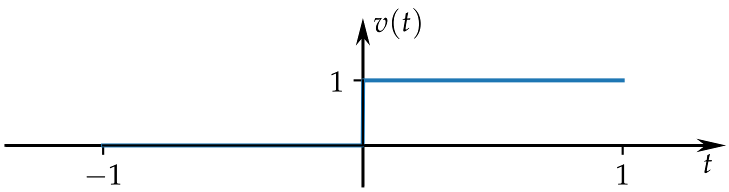
Exponential

$$v(t) = \exp\left(-\alpha t\right). \tag{2.3}$$



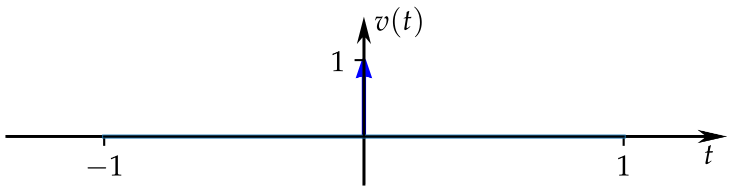
Heaviside unit step

$$v(t) = H(t). \tag{2.4}$$



Dirac delta

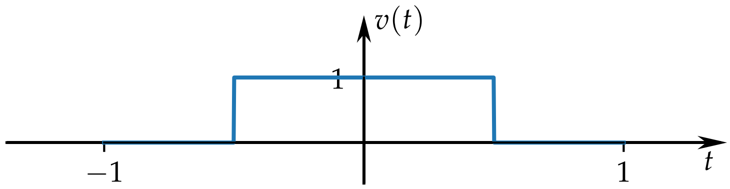
$$v(t) = \delta(t) = \frac{\mathrm{d}}{\mathrm{d}t} H(t). \tag{2.5}$$



The Dirac delta has infinite height, zero width, and unit area so it is drawn as an arrow.

Rectangle

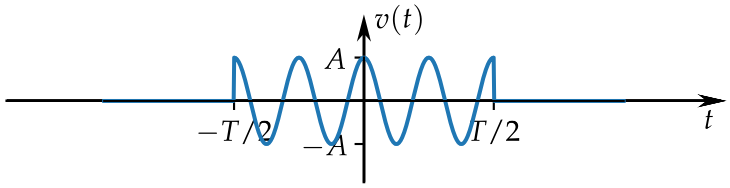
$$v(t) = \mathrm{rect}(t) = H\left(t + \frac{1}{2}\right) - H\left(t - \frac{1}{2}\right). \tag{2.6}$$



The rectangle function is not physically realisable since it is discontinuous. In practice, there will be a finite rise and fall time.

Tone burst

$$v(t) = A\,\mathrm{rect}\left(\frac{t}{T}\right) \cos\left(2\pi f_0 t + \phi\right). \tag{2.7}$$



This can describe a radar or sonar pulse of amplitude A, duration T, frequency f_0, and phase ϕ.

3 Discrete-time signals

A discrete-time voltage signal is denoted $v[n]$. Here n is the time index. It is related to continuous time by

$$t = n\Delta t, \tag{2.8}$$

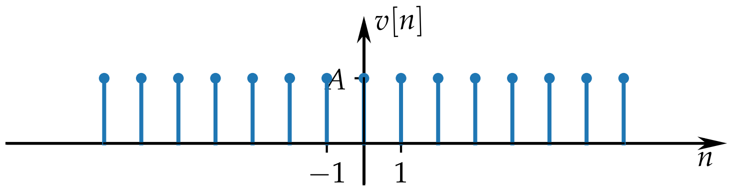
where Δt is the sampling period.

3.1 Common discrete-time signals

Discrete-time signals can be sampled versions of continuous time signals. For example:

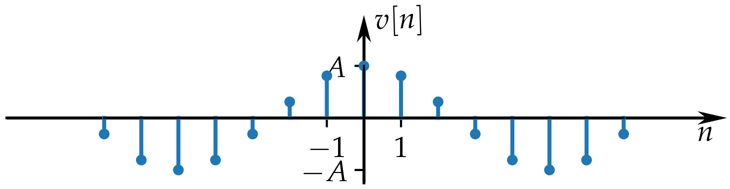
DC

$$v[n] = A \tag{2.9}$$



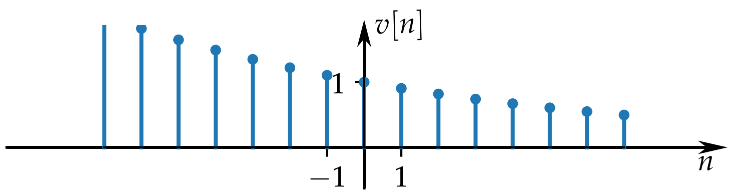
AC

$$v[n] = A\cos\left(2\pi f_0 n\Delta t + \phi\right). \tag{2.10}$$



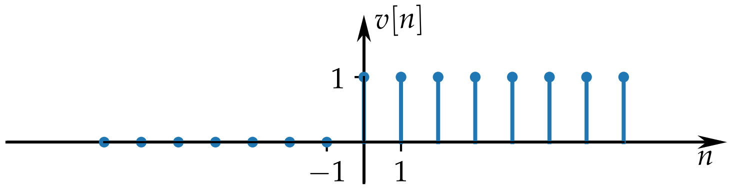
Exponential

$$v[n] = \exp\left(-\alpha n\Delta t\right) = a^n \tag{2.11}$$



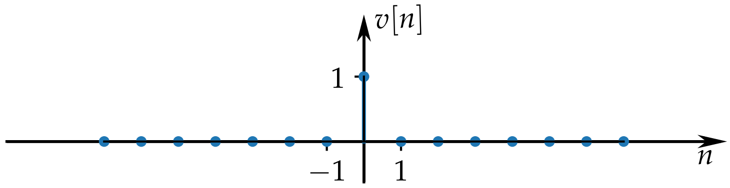
Unit step

$$v[n] = u[n] = \begin{cases} 1 & n \geq 0, \\ 0 & n < 0 \end{cases} = \{\cdots, 0, 0, \underline{1}, 1, 1, \cdots\}. \tag{2.12}$$



Unit impulse (Kronecker delta)

$$v[n] = \delta[n] = \begin{cases} 1 & n = 0, \\ 0 & n \neq 0 \end{cases} = \{\cdots, 0, 0, \underline{1}, 0, 0, \cdots\}. \tag{2.13}$$



3.2 *Discrete-time signal representation*

There are many ways to represent a discrete-time signal. For example, a unit step can be represented as the sequence:

$$H[n] = \{\cdots, 0, 0, \underline{1}, 1, 1, 1, \cdots\}. \tag{2.14}$$

Here the underline[5] denotes the sample where $n = 0$.

Here's another example,

$$v[n] = \{\cdots, 0, 0, \underline{1}, 2, 3, 0, 0, \cdots\}. \tag{2.15}$$

This sequence can be represented as the sum of scaled unit impulses,

$$v[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2]. \tag{2.16}$$

This can be generalised as

$$v[n] = \sum_{m=0}^{2} (n+1)\delta[n-m]. \tag{2.17}$$

[5] Sometimes an arrow is used to denote the zero index.

4 *Complex signals*

The signal produced from an analogue sensor is a set of real numbers. However, the mathematics can be simplified by using complex signals[6]. For example, using Euler's theorem, an AC signal can be represented by the summation of two complex exponential signals,

$$\cos(2\pi f_0 t) = \frac{1}{2}\exp(\mathrm{j}2\pi f_0 t) + \frac{1}{2}\exp(-\mathrm{j}2\pi f_0 t). \tag{2.18}$$

[6] These are a mathematical contrivance—you cannot measure a complex signal.

5 Exercises

1. Match the following equations to the waveforms (where possible).[7].

[7] Hints: The underscore marks the sample at $n = 0$. Euler's formula is $\exp(\mathrm{j}\theta) = \cos(\theta) + \mathrm{j}\sin(\theta)$.

(a) $\cos(4\pi t)\,\mathrm{rect}\left(\frac{t}{2}\right)$

(b) $H(t - 0.5)$

(c) $H(t + 0.5)$

(d) $\mathrm{rect}(t)$

(e) $\mathrm{rect}(t - 2)$

(f) $\mathrm{rect}\left(\frac{t}{4}\right)$

(g) $H(t + 1) - H(t - 1)$

(h) $\cos(2\pi t)\,\mathrm{rect}\left(\frac{t}{2}\right)$

(i) $\{1, 0, 1, \underline{2}, 1, 3\}$

(j) $\{1, 0, \underline{1}, 2, 1, 3\}$

(k) $\mathrm{rect}\left(\frac{t+1}{4}\right)$

(l) $\mathrm{rect}\left(\frac{t-1}{4}\right)$

(m) $\delta[n + 4] + \delta[n + 2] + 2\delta[n + 1] + \delta[n] + 3\delta[n - 1]$

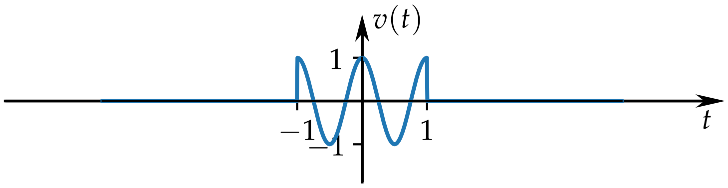
(n) $\mathrm{rect}\left(\frac{t}{3}\right)$

(o) $\cos(2\pi t)\,\mathrm{rect}(t/4)$

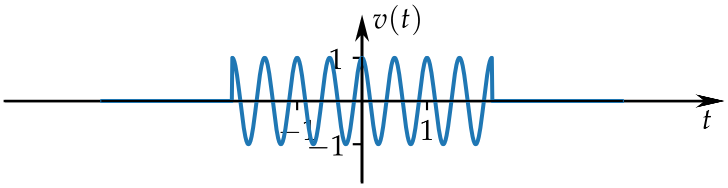
(p) $0.5\left[\exp(\mathrm{j}4\pi t) + \exp(-\mathrm{j}4\pi t)\right]\mathrm{rect}(t/4)$

(q) $\sin(4\pi t)\,\mathrm{rect}(t/4)$

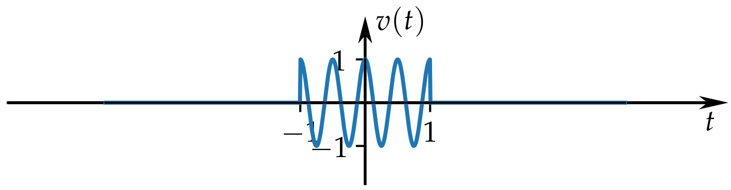
(r) $-0.5\left[\mathrm{j}\exp(\mathrm{j}2\pi t) - \mathrm{j}\exp(-\mathrm{j}2\pi t)\right]\mathrm{rect}(t/4)$



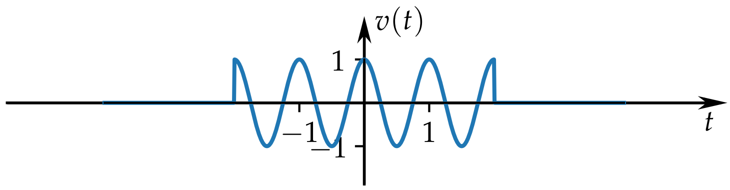
(A)



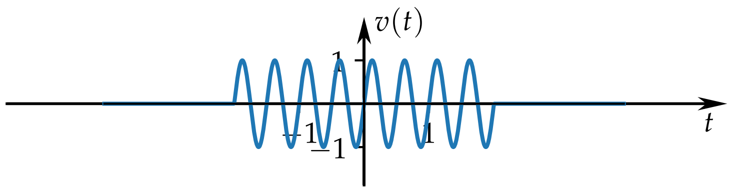
(B)


$v(t)$
1
-1
-1
1
t

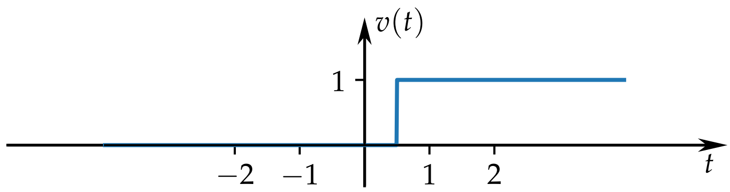
(C)


$v(t)$
1
-1
-1
1
t

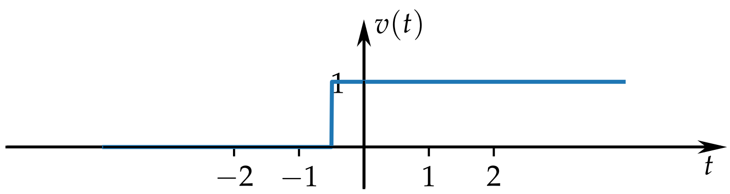
(D)


$v(t)$
1
-1
-1
1
t

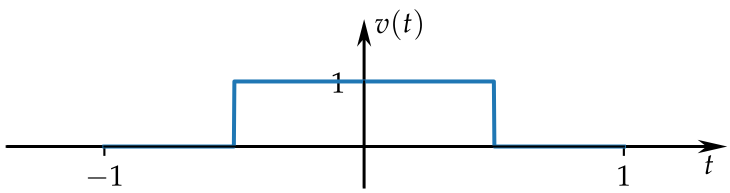
(E)


$v(t)$
1
-2
-1
1
2
t

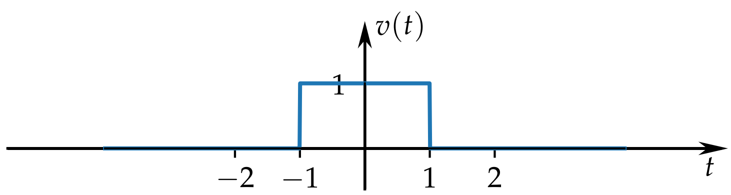
(F)


$v(t)$
1
-2
-1
1
2
t

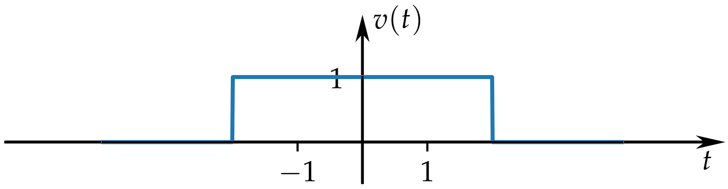
(G)


$v(t)$
1
-1
1
t

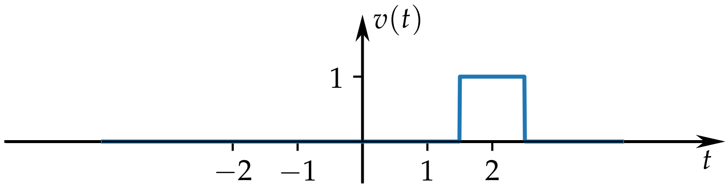
(H)


$v(t)$
1
-2
-1
1
2
t

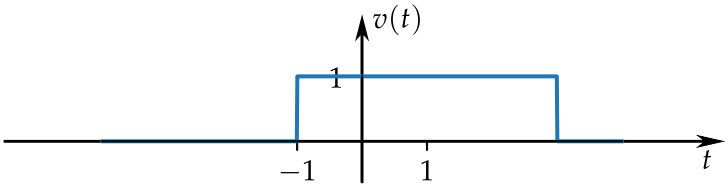
(I)



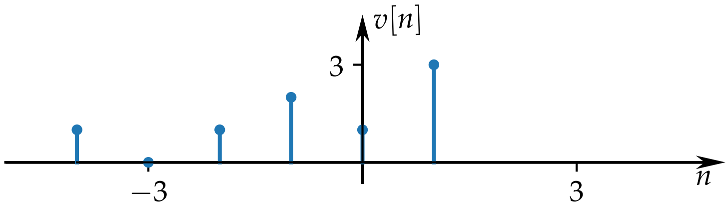
(J)



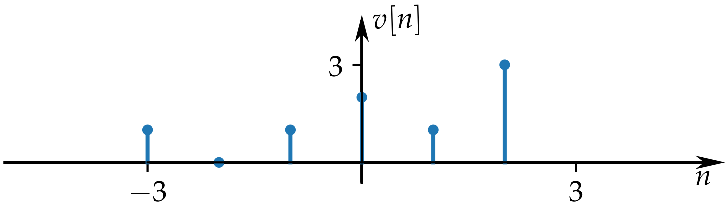
(K)



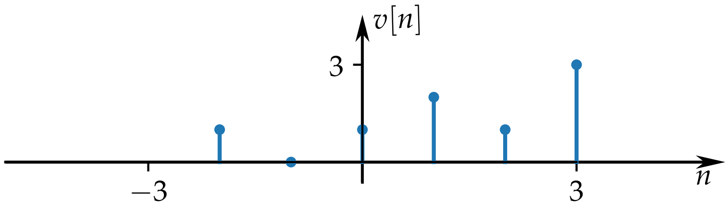
(L)



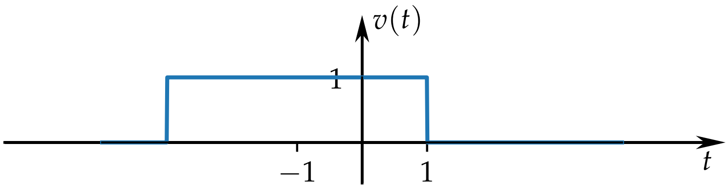
(M)



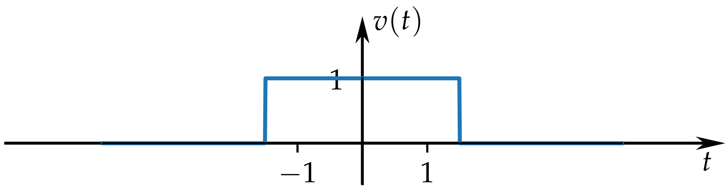
(N)



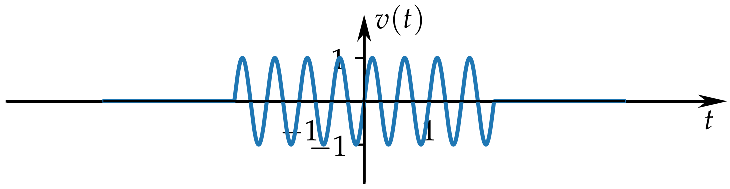
(O)



(P)



(Q)



(R)

2. (a) Sketch the lollipop plot for the signal described by $v[n] = \{1, 2, \underline{3}, 4, 5\}$.
   (b) Sketch the lollipop plot for the signal described by $v[n] = u[n+2] - u[n-2]$.
   (c) Sketch the lollipop plot for the signal described by $v[n] = \delta[n+1] + 2\delta[n] + 3\delta[n-1]$.
   (d) Sketch the lollipop plot for the signal described by $v[n] = \sum_{m=0}^{2}(n+2)\delta[n-m]$.
   (e) Sketch the lollipop plot for the signal described by $v[n] = \sum_{m=0}^{2}(m+2)\delta[n-m]$.
   (f) Express the signal described by $v[n] = \{1, 2, \underline{3}, 4, 5\}$ in terms of unit impulses.
   (g) Express the signal described by $v[n] = \{\underline{1}, 2, 3, 4, 5\}$ in terms of unit impulses.

3
Sonar II

The echo response from a target depends on a number of factors:

- distance of target from sonar
- sonar wavelength
- target size
- target orientation
- surface roughness of target
- angle of target from sonar boresight

1 Target scattering

A target is a scattering object of interest. The type of echo depends on the wavelength of the sonar and the geometry of the target.

There are two types:

specular targets are smooth compared to the wavelength[1]. They act like a mirror and reflect energy strongly in one direction where the angle of reflection is equal to the angle of incidence.

Specular targets can produce strong echoes but only if they are oriented normal to the boresight direction.

A special type of specular target called a retroreflector strongly reflects the energy back in the direction it came from.

diffuse targets are rough compared to the wavelength. They scatter energy weakly in all directions.

Corners tend to weakly scatter the energy in all directions.

[1] Typically if the rms roughness is smaller than $\lambda/8$, the target is specular.

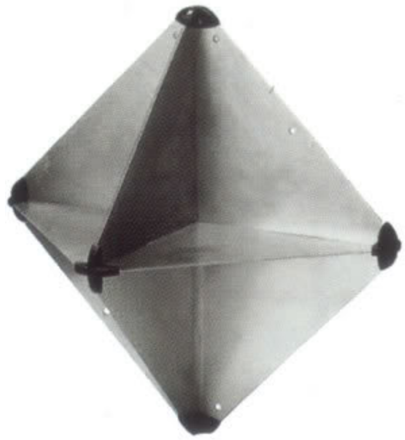


Figure 3.1: Octahedral radar corner reflector as used on yacht masts so they show up strongly on a radar.

1.1 Wavelength

The wavelength of a sonar signal can be found from the dispersion relation. For an ultrasonic signal in air,

$$c = f\lambda, \tag{3.1}$$

where c is the speed of propagation[2] , f is the frequency, and λ is the wavelength. For air $c \approx 340\,\mathrm{m/s}$. This depends on temperature and humidity.

[2] This can vary with frequency.

Modality	c (m/s)	f	λ	Notes
Air sonar	340	40 kHz	8.5 mm	
Sonar	1500	40 kHz	37.5 mm	Sea water
Radar	3×10^8	1 GHz	300 mm	
Radar	3×10^8	79 GHz	3.8 mm	
Lidar	3×10^8	430 THz	700 nm	Infrared

Table 3.1: Relationship between frequency and wavelength for different range sensors.

Lidar uses extremely short wavelengths ($\approx$ 700 nm corresponding to a frequency of 430 THz). Thus most light reflections are diffuse.

2 Spreading losses

The amplitude of the transmitted signal[3] decreases due to spreading losses. This is inversely proportional to range[4].

The amplitude of the signal scattered by the target also decreases inversely proportional to range. Thus the received echo amplitude decreases by $1/r \times 1/r = 1/r^2$.

[3] The excess sound pressure relative to the ambient pressure.

[4] The power reduces with the inverse square of the range to ensure the conservation of energy.

3 Absorption

When a wave propagates some of its energy is lost as heat. This is called absorption. The reduction in amplitude can be modelled as

$$\exp\left(-2qr\right) \tag{3.2}$$

where q is the absorption coefficient[5].

[5] This is a function of frequency. In general, the higher the frequency the greater the absorption.

4 Beam patterns

The transmitted signal from a sonar (or radar) is stronger at some angles than at others, similar to a torch beam. The signal is strongest on-axis (also called the boresight direction) and drops off as the angle from boresight increases. This variation with angle is called the *beam pattern*.

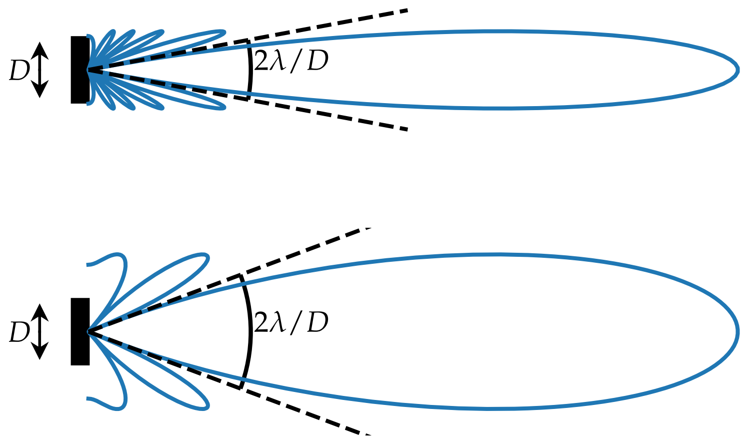


Figure 3.2: Beampatterns of a rectangular aperture. The second beampattern is at half the frequency (twice the wavelength) and is thus wider.

The angular width (in radians) of the beampattern[6] is proportional to

$$\frac{\lambda}{D}, \tag{3.3}$$

where D is the dimension of the transducer. Thus the width of the beampattern at range r is approximately given by

$$\frac{\lambda}{D}r. \tag{3.4}$$

Thus a narrow beam pattern either requires a short wavelength (high frequency) or large transducer. This is why large telescopes are required to resolve distant

[6] For an aperture transmittance $a(x)$, the beam pattern is given by $B(\theta) = A\left(\sin\theta/\lambda\right)$ where $A(u)$ is the Fourier transform of $a(x)$. So for a rectangular aperture of dimension D, where $a(x) = \operatorname{rect}\left(x/D\right)$, the beam pattern is $B(\theta) = D\operatorname{sinc}\left(D\sin\theta/\lambda\right)$.

objects in space. Similarly, high resolution photographs require a large lens[7].

Conversely, if you want a loudspeaker to spread sound[8] over a wide angle, its size needs to be small compared to the wavelength.

[7] And a large f-stop setting.

[8] This is why loudspeaker tweeters are smaller than woofers.

System	f	λ	D	λ/D
Tweeter	2 kHz	170 mm	75 mm	2.3 radians
Air sonar	40 kHz	8 mm	10 mm	0.8 radians
Camera	500 THz	600 nm	10 mm	6×10^{-5} radians

Table 3.2: Beamwidth of different transducers.

5 Time of flight

The time delay for a target at a range r is

$$\tau = \frac{2r}{c}. \tag{3.5}$$

Modality	c (m/s)	r (m)	τ	Notes
Air sonar	340	1	5.9 ms	
Sonar	1500	1	1.3 ms	Sea water
Radar	3×10^8	1	6.7 ns	
Lidar	3×10^8	1	6.7 ns	

Table 3.3: Relationship between speed of propagation and time delay for different range sensors.

The time of flight delays for lidar and radar are very short due to the fast speed of propagation. This makes time of flight techniques difficult to implement.

6 Sonar signal modelling

This section considers simple models for the sonar signals.

6.1 Transmit signal

A typical sonar signal is a periodic toneburst[9]. We can describe this pulse as a real-value continuous time signal representing the instantaneous pressure (with respect to ambient) at any instant. For example,

$$s(t) = \mathrm{rect}\left(\frac{t}{T_p}\right) \sin\left(2\pi f_0 t\right). \tag{3.6}$$

[9] Also called a gated CW (Continuous Wave) pulse.

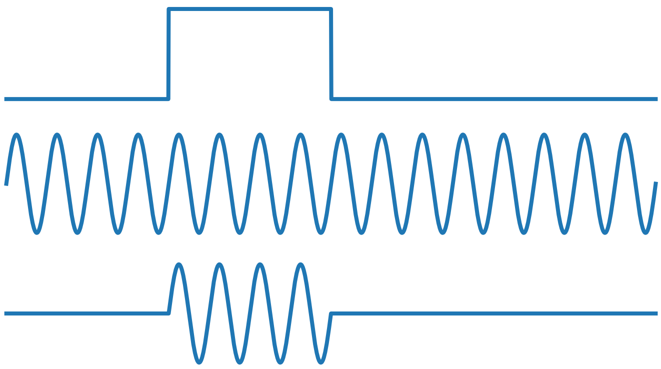


Figure 3.3: A gated CW pulse: the product of a sinewave and a rectangle function.

Here f_0 is the frequency of a sinewave and the rect function denotes the gating operation that creates a pulse of period T_p.

The toneburst is often generated by passing a square wave pulse signal through a low-pass filter; often this is performed by the transducers.

6.2 *Simple echo model*

If the transmit signal is $s(t)$, the echo signal for a single target, $e(t)$, has the form[10],

$$e(t) = \frac{a}{r^2} s\left(t - \frac{2r}{c}\right), \tag{3.7}$$

[10] Assuming that nothing is moving that will induce a Doppler shift.

where a describes the fraction of the transmit signal that is scattered by the target. The r^2 factor describes the spreading losses[11].

[11] This ignores the absorption losses.

When there are multiple targets, the echo signal is a superposition of scaled and delayed transmit signals:

$$e(t) = \sum_{i=1}^{N} \frac{a_i}{r_i^2} s\left(t - \frac{2r_i}{c}\right). \tag{3.8}$$

6.3 *Transducer response*

If we were to measure the velocity of the transmitting transducer membrane, we would see that it changes more slowly than the applied voltage pulse, see Figure 3.4.

This is because the transducers act like bandpass filters, see Figure 3.5.

The operation of the transducer can be described by a transfer function. For example, if $p(t)$ denotes the

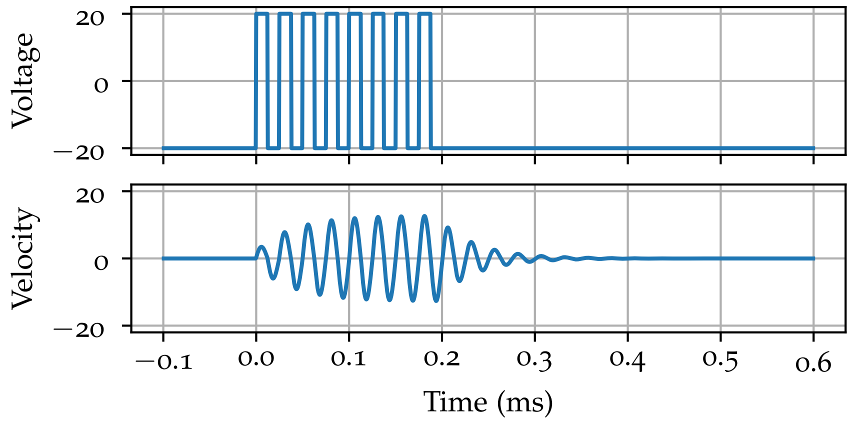


Figure 3.4: The transducer membrane velocity resulting from the applied voltage pulse. Note, when the excitation stops, the output continues for a while due to the transient response.

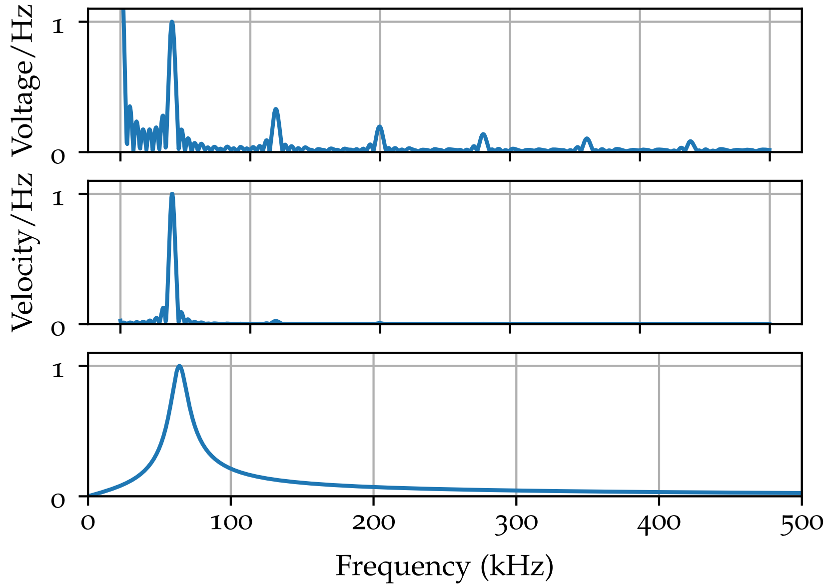


Figure 3.5: Spectra of the transducer membrane velocity and applied voltage pulse. Note, the transducer acts like a band-pass filter; shown in the bottom plot.

electrical pulse signal and $u(t)$ denotes the membrane velocity signal, then in the Laplace domain,

$$U(s) = P(s)H(s), \tag{3.9}$$

where $H(s)$ is the transfer function for the transducer.

A simple model of a transducer is mass-spring-damper second-order system with a transfer function

$$H(s) = \frac{\zeta\omega_0 s}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \tag{3.10}$$

where $\omega_0 = 2\pi f_0$ is the angular natural frequency and $\zeta = 1/Q$ is the damping factor.

4
Infrared range sensing

The IR ranging modules such as the Sharp GP2D12 estimate range by illuminating the target with a beam of IR light and then measuring the angle of the received beam[1].

[1] In comparison, a LIDAR uses time of flight to determine range. This requires fancy electronics to measure the extremely short time delays

1 *Principle of operation*

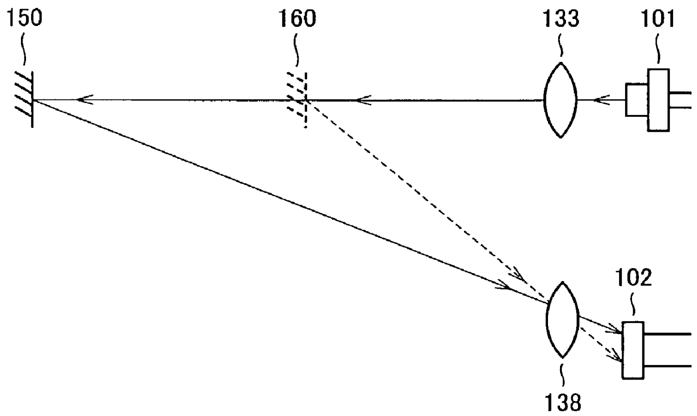


Figure 4.1: Principle of triangulating position sensor.

Consider a simple ranging system consisting of an IR source and a receiver consisting of a pinhole and a position sensitive device (PSD) [2]. A ray of light from a IR source strikes a target at a range, r. The ray reflected from the target that passes through the pinhole will fall on the PSD at a distance x. Using similar triangles,

$$\frac{x}{f} = \frac{d}{r}, \tag{4.1}$$

and thus

$$x = \frac{fd}{r}. \tag{4.2}$$

[2] This can be a lateral photodiode (LPD) or a CCD/CMOS image sensor.

The electronics produces an output voltage in proportional to the displacement of the spot from the centre of the PSD:

$$v = ax + b, \tag{4.3}$$

and thus we can model the variation in voltage with range as

$$v = k_1 + \frac{k_2}{r}. \tag{4.4}$$

The Sharp IR distance sensors use a PSD (position sensitive detector) to measure the position of the received light spot. This is a lateral photodiode with two current outputs, I_1 and I_2, where

$$x = \frac{I_1 - I_2}{I_1 + I_2}. \tag{4.5}$$

Note, the currents are equal when the spot is centred on the PSD. Electronics convert the currents to voltages[3] and I believe the division is cunningly avoided by sampling $I_1 - I_2$ when $I_1 + I_2$ reaches a preset level.

[3] Using transimpedance amplifiers.

2 Practical aspects

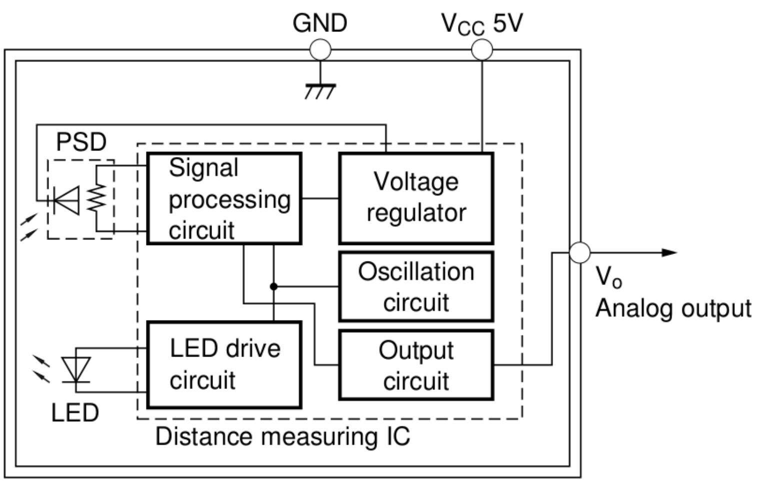


Figure 4.2: Block diagram of electronics for Sharp GP2D12 IR distance sensor. PSD is the light position sensitive detector.

1. A convex lens is used to collect more light than a pinhole.
2. An optical filter coats the receiver optics to reduce the effect of other sources of illumination[4].
3. The received light intensity reduces approximately with the square[5] of the measurement range, r.

[4] To cope with ambient IR light competing with the emitted IR light, some IR sensors modulate the IR light. The IR receiver uses a filter tuned to the modulation frequency to reject other source of the IR light.

[5] Since the transmit spot is usually smaller than the target we can neglect spreading losses for the transmit beam. However, the amplitude of the reflected beam reduces in proportional to r due to spreading losses and thus the intensity of the light (proportional to the amplitude squared) reduces with the square of r.

4. The received light intensity also depends on the reflectivity of the surface and its angle to the sensor in a similar manner to an air sonar.

5. Use a decoupling capacitor as in Figure 4.4. The datasheet for the GP2F12 says it requires 33 mA but this is an average figure. The IR LED is pulsed on and off and much more current is required when it is on. From measurements using an oscilloscope it appears that the LED is driven with a duty cycle of 0.125. Thus assuming that the average LED current is 25 mA, it requires 200 mA when it is on. This is a large switched current that needs to be drawn from the power supply, see Figure 4.3. Careful power supply considerations are required to obtain accurate readings and to prevent interference with other sensors[6].

6. If there is a slight angle, ϕ, of the transmit beam, then a more accurate model is

$$x = \frac{fr}{d} - f\tan\phi. \tag{4.6}$$

[6] This link http://askrprojects.net/lego/sharp.html contains a really detailed analysis and suggestions of how to improve the performance of Sharp IR distance sensors.

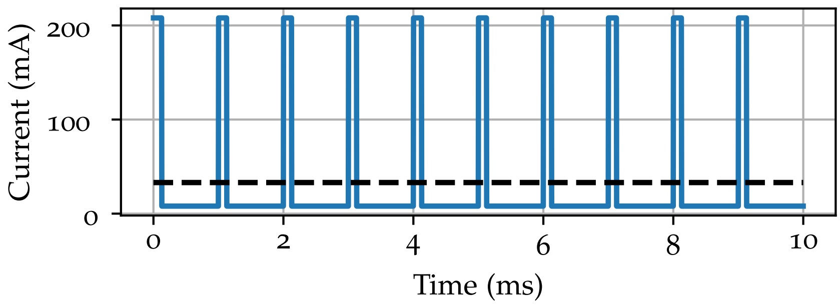


Figure 4.3: Instantaneous current draw for the GP2D12. The average is 33 mA but the peak is over 200 mA.

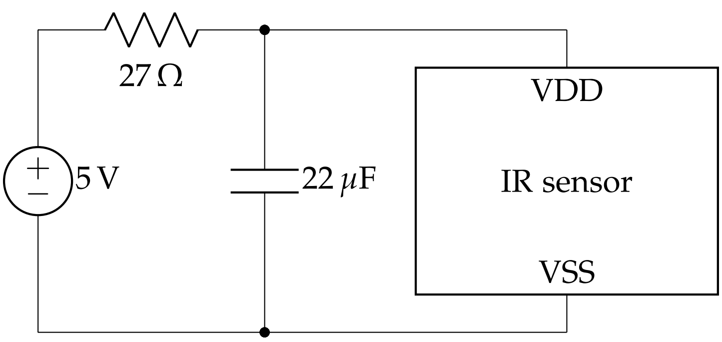


Figure 4.4: Power supply decoupling. Use a multilayer ceramic capacitor (MLCC) or tantalum capacitor placed as close to the sensor as possible. The larger the capacitor, the longer the time constant, and thus the smaller the ripple current.

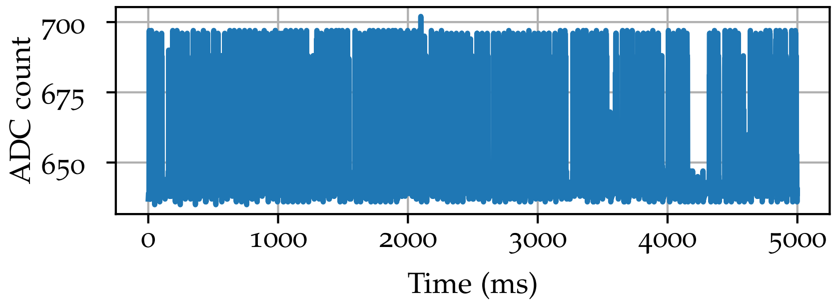


Figure 4.5: Measured data for a target at 5 cm showing the effect of a power supply fluctuation corrupting the output voltage.

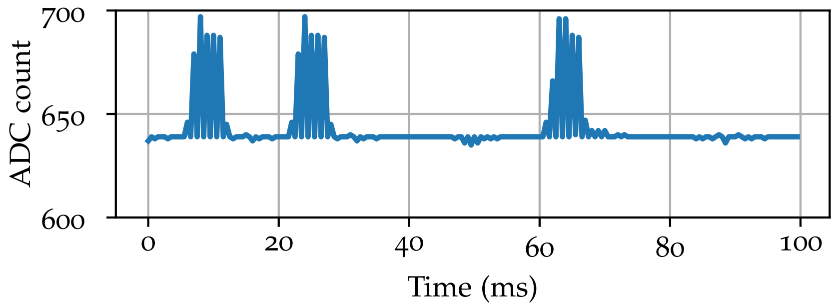


Figure 4.6: Zoomed-in region of measured data showing effect of power supply fluctuation. The inconsistency is due to aliased sampling of the interfering signal.

3 *Calibration lookup table*

A crude method for calibration is to construct a lookup table and then using linear interpolation. However, this requires a search of the table to find the bracketing interval and it is not very accurate due to measurement noise.

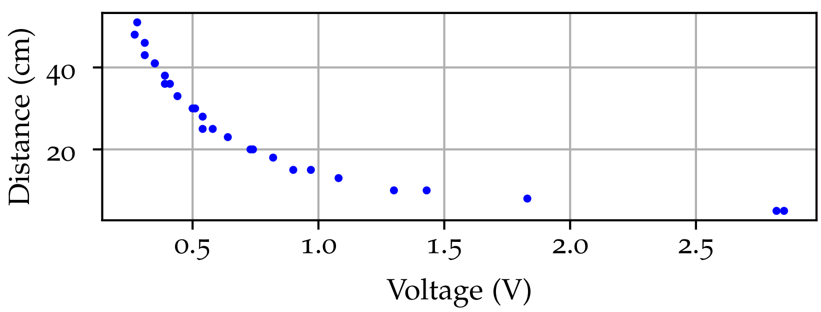


Figure 4.7: Calibration table made from measured data.

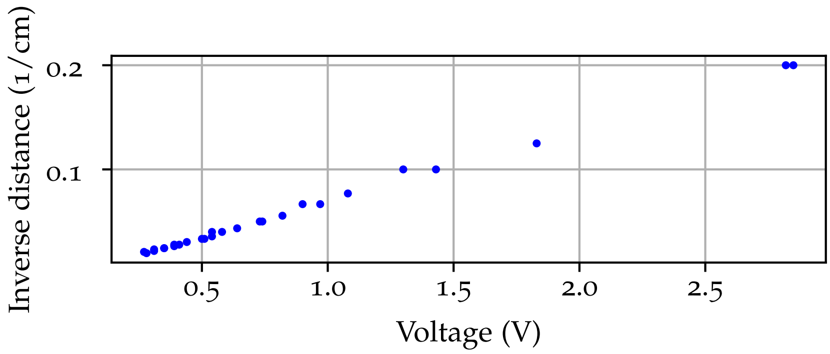


Figure 4.8: Calibration table made from measured data using reciprocal of range.

4 *Calibration model*

A simple model for the measured voltage, v, as a function of range, r, is

$$v = k_1 + \frac{k_2}{r}. \tag{4.7}$$

This is a non-linear relationship, see Figure 4.10. However, it is linear in $1/r$, see Figure 4.9.

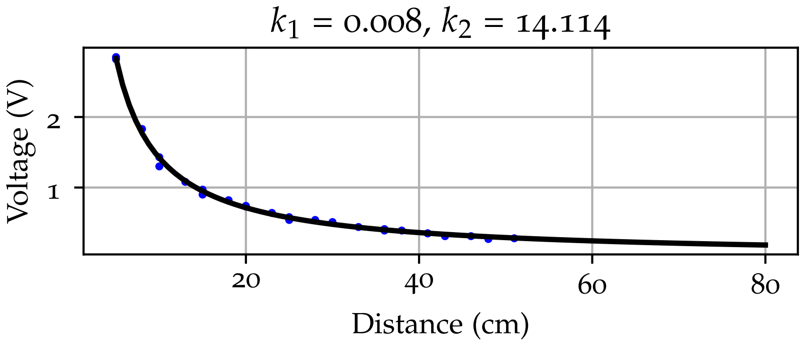


Figure 4.9: Measured data and fitted model.

5 *Calibration model estimation*

We can find the unknown parameters k_1 and k_2 using the method of least squares, using vectors of measurements $\mathbf{v} = [v_1, v_2, \cdots, v_N]^T$ and $\mathbf{r} = [r_1, r_2, \cdots, r_N]^T$. The

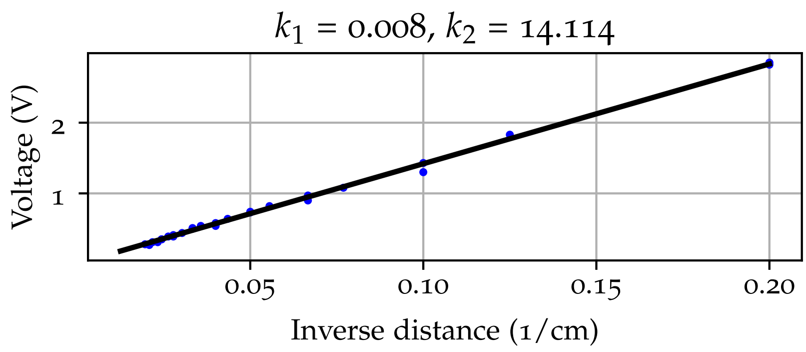


Figure 4.10: Measured data and fitted model.

system of equations can be written in matrix form as

$$\begin{bmatrix} 1 & \frac{1}{r_1} \\ 1 & \frac{1}{r_2} \\ \vdots & \vdots \\ 1 & \frac{1}{r_N} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix}. \tag{4.8}$$

This is for the form $\mathbf{Ax} = \mathbf{b}$. The least squares solution is found from the pseudo-inverse of $\mathbf{A}$,

$$\hat{\mathbf{x}} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}. \tag{4.9}$$

Once we have estimated the calibration parameters we can determine the range given a voltage measurement using the inverse model of (4.7),

$$r = \frac{k_2}{v - k_1}. \tag{4.10}$$

6 *A possibly better model*

A possibly better model calibration model with three parameters is

$$v = k_1 + \frac{k_2}{r + k_3}. \tag{4.11}$$

However, the parameters of this model cannot be found using linear least squares.

One approach is to use optimisation to find the parameters that give the minimum total squared error between the model and the measurements:

$$\hat{\mathbf{k}} = \arg\min_{\mathbf{k}} \sum_{i=1}^{N} \left(v_i - \left(k_1 + \frac{k_2}{r_i + k_3}\right)\right)^2, \tag{4.12}$$

where $\mathbf{k} = [k_1, k_2, k_3]^{\mathrm{T}}$.

A faster approach is to linearise the problem:

$$f(r_i, \mathbf{k}_n) \approx f(r_i, \mathbf{k}_{n-1}) + \frac{\partial f}{\partial \mathbf{k}} (\mathbf{k}_n - \mathbf{k}_{n-1}) \qquad (4.13)$$

where

$$f(r_i, \mathbf{k}) = k_1 + \frac{k_2}{r_i + k_3}, \qquad (4.14)$$

and $\frac{\partial f}{\partial \mathbf{k}}$ is the Jacobian,

$$\frac{\partial f}{\partial \mathbf{k}} = \begin{bmatrix} \frac{\partial f}{k_1} & \frac{\partial f}{k_2} & \frac{\partial f}{k_3} \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{r_i + k_3} & \frac{-1}{(r_i + k_3)^2} \end{bmatrix}. \qquad (4.15)$$

The linearised least squares problem needs to be solved iteratively where $\mathbf{x} = \mathbf{k}_n - \mathbf{k}_{n-1}$, $\mathbf{b} = \mathbf{v} - f(\mathbf{d}, \mathbf{k}_n)$, and

$$\mathbf{A} = \begin{bmatrix} 1 & \frac{1}{r_1 + k_3} & \frac{-1}{(r_1 + k_3)^2} \\ 1 & \frac{1}{r_2 + k_3} & \frac{-1}{(r_2 + k_3)^2} \\ \vdots & \vdots & \vdots \\ 1 & \frac{1}{r_N + k_3} & \frac{-1}{(r_N + k_3)^2} \end{bmatrix}. \qquad (4.16)$$

Thus given an initial estimate of $\mathbf{k}$, say $\mathbf{k}_0 = [0, 0, 0]^{\mathrm{T}}$, we iteratively improve the estimate using

$$\mathbf{k}_n = \mathbf{k}_{n-1} + \left(\mathbf{A}_{n-1}^{\mathrm{T}} \mathbf{A}_{n-1} \right)^{-1} \mathbf{A}_{n-1}^{\mathrm{T}} \mathbf{b}_{n-1}. \qquad (4.17)$$

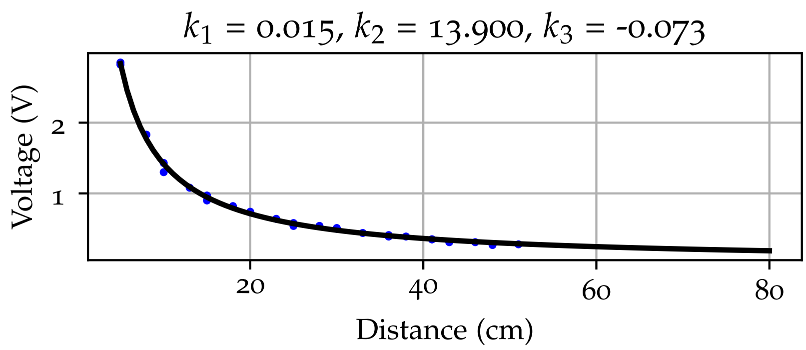


Figure 4.11: Measured data and fitted three parameter model.

7 Python code to solve linear least squares

```
from numpy import loadtxt, ones, zeros, linspace
from numpy.linalg import lstsq
from matplotlib.pyplot import figure, show, savefig

def model(r, k):
    return k[0] + k[1] / r


def calculate_A(r):
    N = len(r)
    A = ones((N, 2))
    A[:, 1] = 1.0 / r
    return A

def model_least_squares_fit(r, v):
    A = calculate_A(r)

    # Use least squares to estimate the parameters.
    k, res, rank, s = lstsq(A, v)
    return k

a = loadtxt('calib_data.csv', delimiter=',')
r = a[:, 0]
v = a[:, 1]

k = model_least_squares_fit(r, v)

r1 = linspace(5, 80, 100)
v1 = model(r1, k)

fig = figure()
ax = fig.add_subplot(111)
ax.plot(1.0 / r, v, 'bo')
ax.plot(1.0 / r1, v1, 'k', linewidth=2)
ax.set_xlabel('Inverse distance (1/cm)')
ax.set_ylabel('Voltage (V)')
ax.set_title('$k_1$ = %.3f, $k_2$ = %.3f' % (k[0], k[1]))
ax.grid(True)
savefig(__file__.replace('.py', '.pgf'), bbox_inches='tight')
```

Listing 4.1: Example Python code to solve the linear least squares problem for the calibration coefficients.

8 Python code to solve non-linear least squares

```
from numpy import loadtxt, ones, zeros, linspace
from numpy.linalg import lstsq
from matplotlib.pyplot import figure, show, savefig

def model(r, k):
    return k[0] + k[1] / (r + k[2])


def model_nonlinear_least_squares_fit(r, v, iterations=5):

    N = len(r)
    A = ones((N, 3))
    k = zeros(3)

    for i in range(iterations):
        # Calculate Jacobians for current estimate of parameters.
        for n in range(N):
            A[n, 1] = 1 / (r[n] + k[2])
            A[n, 2] = -k[1] / (r[n] + k[2])**2

        # Use least squares to estimate the parameters.
        deltak, res, rank, s = lstsq(A, v - model(r, k))
        k += deltak
        print(k)
    return k

a = loadtxt('calib_data.csv', delimiter=',')
r = a[:, 0]
v = a[:, 1]

k = model_nonlinear_least_squares_fit(r, v)

r1 = linspace(5, 80, 100)
v1 = model(r1, k)

fig = figure()
ax = fig.add_subplot(111)
ax.plot(r, v, 'bo')
ax.plot(r1, v1, 'k', linewidth=2)
ax.set_xlabel('Distance (cm)')
ax.set_ylabel('Voltage (V)')
ax.set_title('$k_1$ = %.3f, $k_2$ = %.3f, $k_3$ = %.3f' % (k[0], k[1], k[2]))
ax.grid(True)
savefig(__file__.replace('.py', '.pgf'), bbox_inches='tight')
```

Listing 4.2: Example Python code to solve the non-linear least squares problem for the calibration coefficients.

Device	Range (cm)	d (cm)
Sharp GP2D120	4–30	1.97
Sharp GP2D12	10–80	2.0
Sharp GP2Y0A710K0F	100–550	3.8
Sharp GP2Y0A02YK0F	20–150	1.98

Table 4.1: Comparison of IR range sensors.

9 *Example GP2D12 Arduino code*

Interfacing the GP2D12 to a MCU is easy. It requires +5 V (well decoupled) and a ground connection. It outputs an updated analogue voltage every 40 ms.

```
// For use with Sharp IR position sensors with analog output
// This samples much faster than necessary since the update rate is 25 Hz
const unsigned int repetition_period_us = 500;
const unsigned int IR_output_pin = A0;
const unsigned long serial_baud_rate = 115200;

void setup()
{
    Serial.begin(serial_baud_rate);
}

void loop()
{
    static unsigned long time_us_previous = 0;
    unsigned long time_us;
    int value;

    time_us = micros();
    if ((time_us - time_us_previous) < repetition_period_us)
        return;
    time_us_previous = time_us;

    value = analogRead(IR_output_pin);

    Serial.print(value);
    Serial.print("\n");
}
```

Listing 4.3: Example Arduino code to drive a GP2D12 IR range sensor.

5
Fourier transform

The Fourier transform converts a signal into its *spectrum*. This provides insight into operations such as:

- Filtering
- Modulation
- Diffraction

It is used in many fields[1]:

- Vibration analysis
- Communications
- Optics, acoustics
- Medical imaging (ultrasound, X-ray, MRI)
- Remote sensing (radar, sonar, seismic imaging, radio astronomy)
- Crystallography

[1] Especially imaging systems since the diffraction of waves can be approximated by a Fourier transform (in the far field).

Fourier analysis is important for linear time-invariant (LTI) systems since it avoids the direct evaluation of convolution integrals.

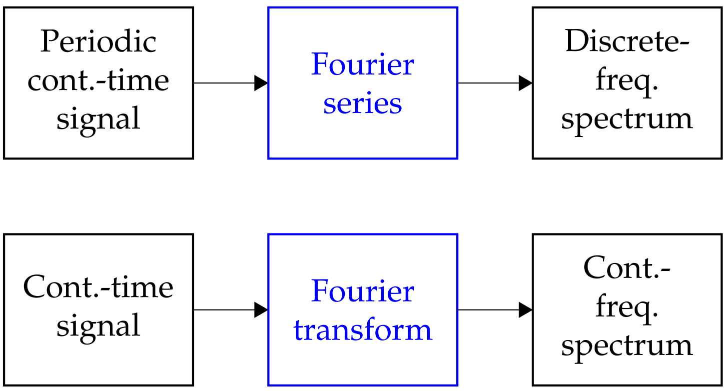


Figure 5.1: Comparison of Fourier series and Fourier transform.

1 *The Fourier series*

Fourier series analysis decomposes a **periodic** continuous-time waveform into a discrete-frequency spectrum[2].

[2] See https://en.wikipedia.org/wiki/Triangle_wave and https://en.wikipedia.org/wiki/Square_wave for good animations of synthesising triangle and square waves.

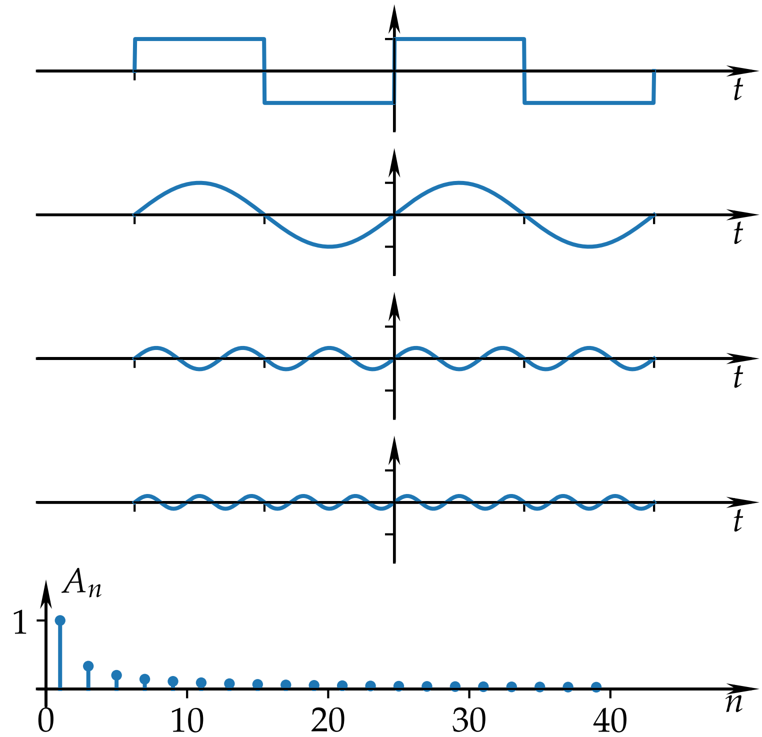


Figure 5.2: Fourier series analysis of a square wave showing first three harmonics. The bottom plot shows the spectrum having only odd harmonics.

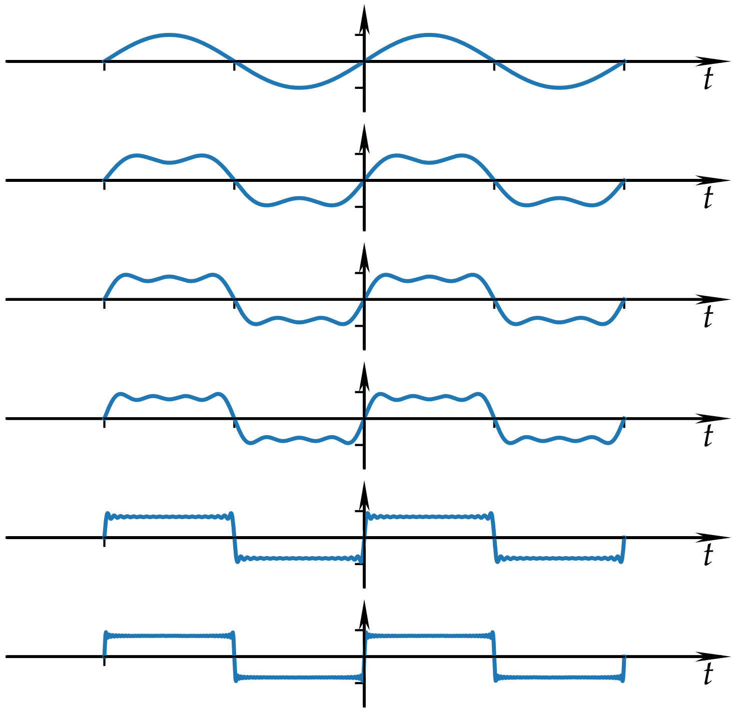


Figure 5.3: Fourier series synthesis of a square wave (1, 2, 3, 4, 20, and 40 harmonics). No matter how many harmonics are summed, a perfect square wave can never be synthesised.

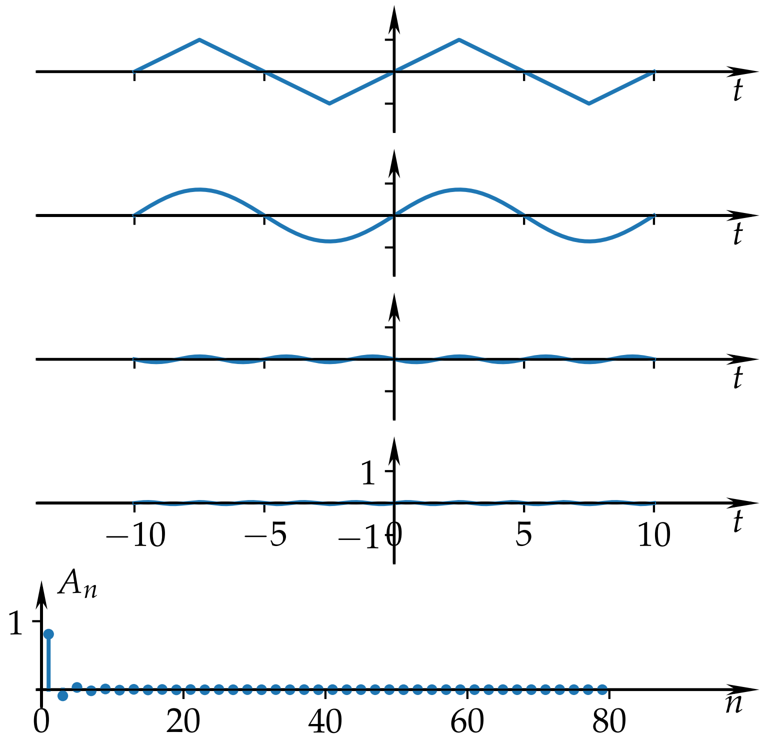


Figure 5.4: Fourier series analysis of a triangle wave showing first three harmonics. The bottom plot shows the spectrum.

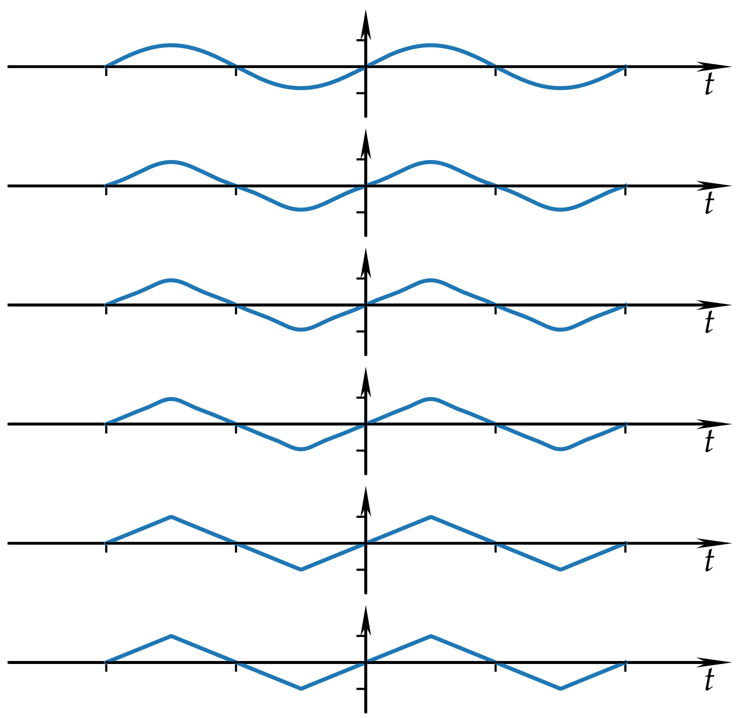


Figure 5.5: Fourier series synthesis of a triangle wave (1, 2, 3, 4, 30, and 20 harmonics). Like a square wave, a triangle wave only has odd harmonics. However, a much better approximation is obtained since the coefficients rapidly decrease.

2 *The Fourier transform*

The Fourier transform is similar to the Fourier series but unlike the Fourier series it does not require a periodic signal. It has many definitions; the most common is[3]

$$X(f) = \int_{-\infty}^{\infty} x(t) \exp(-\mathrm{j}2\pi f t)\,\mathrm{d}t. \qquad (5.1)$$

[3] From Euler's theorem, $\exp(-\mathrm{j}2\pi f t) = \cos(2\pi f t) - \mathrm{j}\sin(2\pi f t)$.

$x(t)$ is a real time signal and $X(f)$ is a complex-valued spectrum that combines the magnitude and phase of the spectrum,

$$X(f) = |X(f)| \exp(\mathrm{j}\psi(f)). \qquad (5.2)$$

The Fourier transform is often denoted as

$$X(f) = \mathcal{F}\{x(t)\}, \qquad (5.3)$$

or as a Fourier transform pair denoted by

$$X(f) \longleftrightarrow x(t). \qquad (5.4)$$

2.1 *Hermitian spectrum*

A quirk of the Fourier transform is that it introduces negative frequencies[4]. This is a mathematical artefact; you cannot measure them. However, if the signal $x(t)$ is real, the spectrum, $X(f)$, has a *Hermitian* symmetry, where the negative frequencies are the complex conjugate of the positive frequencies[5]:

$$X(-f) = X^*(f) = |X(f)| \exp(-\mathrm{j}\psi(f)). \qquad (5.5)$$

[4] They become useful when we consider complex signals but this is beyond the scope of this course.

[5] Thus the negative frequencies have the same magnitude as the positive frequencies but with negative phase.

This symmetry means that we do not need the negative frequency components of $X(f)$ since we can infer them from the positive frequency components.

2.2 *Spectral density*

Consider the units of a spectrum. Say the units of the signal, $x(t)$, are volts (V). Then since $\mathrm{d}t$ has units of seconds (s) in the definition of the Fourier transform (5.1), the units of $X(f)$ are volt-seconds (V s). However, we usually interpret this as volts per hertz (V/Hz). We thus call $X(f)$ the *spectral density*[6].

[6] In comparison, the Fourier series produces a discrete sequence of amplitudes each with units of volts.

3 Fourier transform pairs

The Fourier transform integral cannot be evaluated for any signal with infinite energy, e.g., DC and AC signals. However, with some mathematical skullduggery and the use of Dirac deltas, these signals can be assigned a spectrum, see Table 5.1.

3.1 DC

A DC signal does not formally have a Fourier transform; its spectrum is a Dirac delta:

$$1 \longleftrightarrow \delta(t). \tag{5.6}$$

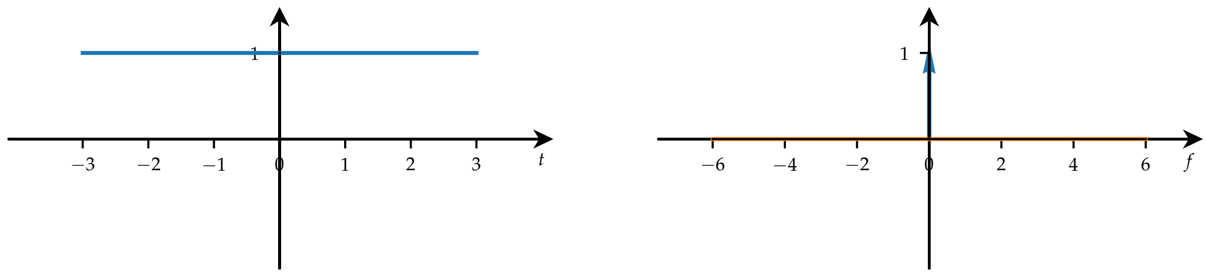


Figure 5.6: The spectrum of a DC signal is a Dirac delta.

3.2 AC

AC signals do not formally have a Fourier transform; their spectra are a pair of Dirac deltas[7].

[7] Mirrored to ensure Hermitian symmetry.

$$\cos\left(2\pi f_0 t\right) \longleftrightarrow \frac{1}{2}\delta(f - f_0) + \frac{1}{2}\delta(f + f_0), \tag{5.7}$$

$$\sin\left(2\pi f_0 t\right) \longleftrightarrow -\mathrm{j}\frac{1}{2}\delta(f - f_0) + \mathrm{j}\frac{1}{2}\delta(f + f_0). \tag{5.8}$$

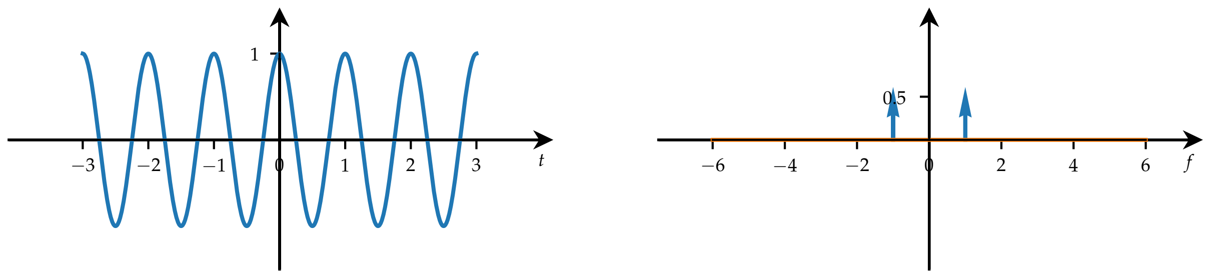


Figure 5.7: The spectrum of a cosinusoidal AC signal is a pair of Dirac deltas.

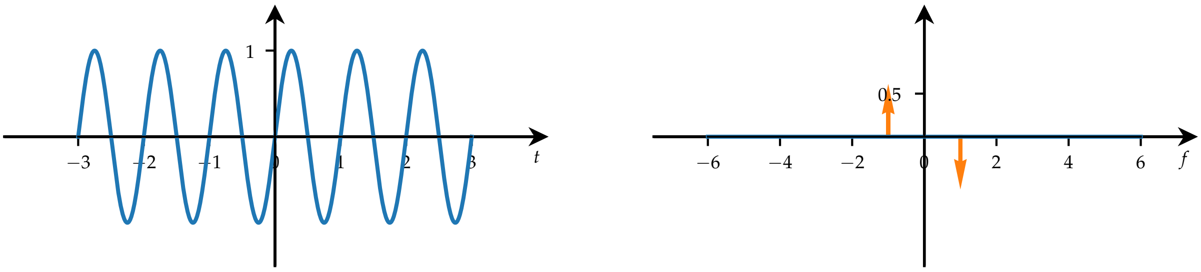


Figure 5.8: The spectrum of a sinusoidal AC signal is a pair of imaginary Dirac deltas. Blue denotes the real component; orange the imaginary component.

3.3 *Rect*

The spectrum of a rectangle signal is a sinc function:

$$\text{rect}(t) \longleftrightarrow \text{sinc}(f). \tag{5.9}$$

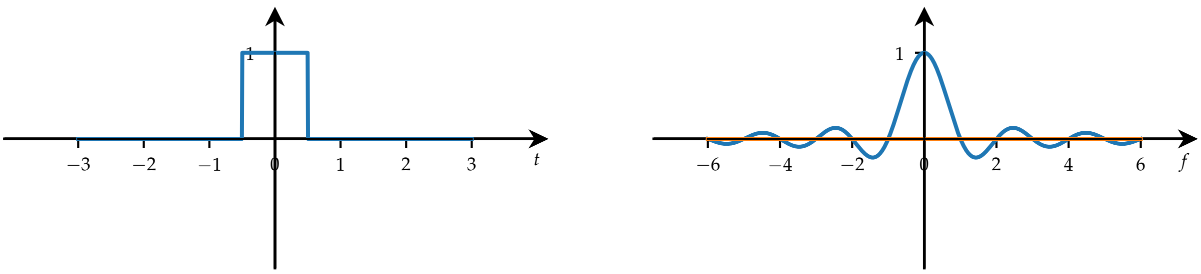


Figure 5.9: The spectrum of a rect signal is a sinc function.

The sinc[8] function is defined by

$$\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}. \tag{5.10}$$

The value of $\text{sinc}(t)$ at $t = 0$ appears indeterminate, since

$$\text{sinc}(0) = \frac{\sin(\pi 0)}{\pi 0} = \frac{0}{0}. \tag{5.11}$$

However, using l'Hôpital's law, this evaluates to 1.

[8] sinc is an abbreviation for cardinal sine. It is similar but subtly different from $\sin(t)/t$.

$x(t)$	$X(f)$
1	$\delta(f)$
$\delta(t)$	1
$\cos(2\pi f_0 t)$	$\frac{1}{2}\delta(f - f_0) + \frac{1}{2}\delta(f + f_0)$
$\sin(2\pi f_0 t)$	$-\mathrm{j}\frac{1}{2}\delta(f - f_0) + \mathrm{j}\frac{1}{2}\delta(f + f_0)$
$\text{rect}(t)$	$\text{sinc}(f)$
$\text{sinc}(t)$	$\text{rect}(f)$

Table 5.1: Common signals and their Fourier transform.

4 The Dirac delta

Not all functions have a Fourier transform[9]. For example, consider a constant value of 1,

$$x(t) = 1. \tag{5.12}$$

This has a Laplace transform of $1/s$. However, if we use calculus to solve the Fourier transform integral we find that the integral does not converge.

Since, the Fourier transform concept is so useful, the work-around for awkward functions is to use the Dirac delta. This has two key properties:

$$\delta(t) = 0 \quad t \neq 0, \tag{5.13}$$

and

$$\int_{-\infty}^{\infty} \delta(t)\mathrm{d}t = 1. \tag{5.14}$$

Using these properties,

$$x(t)\delta(t) = x(0)\delta(t), \tag{5.15}$$

and

$$x(t)\delta(t-u) = x(u)\delta(t-u). \tag{5.16}$$

The Dirac delta[10] is an idealised but useful concept. However, it can cause mathematical grief. The trick is to consider the Dirac delta as a limiting case of a better behaved function.

[9] Indeed, most of the common signals, such as DC, AC, Heaviside step, do not have a Fourier transform without some mathematical sleight of hand. In comparison, the Laplace transform exists for a wider range of functions.

[10] Mathematically, the Dirac delta is not a function; it is a generalised distribution.

5 The Fourier transform of a DC signal

The Fourier transform of a unit DC signal is

$$\begin{aligned} X(f) &= \int_{-\infty}^{\infty} 1 \exp\left(-\mathrm{j}2\pi f t\right) \mathrm{d}t, && (5.17)\\ &= \left[\frac{\exp\left(-\mathrm{j}2\pi f t\right)}{-\mathrm{j}2\pi f}\right]_{-\infty}^{\infty}. && (5.18)\end{aligned}$$

At this point the integral cannot be solved since it is an oscillatory function and does not converge.

The mathematical hack to this problem is to apply a convergence factor, $\exp\left(-\alpha\,|t|\right)$, and to consider a limit,

$$X(f) = \lim_{\alpha\to 0} \int_{-\infty}^{\infty} \exp\left(-\alpha\,|t|\right) \exp\left(-\mathrm{j}2\pi f t\right) \mathrm{d}t. \tag{5.19}$$

Let's define

$$X(f) = \lim_{\alpha\to 0} I(f), \tag{5.20}$$

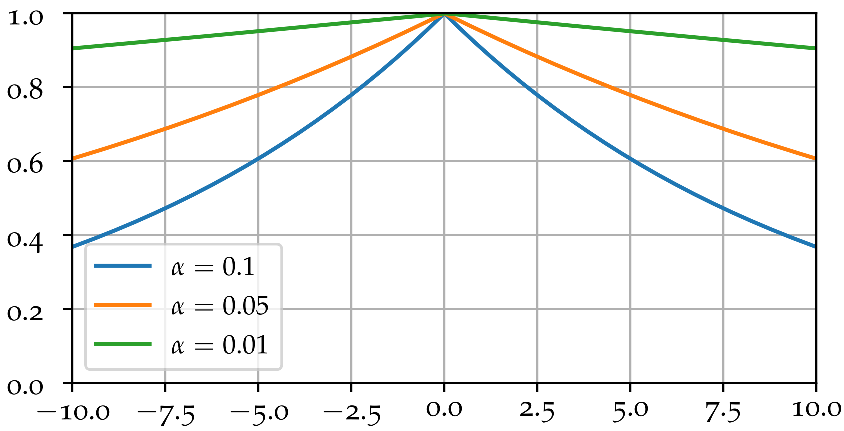


Figure 5.10: $\exp(-\alpha|t|)$ as a convergence factor.

where

$$
\begin{aligned}
I(f) &= \int_{-\infty}^{\infty} \exp(-\alpha|t|)\exp(-j2\pi f t)\,dt, && (5.21)\\
&= \int_{0}^{\infty} \exp(-\alpha t)\exp(-j2\pi f t)\,dt + \int_{-\infty}^{0} \exp(\alpha t)\exp(-j2\pi f t)\,dt, && (5.22)\\
&= \int_{0}^{\infty} \exp(-\alpha t)\exp(-j2\pi f t)\,dt + \int_{0}^{\infty} \exp(-\alpha t)\exp(j2\pi f t)\,dt, && (5.23)\\
&= \left[\frac{\exp(-(\alpha + j2\pi f)\,t)}{-(\alpha + j2\pi f)}\right]_0^{\infty} + \left[\frac{\exp(-(\alpha - j2\pi f)\,t)}{-(\alpha - j2\pi f)}\right]_0^{\infty}, && (5.24)\\
&= \left(0 + \frac{1}{\alpha + j2\pi f}\right) + \left(0 + \frac{1}{\alpha - j2\pi f}\right), && (5.25)\\
&= \frac{(\alpha - j2\pi f) + (\alpha + j2\pi f)}{(\alpha + j2\pi f)(\alpha - j2\pi f)}, && (5.26)\\
&= \frac{2\alpha}{\alpha^2 + (2\pi f)^2}. && (5.27)
\end{aligned}
$$

Thus

$$\mathcal{F}\{1\} = \lim_{\alpha \to 0} \frac{2\alpha}{\alpha^2 + (2\pi f)^2}. \tag{5.28}$$

This is plotted in Figure 5.11. Note, how the function gets narrower and taller as α gets smaller, while retaining unit area (see Figure 5.11). In the limit it has has zero width and infinite height and is equivalent to the Dirac delta[11]. Thus we have a Fourier transform pair:

$$1 \longleftrightarrow \delta(f). \tag{5.29}$$

This states that a DC signal has a spectrum with no frequency components except at $f = 0$.

[11] This is not a function but a generalised distribution.

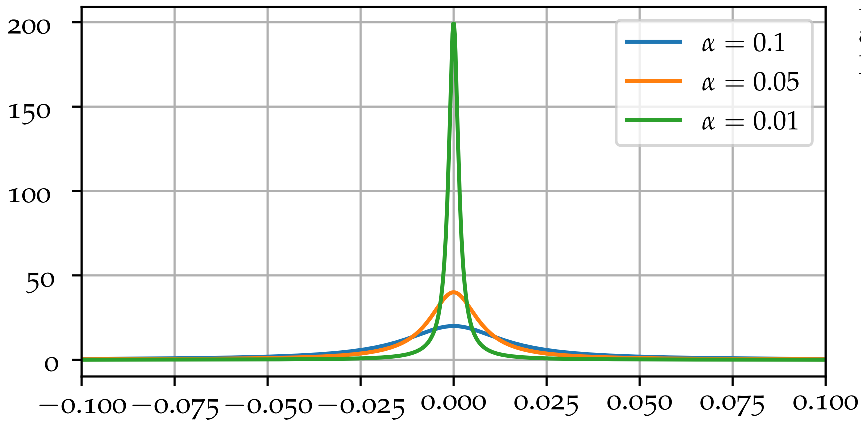


Figure 5.11: The Dirac delta as a generalised distribution: $\lim_{\alpha \longrightarrow 0} \frac{2\alpha}{\alpha^2 + (2\pi f)^2}$.

Due to the symmetry of the forward and inverse Fourier transforms, there is a similar Fourier transform pair

$$\delta(t) \longleftrightarrow 1. \tag{5.30}$$

Thus a Dirac delta has a spectrum with all frequency components weighted equally.

It is worth noting that the Laplace transform exists for a much wider variety of signals and does not require any mathematical skullduggery[12]. For example,

$$\mathcal{L}\left\{1\right\} = \frac{1}{s}. \tag{5.31}$$

[12] Note, since DC is not a causal signal, we cannot find the Fourier transform from the Laplace transform by substituting s with $\mathrm{j}2\pi f$.

5.1 *Alternative derivation*

Another way to determined the Fourier transform of 1 is to the considered the limiting case of the Fourier transform of a rect function.

$$\begin{aligned}
\mathcal{F}\left\{1\right\} &= \int_{-\infty}^{\infty} 1 \times \exp\left(-\mathrm{j}2\pi f t\right) \mathrm{d}t, && (5.32)\\
&= \lim_{T\to\infty} \int_{-\frac{T}{2}}^{\frac{T}{2}} \exp\left(\mathrm{j}2\pi f t\right) \mathrm{d}t, && (5.33)\\
&= \lim_{T\to\infty} \left[\frac{\exp\left(\mathrm{j}2\pi f t\right)}{\mathrm{j}2\pi t}\right]_{-\frac{T}{2}}^{\frac{T}{2}}, && (5.34)\\
&= \lim_{T\to\infty} \frac{\exp\left(\mathrm{j}\pi f T\right)}{\mathrm{j}2\pi t} - \frac{\exp\left(-\mathrm{j}\pi f T\right)}{\mathrm{j}2\pi t}, && (5.35)\\
&= \lim_{T\to\infty} \frac{\sin(\pi f T)}{\pi t}, && (5.36)\\
&= \lim_{T\to\infty} T \operatorname{sinc}(f T). && (5.37)
\end{aligned}$$

This result is plotted in Figure 5.12.

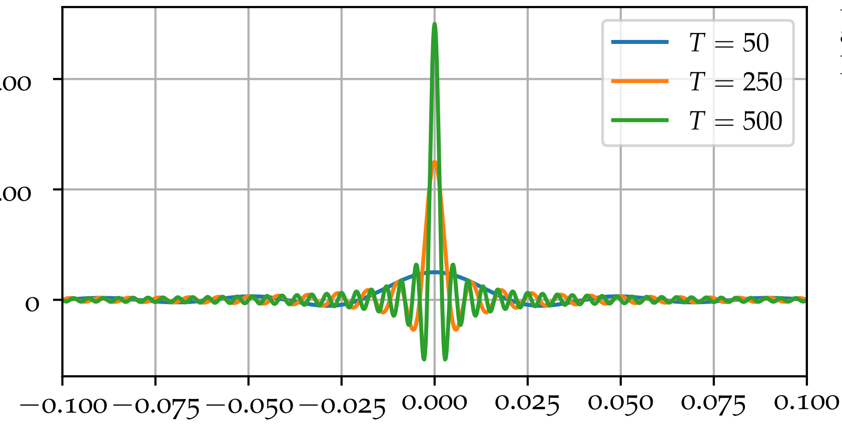


Figure 5.12: The Dirac delta as a generalised distribution: $\lim_{T \longrightarrow \infty} T \operatorname{sinc}(fT)$.

6 *Exercises*

These exercises are for the mathematically curious.

1. (a) Using calculus determine the Fourier transform of $\operatorname{rect}(t)$ and sketch the signal and the magnitude of its spectrum, where

$$\operatorname{rect}(t) = \begin{cases} 1 & |t| < 0.5, \\ 0 & |t| \geq 0.5. \end{cases}$$

   (b) Using calculus determine the Fourier transform of $\operatorname{rect}\left(\frac{t}{T}\right)$ and sketch the signal and the magnitude of its spectrum. Note,

$$\begin{aligned} \operatorname{rect}\left(\frac{t}{T}\right) &= \begin{cases} 1 & \left|\frac{t}{T}\right| < 0.5, \\ 0 & \left|\frac{t}{T}\right| \geq 0.5, \end{cases}, \\ &= \begin{cases} 1 & |t| < 0.5T, \\ 0 & |t| \geq 0.5T. \end{cases} \end{aligned}$$

   (c) Using calculus determine the Fourier transform of $\operatorname{rect}(t - \tau)$ and sketch the signal, the magnitude of its spectrum, and the phase of the spectrum. *Hint: use a change of variable where $u = t - \tau$ and $\mathrm{d}t = \mathrm{d}u$.*

   (d) Using calculus determine the Fourier transform of $\operatorname{rect}\left(\frac{t}{T} - \frac{1}{2}\right)$ and sketch the signal, the magnitude of its spectrum, and the phase of the spectrum.

   (e) What happens if you try using calculus to determine the Fourier transform of 1?

   (f) What happens if you try using calculus to determine the inverse Fourier transform of 1?

6
Oscilloscope I

Think twice before using a multimeter! They are only useful for continuity testing or perhaps testing of a battery voltage. Real engineers will use an oscilloscope, a.k.a., scope.

1 Multimeter

A multimeter is based on a high impedance voltmeter. Current can be measured from the voltage dropped over a low resistance shunt.

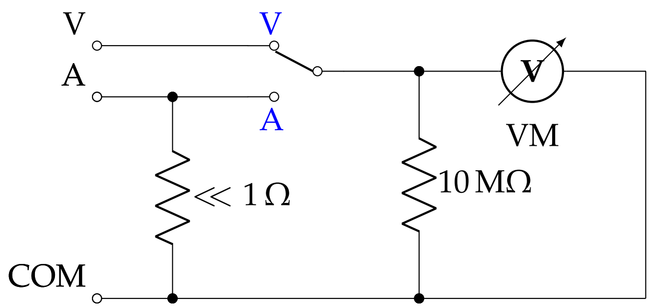


Figure 6.1: Simplified circuit of a multimeter.

Warnings:

1. Do not measure voltage on a current setting; this is the most common way of destroying a multimeter.
2. Be careful measuring mains voltage. Many multimeters default to a DC setting and will show 0 V.

2 Oscilloscope operation

1. When a new probe is used on a scope, it must be compensated for that scope.
2. The ground clip is connected to mains earth so be careful where you connect it to.

2.1 *Probe compensation*

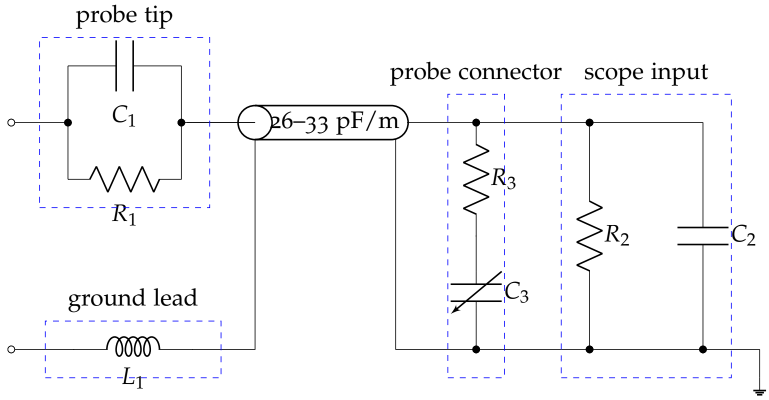


Figure 6.2: Schematic of scope probe, cable, and input. R_1 =9 MΩ, R_2 =1 MΩ, C_1 =8–12 pF, C_2 =20 pF, R_3 =500 Ω, C_3 =7–50 pF, L_1 ≈ 60 nH depending on loop area.

10× scope probes have an adjustable capacitor that affects the displayed pulse shape. This must be adjusted before the scope is used by connecting the scope probe (and ground) to the calibration signals on the scope. The goal is to adjust the probe so that no undershoot or overshoot is visible.

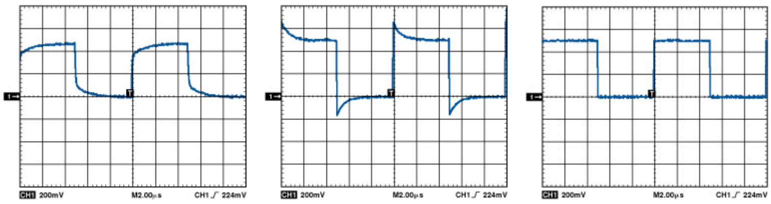


Figure 6.3: Probe compensation: under, over, and proper. From http://www.analog.com/library/analogdialogue/archives/41-03/time_domain.html.

2.2 *Ground connection*

The ground connections for each scope probe are connected inside the scope and to the mains ground. Thus care is required especially with power electronics[1] since you can short things out by connecting the ground to the wrong place.

The scope probe ground lead must be connected close to the measurement point. If this is not connected, a large 50 Hz signal will be measured (see Figure 6.4).

[1] Sometimes it is necessary to use an isolating transformer to break the common ground connection.

3 *Exercises*

1. When measuring the voltage across a large motor with a multimeter you hear a load bang and find that the multimeter no longer works. What happened?

2. You wisely decide to check that the mains is off before working on some mains powered equipment.

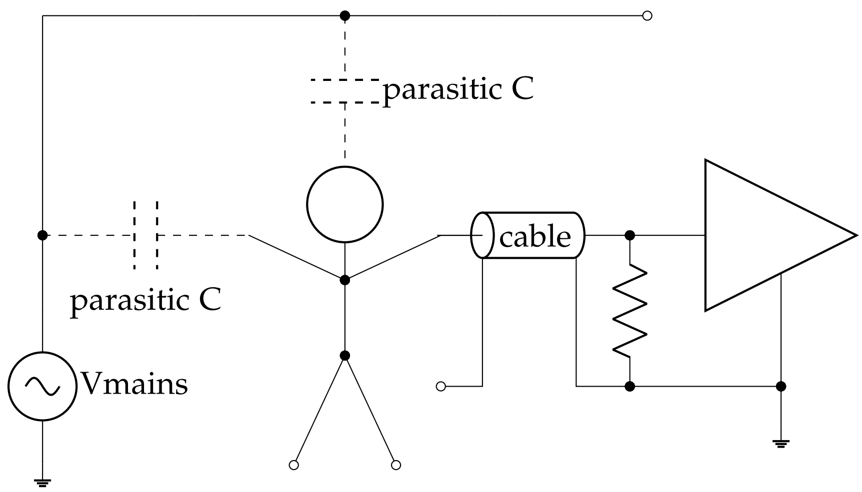


Figure 6.4: Schematic showing pickup of electric field due to mains when the scope probe is touched. The changing electric field produces a current that flows through parasitic capacitances and develops a voltage across the input resistance of the amplifier.

However, after using a multimeter to check that the voltage is zero, you get an electric shock. What happened?

3. Why do scope probes need to be compensated?
4. When do scope probes need to be compensated?
5. When working on some power electronics, you clip on the scope ground probe and there is a loud bang and some smoke appears! Why?

7
The Fourier Transform II

1 Inverse Fourier transform

The inverse Fourier transform requires $X(f)$ to be known for all frequencies, both positive and negative. It recovers the signal using a similar integral to the Fourier transform,

$$x(t) = \int_{-\infty}^{\infty} X(f) \exp\left(\mathrm{j}2\pi f t\right) \mathrm{d}f. \tag{7.1}$$

Note the symmetry with the forward Fourier transform,

$$X(f) = \int_{-\infty}^{\infty} x(t) \exp\left(-\mathrm{j}2\pi f t\right) \mathrm{d}t. \tag{7.2}$$

1.1 Inverse Laplace transform

The Fourier and Laplace transforms are similar. However, the inverse Fourier transform is much easier to evaluate than the inverse Laplace transform[1].

The inverse Fourier transform is valid of all t, unlike the inverse Laplace transform. This can be seen by considering the Laplace transform definition:

$$X(s) = \int_{0-}^{\infty} x(t) \exp\left(-st\right) \mathrm{d}t. \tag{7.3}$$

Since the lower limit[2] is not $-\infty$, an infinite number of functions have the same Laplace transform. For example,

$$\mathcal{L}\left\{1\right\} = \frac{1}{s}, \tag{7.4}$$

$$\mathcal{L}\left\{H(t)\right\} = \frac{1}{s}. \tag{7.5}$$

Thus the inverse Laplace transform of $1/s$ is ambiguous; the result is only defined for $t \geq 0$, e.g.,

$$\mathcal{L}^{-1}\left\{\frac{1}{s}\right\} = 1 \quad t \geq 0. \tag{7.6}$$

[1] This is a contour integral over a complex variable s.

[2] The lower time limit is $0-$ to capture Dirac deltas and Heaviside steps.

2 *Fourier transform theorems*

The Fourier transform of an arbitrary signal can be solved using integration. However, it is easier to use a table of common transforms and Fourier transform theorems.

2.1 *Linearity theorem*

Integration is a linear operator and so:

$$ax(t) \longleftrightarrow aX(f), \tag{7.7}$$

and

$$x(t) + y(t) \longleftrightarrow X(f) + Y(f). \tag{7.8}$$

Thus, in general,

$$ax(t) + by(t) \longleftrightarrow aX(f) + bY(f). \tag{7.9}$$

This theorem is useful when a signal is the superposition of other signals.

2.2 *Similarity theorem*

The similarity theorem is:

$$x(at) \longleftrightarrow \frac{1}{|a|} X\left(\frac{f}{a}\right). \tag{7.10}$$

Thus a time scaling results in an inverse frequency scaling. So if we make a signal narrower, its spectrum becomes broader and lower[3]. Conversely, when we make a signal wider, its spectrum becomes narrower and higher. For example, consider a rectangle function of duration T[4]. Using the similarity theorem:

[3] To ensure energy is conserved.

[4] Where $a = 1/T$.

$$\mathrm{rect}\left(\frac{t}{T}\right) \longleftrightarrow T\,\mathrm{sinc}(fT). \tag{7.11}$$

This relationship is plotted in Figure 7.1 for $T = 1$ and $T = 2$. What do you think happens in the limit as $T \to \infty$?

2.3 *Time-domain convolution*

The Fourier transform converts a convolution to a multiplication:

$$x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)\mathrm{d}\tau \longleftrightarrow X(f)H(f). \tag{7.12}$$

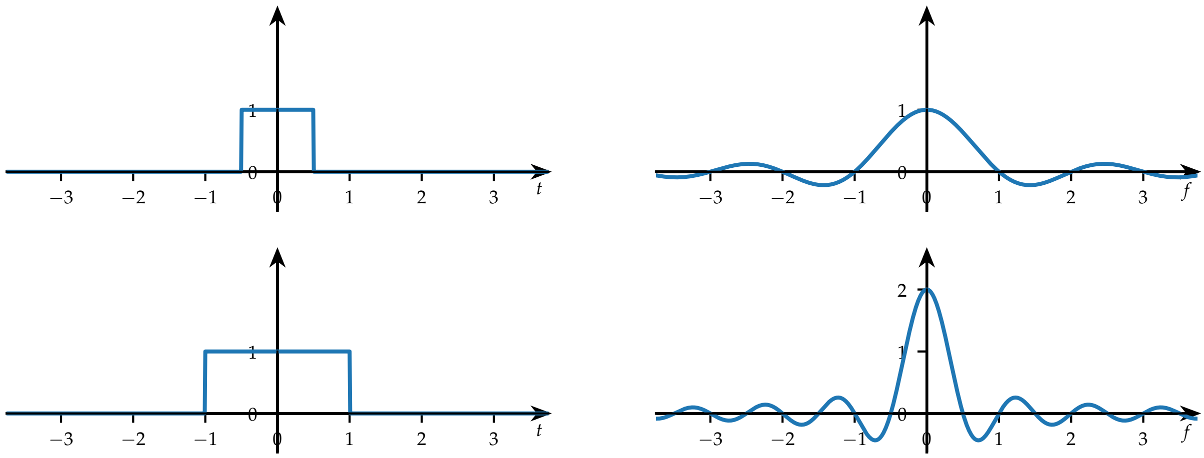


Figure 7.1: The similarity theorem demonstrated with the rect function and its Fourier transform, the sinc function.

Let's consider the case

$$y(t) = x(t) * \delta(t). \tag{7.13}$$

From the convolution integral:

$$y(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)\mathrm{d}\tau. \tag{7.14}$$

Since $\delta(t-\tau) = 0$ when $t \neq \tau$, then

$$y(t) = \int_{-\infty}^{\infty} x(t)\delta(t-\tau)\mathrm{d}\tau. \tag{7.15}$$

Now $x(t)$ can be moved outside the integral,

$$y(t) = x(t)\int_{-\infty}^{\infty} \delta(t-\tau)\mathrm{d}\tau. \tag{7.16}$$

Since the integral of a Dirac delta is 1, then

$$y(t) = x(t). \tag{7.17}$$

Using the convolution theorem,

$$Y(f) = X(f) \times 1, \tag{7.18}$$

where $\delta(t) \longleftrightarrow 1$. Since $Y(f) = X(f)$,

$$y(t) = x(t). \tag{7.19}$$

This is a simple demonstration of the convolution theorem.

2.4 Frequency-domain convolution

Due to the symmetry of the forward and inverse Fourier transforms, there is a similar convolution theorem:

$$x(t)w(t) \longleftrightarrow \int_{-\infty}^{\infty} X(\nu)W(f-\nu)\mathrm{d}\nu = X(f) * H(f). \tag{7.20}$$

For example, consider the function,

$$y(t) = \cos(2\pi f_0 t)\,\mathrm{rect}\left(\frac{t}{T}\right). \tag{7.21}$$

This describes a tone burst of duration T as used by a sonar or radar. This can be considered the product of two signals,

$$x(t) = \cos(2\pi f_0 t), \tag{7.22}$$

$$w(t) = \mathrm{rect}\left(\frac{t}{T}\right). \tag{7.23}$$

These signals have well known Fourier transforms,

$$X(f) = \frac{1}{2}\delta(f+f_0) + \frac{1}{2}\delta(f-f_0), \tag{7.24}$$

$$W(f) = T\,\mathrm{sinc}(fT). \tag{7.25}$$

Using the convolution theorem,

$$Y(f) = X(f) * W(f) = \left(\frac{1}{2}\delta(f+f_0) + \frac{1}{2}\delta(f-f_0)\right) * T\,\mathrm{sinc}(fT). \tag{7.26}$$

As shown in the previous section, convolution with a Dirac delta is easy to determine. In this case, we get,

$$Y(f) = \frac{T}{2}\,\mathrm{sinc}\left((f+f_0)T\right) + \frac{T}{2}\,\mathrm{sinc}\left((f-f_0)T\right). \tag{7.27}$$

This shows that truncation of an AC signal results in spreading of the spectrum.

3 Fourier transform theorems summary

The basic Fourier transform theorems are[5]:

Linearity

$$ax(t) + by(t) \longleftrightarrow aX(f) + bY(f). \tag{7.28}$$

Similarity

$$x(at) \longleftrightarrow \frac{1}{|a|}X\left(\frac{f}{a}\right). \tag{7.29}$$

[5] Most can be derived using a change of variable while doing the Fourier integral.

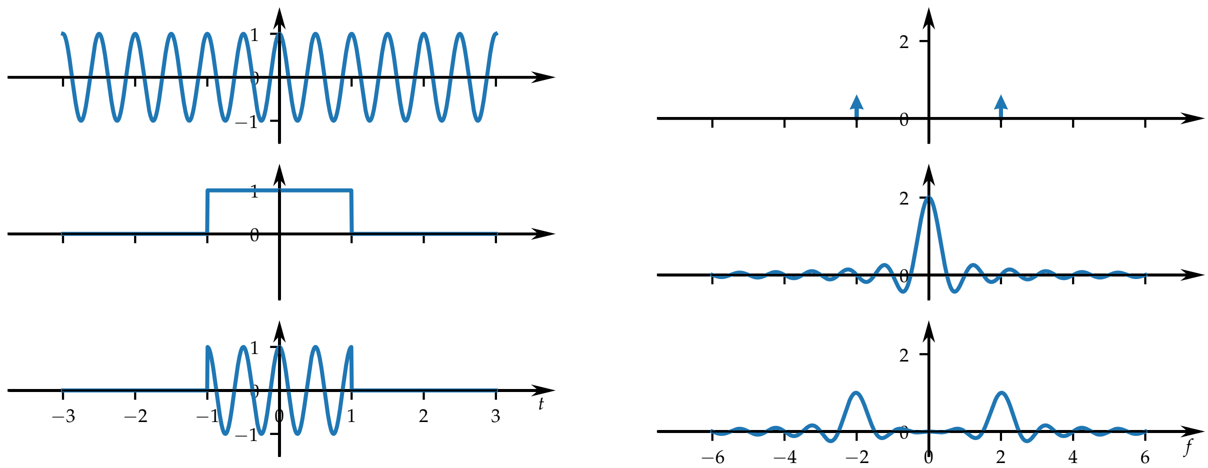


Figure 7.2: Multiplication in the time domain is equivalent to convolution in the frequency domain.

Time-domain convolution

$$x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)\mathrm{d}\tau \longleftrightarrow X(f)H(f). \tag{7.30}$$

Frequency-domain convolution

$$x(t)w(t) \longleftrightarrow \int_{-\infty}^{\infty} X(\nu)W(f-\nu)\mathrm{d}\nu = X(f) * H(f). \tag{7.31}$$

Time shift

$$x(t-\tau) \longleftrightarrow X(f)\exp\left(-\mathrm{j}2\pi f\tau\right). \tag{7.32}$$

Frequency shift [6]

$$x(t)\exp\left(\mathrm{j}2\pi\nu t\right) \longleftrightarrow X(f-\nu). \tag{7.33}$$

[6] This breaks the Hermitian symmetry and creates a complex time signal.

4 *Alternative Fourier transform definition*

There are several definitions of the Fourier transform. A common alternative is

$$X(\omega) = \int_{-\infty}^{\infty} x(t)\exp\left(-\mathrm{j}\omega t\right)\mathrm{d}t. \tag{7.34}$$

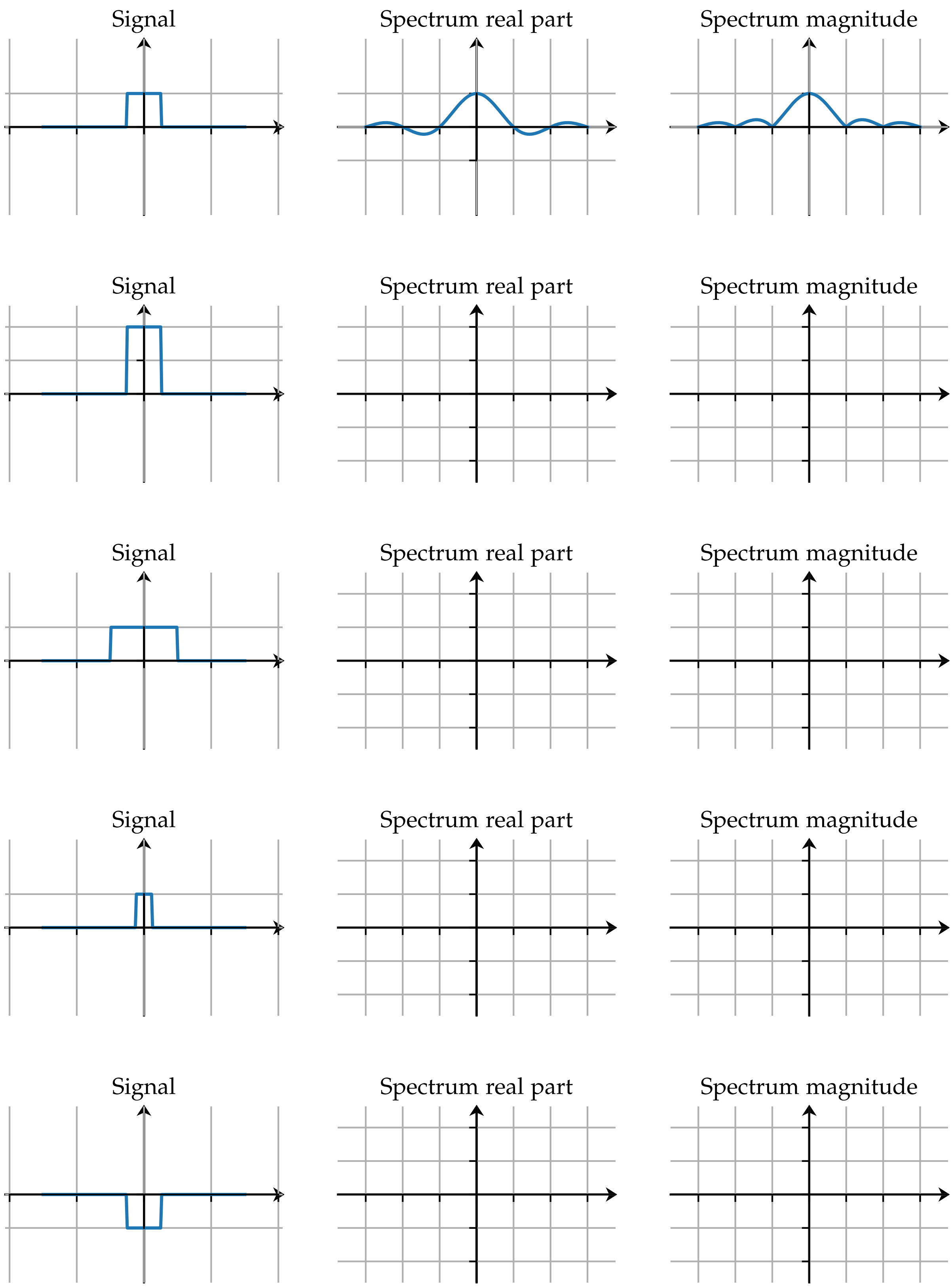
This is easy to derive from (5.1) with $\omega = 2\pi f$. Unfortunately, the inverse transform has an additional scale factor

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)\exp\left(\mathrm{j}\omega t\right)\mathrm{d}\omega. \tag{7.35}$$

This can be derived from (7.1) with $\omega = 2\pi f$ and thus $\mathrm{d}\omega/\mathrm{d}f = 2\pi$ giving $\mathrm{d}f = \mathrm{d}\omega/(2\pi)$.

5 *Exercises*

1. Complete the following Fourier transform pairs:



2. For the following questions, derive the Fourier transform of the specified signals using the common Fourier transform pairs in Table 5.1 and the basic theorems. Also sketch the signals and the magnitude of their spectra.

   (a) $\text{rect}(2t)$.

   (b) $\text{rect}(t/2)$.

   (c) $\text{sinc}(2t)$.

   (d) $\text{sinc}(t/2)$.

   (e) $1 + \cos(2\pi f_0 t)$.

   (f) $\text{rect}(t) + \text{sinc}(t)$.

   (g) $\exp(\mathrm{j}2\pi f_0 t)$. *Hint: use Euler's formula and the linearity theorem.*

   (h) $\delta(t - \tau)$.

   (i) $\text{rect}(t - \tau)$.

   (j) $\text{rect}(t/T)\cos(2\pi f_0 t)$. *Hint: use the convolution theorem.*

3. For the following questions, derive the inverse Fourier transform of the specified signals using the common Fourier transform pairs in Table 5.1 and the basic theorems:

   (a) $\text{rect}(2f)$.

   (b) $\text{rect}(f/2)$.

   (c) $\text{sinc}(2f)$.

   (d) $\text{sinc}(f/2)$.

   (e) $\delta(f - 2) + \delta(f + 2)$.

   (f) $\delta(f - 2) - \delta(f + 2)$.

   (g) $\delta(f - 2)$. *Hint: use the linearity theorem and the previous two problems.*

8
Oscilloscope II

Modern scopes are based around a high-speed ADC, typically sampling at 1 GHz with 8-bit resolution. The samples are stored in a circular memory buffer. When a trigger event is detected, the samples on either side of the trigger time are displayed.

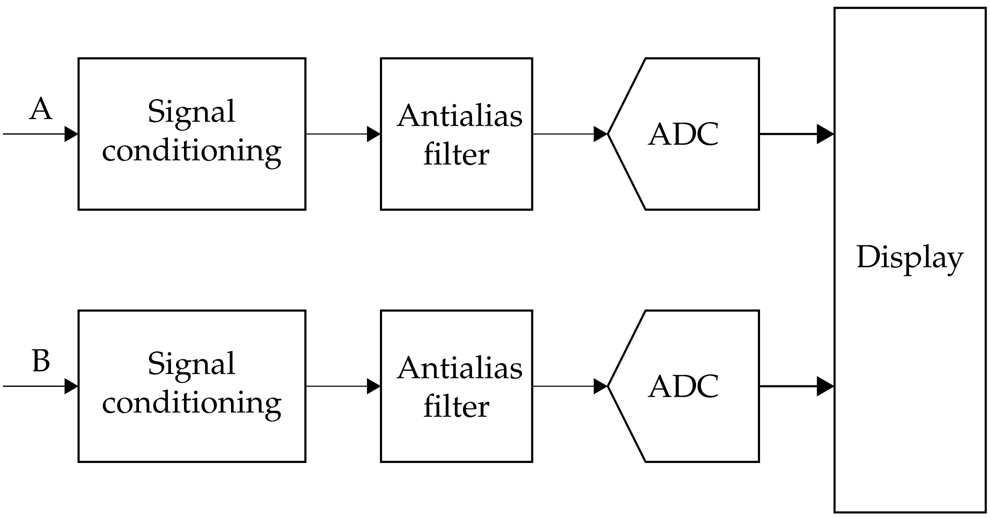


Figure 8.1: System diagram of a digital scope.

1 Input channels

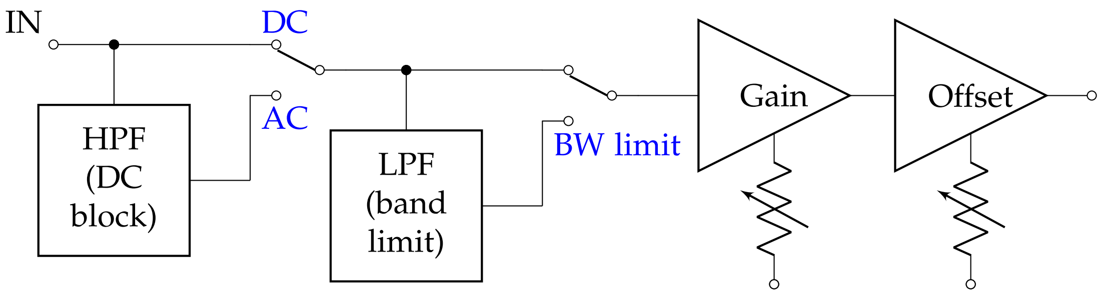


Figure 8.2: Simplified system diagram of a digital scope input channel.

Scopes typically have two or four input channels. Each channel has a number of features:

1. AC/DC coupling (AC coupling selects a high-pass filter so that small signals can be viewed in the presence of large DC levels).

2. Gain

3. Offset

4. Bandwidth limit (this selects a low-pass filter[1], typically 20 MHz to reduce the noise bandwidth).

[1] By default the only other lowpass filtering is an antialias filter prior to the ADC with a typical bandwidth of 100 MHz.

2 Horizontal timebase

The input channels control the vertical display of the waveforms. The horizontal timebase controls the duration of the signal displayed on the screen.

The display has a finite number of pixels and so there is a tradeoff between the signal duration and the time resolution. Even though the input signal may be adequately sampled by the ADC, the displayed signal can be aliased[2].

[2] There are not enough screen pixels to show fast events.

3 Probes

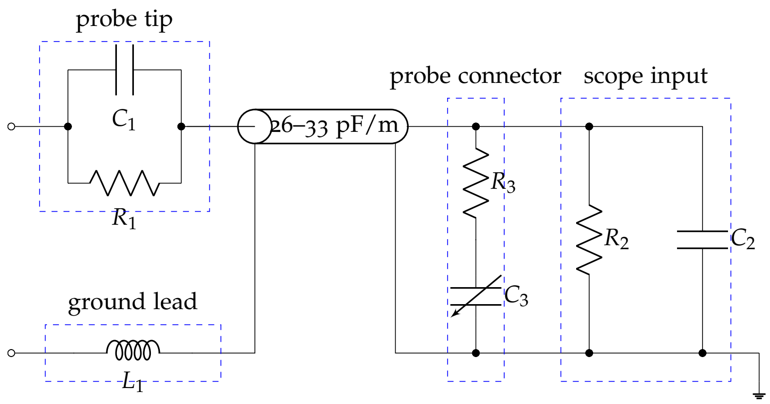


Figure 8.3: Schematic of scope probe, cable, and input. $R_1 = 9\,\mathrm{M\Omega}$, $R_2 = 1\,\mathrm{M\Omega}$, $C_1 = 8$–$12\,\mathrm{pF}$, $C_2 = 20\,\mathrm{pF}$, $R_3 = 500\,\Omega$, $C_3 = 7$–$50\,\mathrm{pF}$, $L_1 \approx 60\,\mathrm{nH}$ depending on loop area.

Each probe has a ground lead that must be connected close to the measurement point. If this is not connected, a large 50 Hz signal will be measured (see Figure 6.4).

The loop created by the ground lead has significant inductance and limits the maximum bandwidth of the scope. In particular, it makes it hard to accurately measure the fast switching signals produced by high-speed digital circuits.

Oscilloscope cables have a high capacitance and thus measuring a circuit can alter its operation. The probe capacitance can make an unstable circuit stable and vice versa!

It is best to use 10× probes for high-frequency or pulse measurement since they have a higher bandwidth and less capacitance[3].

[3] 500 MHz and 9-15 pF for 10× compared to 20 MHz and 50–100 pF for 1×.

3.1 10× probe

The scope cables have a high capacitance and so a 10× probe uses a capacitive voltage divider to reduce the input capacitance. So while it is called 10× it is really an attenuator[4].

The transfer function for a capacitive divider[5] is

$$H(s) = \frac{V_o(s)}{V_i(s)}, \tag{8.1}$$

$$= \frac{\frac{1}{sC_2}}{\frac{1}{sC_1} + \frac{1}{sC_2}}, \tag{8.2}$$

$$= \frac{C_1}{C_1 + C_2}. \tag{8.3}$$

[4] With old scopes you multiplied the measurement on the screen by 10 to get the correct value. These days scopes can sense the type of probe and do this automatically.

[5] Note, that this transfer function appears to be frequency independent but a capacitive divider does not work at DC.

So to divide by 10 requires $C_1 = C_2/9$.

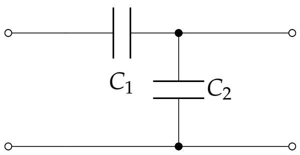


Figure 8.4: Schematic of capacitive voltage divider.

The input capacitance between the probe tip and ground is

$$C_{\text{in}} = \frac{C_1 C_2}{C_1 + C_2}. \tag{8.4}$$

So with $C_1 = C_2/9$ then $C_{\text{in}} = C_2/10$.

To work at DC and to increase the input resistance, a capacitor/resistor divider is used as shown in Figure 8.5.

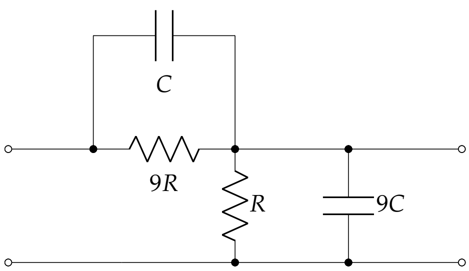


Figure 8.5: Schematic of 10× oscilloscope probe. Its transfer function is $H(s) = 1/10$.

4 Triggering modes

normal A trigger event is generated when the trigger source exceeds a preset trigger level[6]. This mode is useful for capturing transient events or non-repetitive signals such as UART waveforms.

auto A trigger event is automatically generated if a trigger event has not occurred for some time. This mode is useful for DC and AC waveforms and is selected by autoset.

[6] Normal triggering can be confusing when there is no trigger event. Some scopes have a force trigger button to refresh the screen to see what is happening.

4.1 Normal trigger setup

1. Select normal mode triggering.
2. Select trigger source (usually the input channel you are measuring).
3. Select the trigger edge (rising, falling, both).
4. Adjust the trigger level (trigger events below this level will be ignored).
5. Set the trigger holdoff to ignore other possible trigger events following the desired trigger event.

4.2 Auto trigger setup

1. Select auto mode triggering.
2. Select the trigger source (usually the channel you are measuring).

5 Sampling considerations

A scope ADC is typically 8-bit giving 256 quantisation steps over the full dynamic range. Thus very small signals require amplification if they are smaller than the size of the quantisation voltage.

The effective number of bits is increased by oversampling and averaging.

Scope manufacturers specify the sampling rate and memory size; these are for all the channels. So a two-channel scope that states 2 GSa/s and 100 KSa of memory means that each channel samples at 1 GSa/s and there is 50 MSa per channel.

To record long duration signals the scope needs to throw away[7] samples.

[7] These may be averaged first.

6 Miscellaneous features

- Most scopes allow the measured waveforms to be stored on a flash drive, either as an image or as a data file of the sample data.
- Signals can be averaged to reduce noise.
- Single trigger mode stops ignores all but the first trigger event is detected.

7 Exercises

1. When is the bandwidth limit option on an input channel useful?
2. How many bytes (octets) of memory are required to store one second of 8-bit data when sampling at 1 Gsamples/s?
3. When is AC coupling on an input channel useful?
4. Why is the autoscale (a.k.a autoset) button evil?
5. What is the scale factor of a 10× probe?
6. What is the main advantage of a 10× probe?
7. Does the ground clip need to be connected?
8. Give two examples for when normal triggering is useful.
9. What is the purpose of trigger holdoff? When might you use it?

9 Analogue filters

Filters attenuate unwanted frequency components in a signal. In electronics, analogue filters are used for

- anti-aliasing before an analogue-to-digital converter
- reconstruction after a digital-to-analogue converter
- radio-frequency applications
- high-power applications

Analogue transducers can also be considered as filters since the transfer function is never perfect. For example, a woofer loudspeaker is hopeless for reproducing high frequencies.

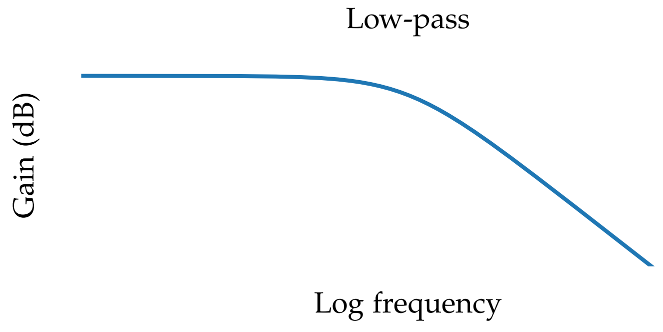
Analogue filter design is also useful for mechanical systems; for example, a car suspension system is designed to suppress unwanted oscillations.

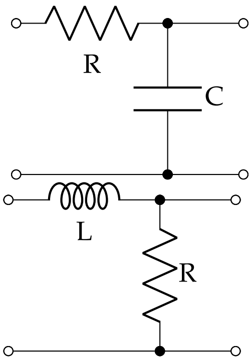
1 Filter examples

The main types of filter are low-pass, high-pass, and band-pass. Other filter types include band-stop, notch, and all-pass[1].

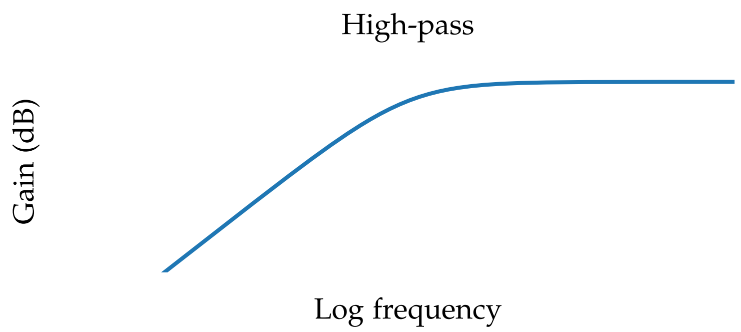
[1] This may have a different phase response.

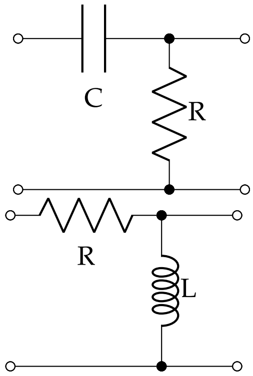
1.1 Low-pass filter



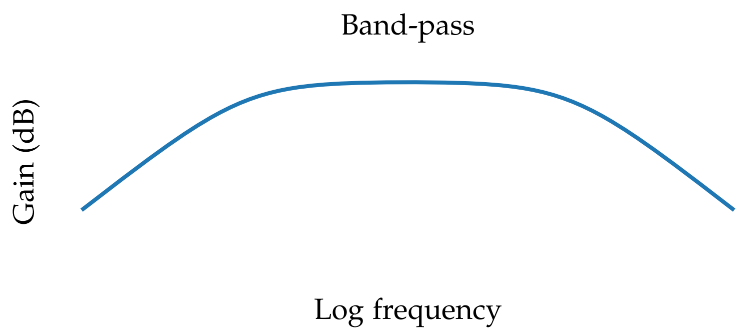


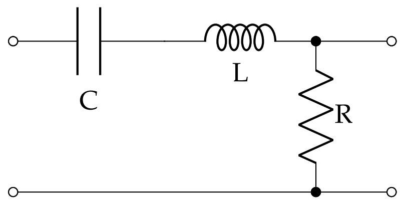
1.2 *High-pass filter*



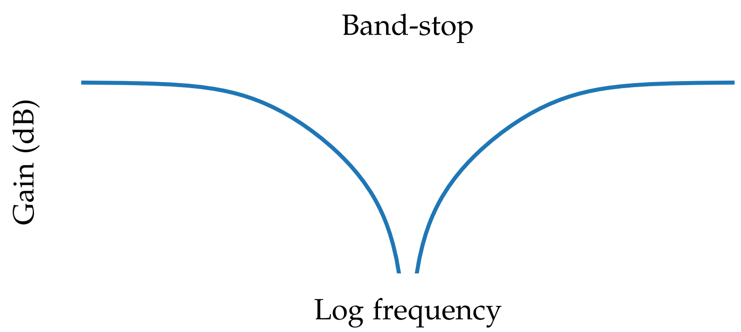


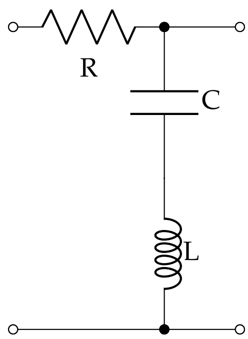
1.3 *Band-pass filter*

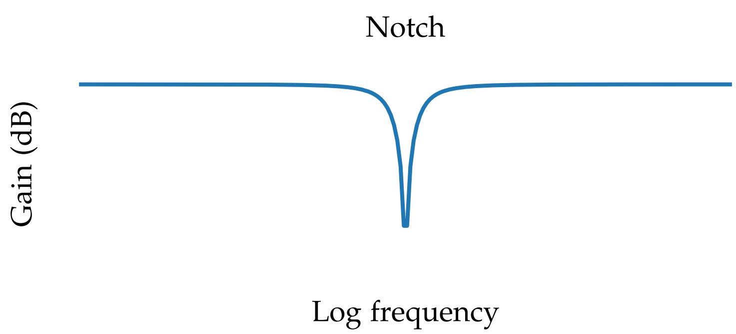




1.4 *Band-stop filter*







2 *Frequency response*

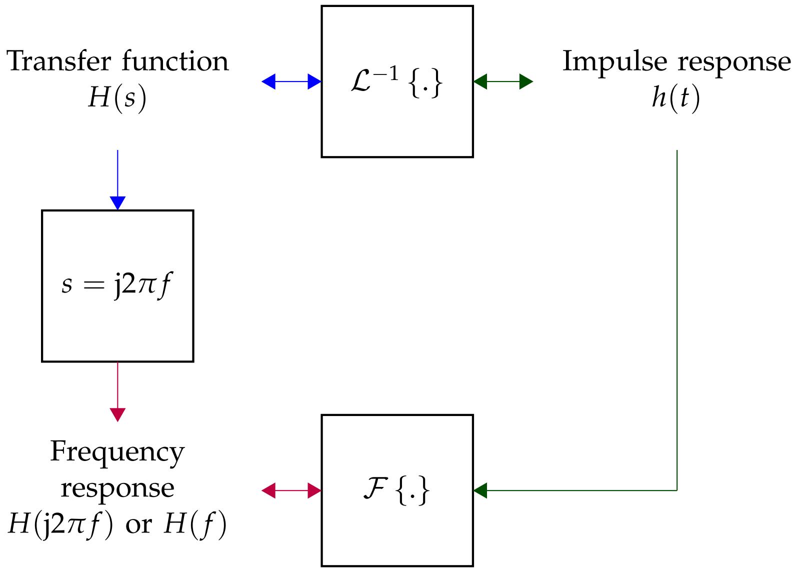


Figure 9.1: Relationship between transfer function, impulse response, and frequency response for a causal system where $h(t) = 0$ for $t < 0$.

The frequency response of a linear time invariant (LTI)[2], causal system can be found two ways:

1. Setting[3] $s = \mathrm{j}\pi f$ in the Laplace domain transfer function to yield $H(\mathrm{j}2\pi f)$.
2. Taking a Fourier transform of the impulse response, $h(t)$, to yield $H(f)$.

[2] Time-invariant means the transfer function does not change with time. For example, an electrical filter with constant component values.

[3] This is equivalent to setting, σ, (the real part of s) to zero.

Note, there are two notations for the same thing[4]:

1. $H(\mathrm{j}2\pi f)$ using Laplace analysis,
2. $H(f)$ using Fourier analysis.

[4] Warning, warning, warning! H has an identity crisis.

This notational gnarliness can be confusing and you have to guess from context[5].

[5] Where there is ambiguity I use the subscript f to denote the Fourier transform version.

3 *Fourier-domain filter analysis*

In the Fourier domain, the filter frequency response relates the input and output spectra:

$$Y(f) = H(f)X(f). \tag{9.1}$$

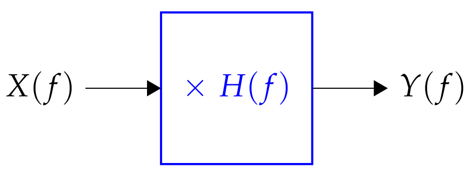
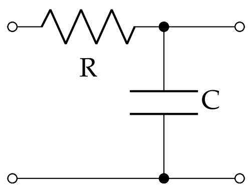


Figure 9.2: Filter analysis in the Fourier domain.

4 Filter examples

4.1 First-order low-pass filter



Let's consider a first-order low-pass filter with an s-domain transfer function,

$$H(s) = \frac{\alpha}{s+\alpha}. \tag{9.2}$$

This response can be created with a capacitor and a resistor where $\alpha = 1/(RC)$. Taking an inverse Laplace transform gives the impulse response,

$$h(t) = \alpha \exp(-\alpha t)\, u(t). \tag{9.3}$$

As expected, this impulse response is causal[6]. It has a Fourier transform,

$$\begin{aligned}
H_f(f) &= \int_{-\infty}^{\infty} h(t) \exp(-\mathrm{j}2\pi f t)\,\mathrm{d}t, && (9.4)\\
&= \alpha \int_{0}^{\infty} \exp(-\alpha t) \exp(-\mathrm{j}2\pi f t)\,\mathrm{d}t, && (9.5)\\
&= \alpha \int_{0}^{\infty} \exp\left(-\left(\mathrm{j}2\pi f + \alpha\right) t\right)\mathrm{d}t, && (9.6)\\
&= \alpha \left[\frac{\exp\left(-\left(\mathrm{j}2\pi f + \alpha\right) t\right)}{-(\mathrm{j}2\pi f + \alpha)}\right]_0^{\infty}, && (9.7)\\
&= \frac{\alpha}{\mathrm{j}2\pi f + \alpha}. && (9.8)
\end{aligned}$$

[6] Indeed, any analogue filter you can create must be causal, i.e., $h(t) = 0$ for $t < 0$.

Note, this can be derived by replacing $s = \mathrm{j}2\pi f$ in the Laplace transform.

$$H_f(f) = H_s(\mathrm{j}2\pi f) = \frac{\alpha}{\mathrm{j}2\pi f + \alpha}. \tag{9.9}$$

4.2 Second-order bandpass filter

A second-order bandpass filter has a transfer-function

$$H(s) = \frac{2\zeta\omega_0 s}{s^2 + 2\zeta\omega_0 s + \omega_0^2}. \tag{9.10}$$

Since it is causal, its Fourier transform is

$$\begin{aligned}
H_f(f) &= H(\mathrm{j}2\pi f), && (9.11)\\
&= \frac{\mathrm{j}2\zeta(2\pi)^2 f_0 f}{-(2\pi f)^2 + \mathrm{j}4\pi\zeta f_0 f + (2\pi f_0)^2}, && (9.12)\\
&= \frac{\mathrm{j}2\zeta(2\pi)^2 f_0 f}{(2\pi)^2\left(f_0^2 - f^2\right) + \mathrm{j}4\pi\zeta f_0 f}. && (9.13)
\end{aligned}$$

This can be converted into polar form (magnitude and phase) with some tedious algebra.

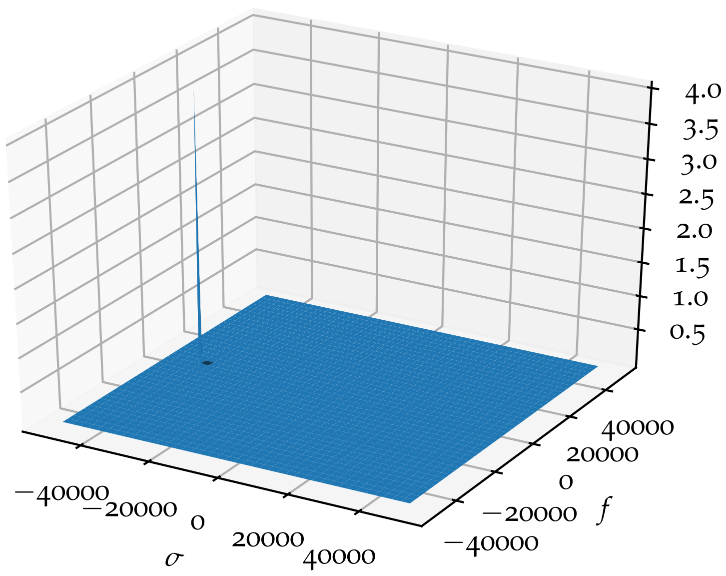


Figure 9.3: A surface plot of $|H(s)|$ for first-order low-pass filter where $H(s) = \alpha/(s+\alpha)$. This plot is difficult to show because of the singularity at the pole. The infinite value has been clipped.

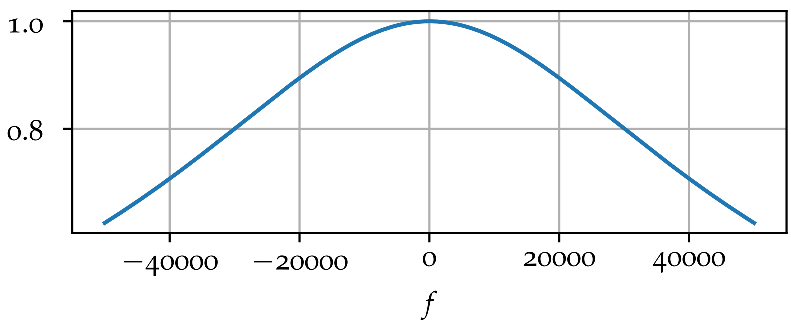


Figure 9.4: A plot of $\left|H_f(f)\right|$ for a first-order low-pass filter. This is a slice of $|H(s)|$ for $s = \mathrm{j}2\pi f$.

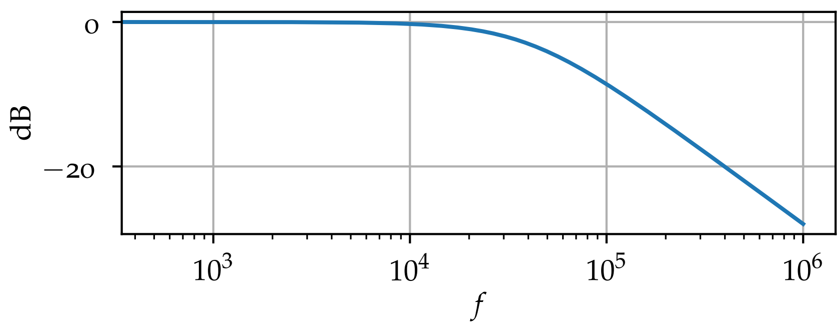


Figure 9.5: A plot of $\left|H_f(f)\right|$ in dB with a logarithmic frequency axis for a first-order low-pass filter.

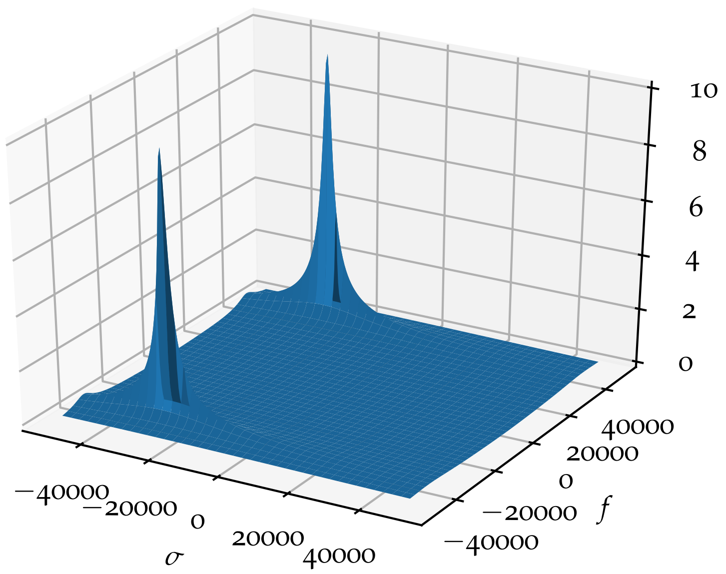


Figure 9.6: A surface plot of $|H(s)|$ for a second-order band-pass filter, where $s = \sigma + j2\pi f$, $f_0 = 40\,\text{kHz}$ and $\zeta = 0.1$. f_0 controls the centre frequency and ζ controls the bandwidth. This plot is difficult to show because of the singularities at the two poles. The infinite values have been clipped.

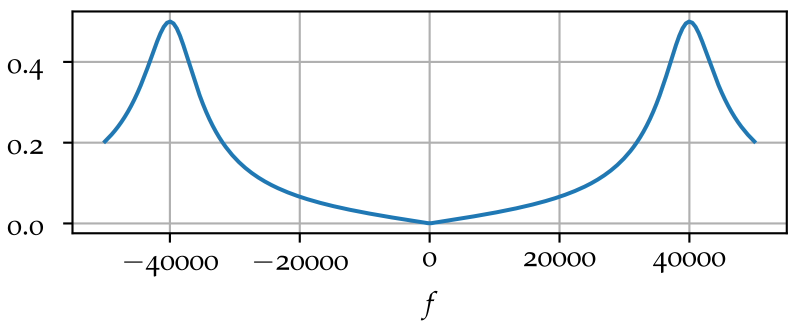


Figure 9.7: A plot of $\left|H_f(f)\right|$ for a second-order band-pass filter. This is a slice of $|H(s)|$ for $s = j2\pi f$.

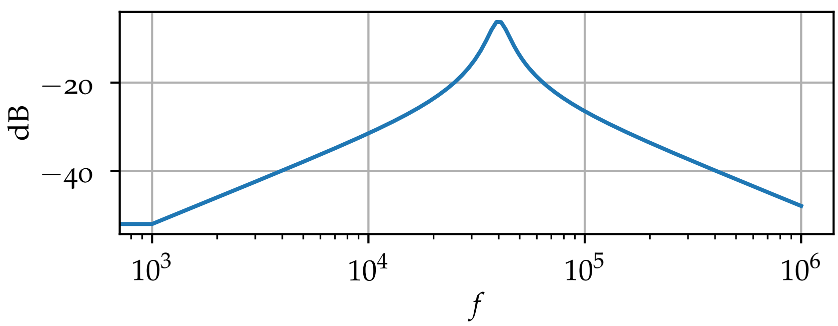


Figure 9.8: A plot of $\left|H_f(f)\right|$ in dB with a logarithmic frequency axis for a second-order band-pass filter.

4.3 The brick-wall filter

A low-pass brick-wall filter[7] has the frequency response:

$$H(f) = \text{rect}\left(\frac{f}{B}\right). \tag{9.14}$$

The corresponding impulse response is

$$h(t) = B\,\text{sinc}(Bt). \tag{9.15}$$

However, this is *non-causal* and cannot be realised.

[7] This is the ultimate filter for anti-aliasing; unfortunately it cannot be realised.

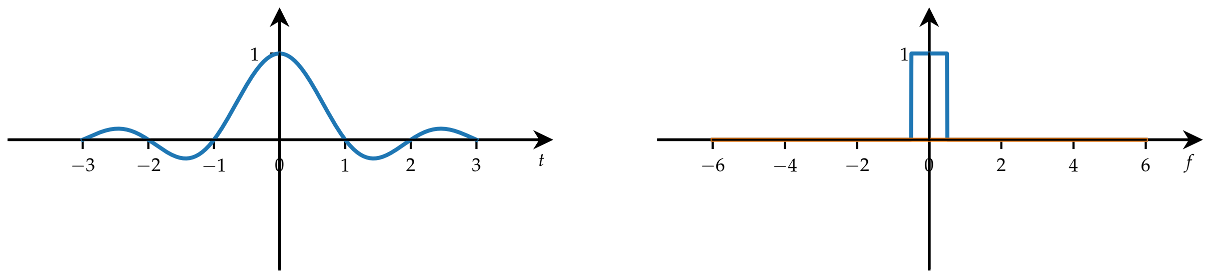


Figure 9.9: The impulse response of a brick-wall function is a sinc function.

4.4 Differentiator

Differentiators and integrators are linear time-invariant filters. For example, consider a differentiator:

$$y(t) = \frac{\mathrm{d}}{\mathrm{d}t}x(t). \tag{9.16}$$

Taking Laplace transforms gives

$$Y(s) = sX(s). \tag{9.17}$$

Thus the s-domain transfer function is

$$H(s) = \frac{Y(s)}{X(s)} = s, \tag{9.18}$$

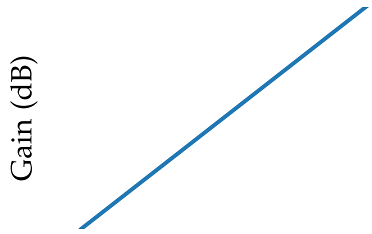
and so the Fourier domain frequency response is

$$H_f(f) = \mathrm{j}2\pi f = 2\pi f \exp\left(\mathrm{j}\frac{\pi}{2}\right). \tag{9.19}$$

The multiplication by j introduces a phase shift of 90 degrees. Note, the multiplication by f removes the time-invariant dc component. Unfortunately, this operation also amplifies high frequency noise. The impulse response of a differentiator can be shown to be the derivative of the Dirac delta,

$$h(t) = \frac{\mathrm{d}}{\mathrm{d}t}\delta(t), \tag{9.20}$$

which is causal as assumed.



4.5 *Integrator*

An integrator is defined as

$$y(t) = \int_{-\infty}^{t} x(\tau) \mathrm{d}\tau. \tag{9.21}$$

This can be expressed as the convolution of the signal, $x(t)$, with a Heaviside step[8],

$$y(t) = \int_{-\infty}^{\infty} u(t-\tau) x(\tau) \mathrm{d}\tau, \tag{9.22}$$

and thus the filter impulse response is the Heaviside step,

$$h(t) = u(t). \tag{9.23}$$

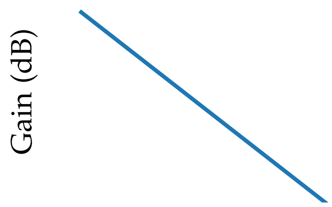
The s-domain transfer function for an integrator is

$$H(s) = \frac{1}{s}, \tag{9.24}$$

and since the filter is causal, the Fourier domain frequency response is

$$H_f(f) = \frac{1}{\mathrm{j}2\pi f} = -\mathrm{j}\frac{1}{2\pi f} = \frac{1}{2\pi f} \exp\left(-\mathrm{j}\frac{\pi}{2}\right). \tag{9.25}$$

The multiplication by $-\mathrm{j}$ introduces a phase shift of -90 degrees. Note, the division by f preferentially amplifies low frequencies.



[8] Denoted here as $u(t)$ to avoid confusion with the impulse response, $h(t)$.

$h(t)$	$H(s)$	$H_f(f)$	
$\delta'(t)$	s	$j2\pi f$	differentiator
$\delta(t)$	1	1	all-pass filter
$u(t)$	$\frac{1}{s}$	$\frac{1}{j2\pi f}$	integrator

Table 9.1: Relationships between differentiator and integrator.

5 Time-domain filter analysis

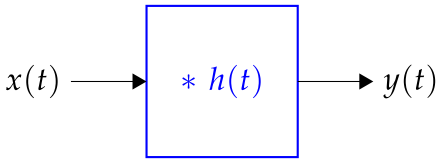


Figure 9.10: Filter analysis in the time domain.

The time-domain response of a LTI filter is described by a convolution[9]

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) h(t-\tau) \mathrm{d}\tau, \tag{9.26}$$

where $h(t)$ is the filter impulse response.

A filter must be causal, this implies that

$$h(t) = 0 \quad t < 0. \tag{9.27}$$

This relaxes the integral limits to

$$y(t) = x(t) * h(t) = \int_{-\infty}^{t} x(\tau) h(t-\tau) \mathrm{d}\tau. \tag{9.28}$$

[9] Convolution is commutative so $x(t) * h(t) = h(t) * x(t)$.

6 Group and phase delay

If a filter with frequency response

$$H_f(f) = |H_f(f)| \exp\left(-j\psi(f)\right), \tag{9.29}$$

is driven with a quasi-sinusoidal signal of the form

$$x(t) = a(t) \cos\left(2\pi f_0 t + \phi\right), \tag{9.30}$$

where $a(t)$ is a slowly varying amplitude, the output signal of the filter has the form,

$$y(t) = |H_f(f)| \, a(t - \tau_g) \cos\left(2\pi f_0 (t - \tau_p) + \phi\right). \tag{9.31}$$

Here τ_p is the phase delay and τ_g is the group delay, where

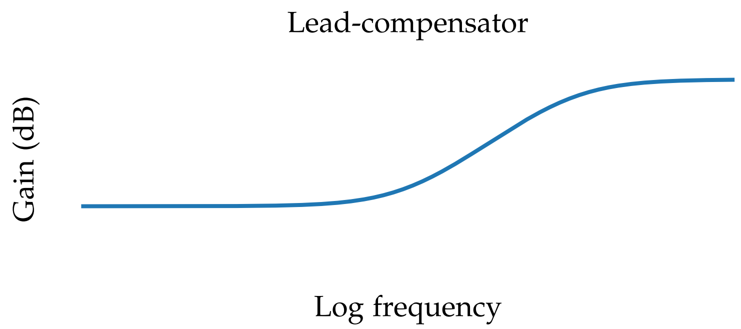
$$\tau_p = -\frac{\psi(f)}{2\pi f}, \tag{9.32}$$

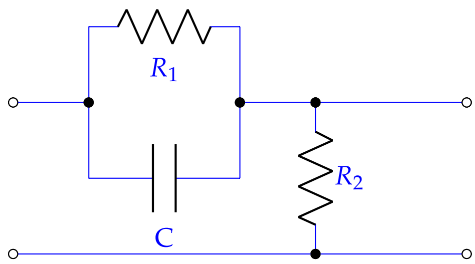
and

$$\tau_g(f) = -\frac{1}{2\pi} \frac{\mathrm{d}}{\mathrm{d}f} \psi(f). \tag{9.33}$$

7 Other filters used in control systems

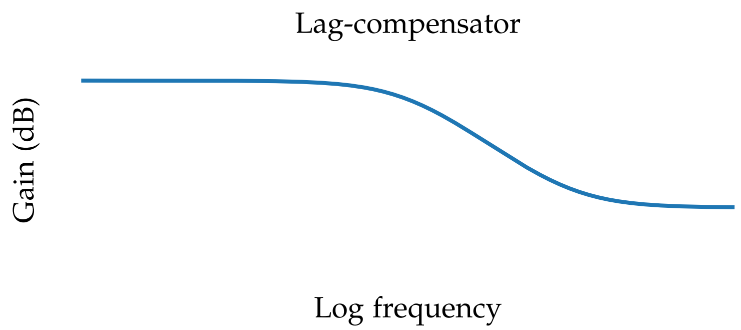
7.1 Lead-compensator

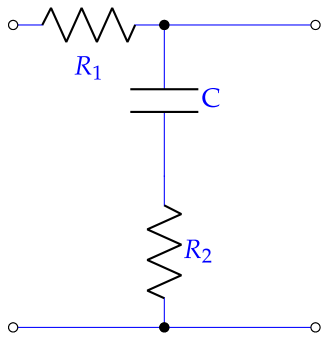




$$H(s) = \frac{s+\beta}{s+\alpha} \quad \beta \ll \alpha. \tag{9.34}$$

7.2 Lag-compensator



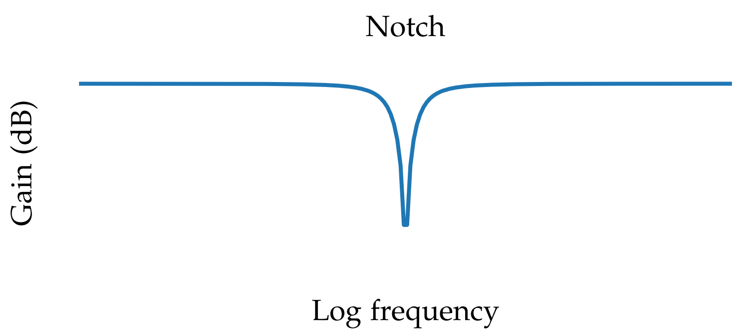


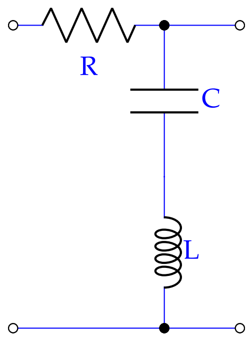
$$H(s) = \frac{s+\beta}{s+\alpha} \quad \beta \gg \alpha. \tag{9.35}$$

7.3 Notch filter

[10]

[10] A notch filter is a special case of a band-stop filter with a low-damping (ζ) and thus a high Q ($1/\zeta$). It is useful for removing mains hum.





$$H(s) = \frac{s^2+\omega_0^2}{s^2+2\zeta\omega_0 s+\omega_0^2} \quad \text{notch / band-stop.} \tag{9.36}$$

8 First-order transfer functions

$$H(s) = \frac{\alpha}{s+\alpha} \quad \text{low-pass.} \tag{9.37}$$

$$H(s) = \frac{s}{s+\alpha} \quad \text{high-pass.} \tag{9.38}$$

9 Second-order transfer functions

Note, in the following, there can be restrictions on the damping factor ζ due to the circuit topology. For example, a second order filter using an inductor and a capacitor is more flexible than a filter using two capacitors.

$$H(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \quad \text{low-pass.} \tag{9.39}$$

$$H(s) = \frac{s^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \quad \text{high-pass.} \tag{9.40}$$

$$H(s) = \frac{2\zeta\omega_0 s}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \quad \text{band-pass.} \tag{9.41}$$

$$H(s) = \frac{s^2 + \omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \quad \text{band-stop.} \tag{9.42}$$

10 Fourier transform from Laplace transform

The Fourier-domain frequency response of a filter can be easily determined from the s-domain transfer function by replacing s with $\mathrm{j}2\pi f$,

$$H_f(f) = H(\mathrm{j}2\pi f). \tag{9.43}$$

Here $H(s)$ is the s-domain transfer function of the filter and $H_f(f)$ is its corresponding Fourier-domain transfer function. Note, this only applies for causal impulse responses where $h(t) = 0$ for $t < 0$.

To prove (9.43), let's recap the two transforms. The Laplace transform[11] of a function $h(t)$ is

$$H(s) = \int_{0_-}^{\infty} h(t) \exp(-st)\, \mathrm{d}t, \tag{9.44}$$

and the Fourier transform[12] is

$$H_f(f) = \int_{-\infty}^{\infty} h(t) \exp(-\mathrm{j}2\pi f t)\, \mathrm{d}t. \tag{9.45}$$

[11] The Laplace transform is more general than the Fourier transform since it exists for a larger variety of functions.

[12] I've used the subscript f here to differentiate the Fourier transform of $h(t)$ from the Laplace transform of $h(t)$.

If $h(t)$ is causal, i.e., $h(t) = 0$ for $t < 0$, then the unilateral Laplace transform is a special case of the bilateral Laplace transform,

$$H(s) = \int_{-\infty}^{\infty} h(t) \exp(-st) \, \mathrm{d}t. \tag{9.46}$$

If we compare (9.46) with (9.45) we find that

$$H_f(f) = H(\mathrm{j}2\pi f). \tag{9.47}$$

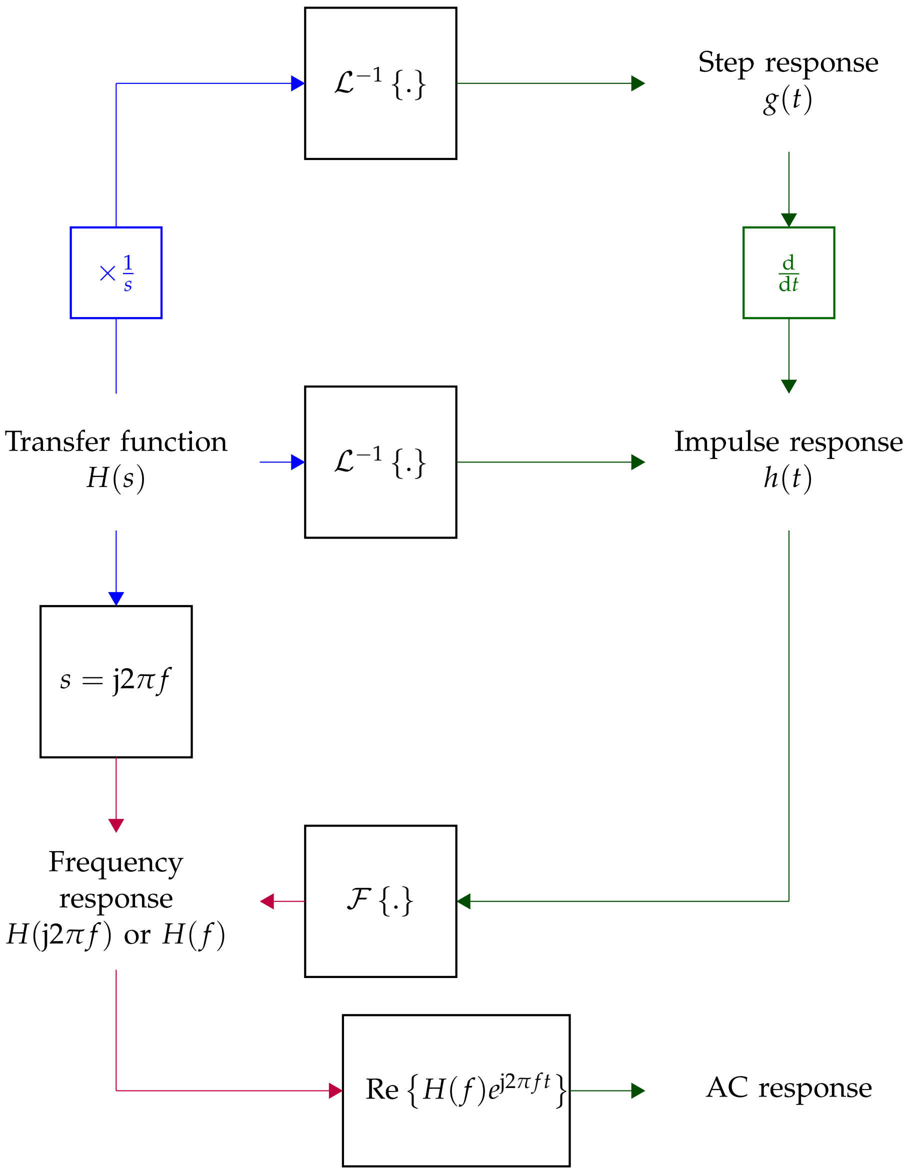


Figure 9.11: Relationship between transfer function, impulse response, step response, and frequency response for a causal system.

11 Exercises

1. Given the following s-domain transfer functions, $H(s)$, determine the corresponding Fourier-domain frequency responses, $H_f(f)$. For example, if $H(s) = s$, then $H_f(f) = H(\mathrm{j}2\pi f) = \mathrm{j}2\pi f = 2\pi f \exp(\mathrm{j}\pi/2)$.

   (a) $\frac{1}{s}$.
   (b) $\frac{1}{s+a}$.
   (c) $\frac{s}{s+a}$.

2. Consider the signal

$$x(t) = \cos(2\pi t) + \cos(1000\pi t) \tag{9.48}$$

   input to a filter with an s-domain transfer function

$$H(s) = \frac{0.5}{s+0.5}. \tag{9.49}$$

   (a) Determine the magnitude of the frequency response of the filter, $|H_f(f)|$.
   (b) Determine the 3 dB break frequency.
   (c) Sketch the magnitude of the frequency response of the filter.
   (d) Sketch the Fourier transform of the signal, $X(f)$.
   (e) Determine how much the higher frequency component is attenuated compared to the lower frequency component.
   (f) Sketch the approximate Fourier transform of the output of the filter.
   (g) Sketch the approximate output of the filter.
   (h) Consider the similar transfer function:

$$H(s) = \frac{1}{s+0.5}, \tag{9.50}$$

   and explain why it cannot be achieved with a passive filter.

3. A first order RC anti-aliasing filter is required with a (3 dB) break frequency of 20 kHz.

   (a) Sketch the circuit.
   (b) Determine the s-domain transfer function.
   (c) Determine the magnitude of the Fourier domain transfer function.

(d) What resistor value is required to achieve the desired break frequency given a 1 nF capacitor.

(e) Determine how much attenuation the filter provides at 2 MHz.

(f) If more attenuation was required at 2 MHz, what would you do?

4. Consider a high-pass filter comprised of a 1 kΩ resistor and a 1 nF capacitor.

   (a) Sketch the circuit.

   (b) Determine the s-domain transfer function.

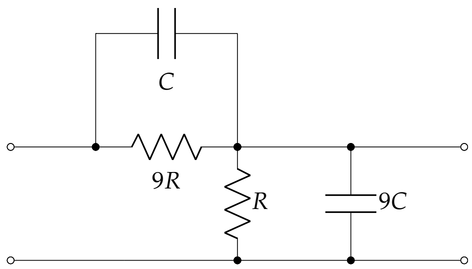
   (c) Determine the break frequency.

   (d) Sketch the magnitude of the frequency response of the circuit.

   (e) For frequencies well below the break frequency, determine the phase shift of the filter.

   (f) For frequencies well below the break frequency, does this circuit act like an integrator or a differentiator?

5. For the following circuit:



   (a) Determine the s-domain transfer function. If this is too x tedious, try ignoring the capacitors first. Then repeat the problem by ignoring the resistors. *Hint: this circuit is used for* $10\times$ *oscilloscope probes.*

   (b) Determine the magnitude and phase of the Fourier domain transfer function.

   (c) Determine what kind of filter it is.

10
Digital filters

Most filters are digital filters. They have many uses; for example,

1. Reducing noise level
2. Reducing interfering signals, e.g., mains hum
3. Performing differentiation
4. Performing integration
5. Image edge detection
6. Image enhancement

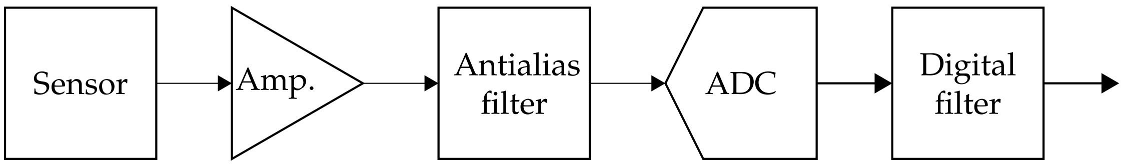


Figure 10.1: Signal processing chain.

1 *Discrete-time noise processes*

All signal measurements are corrupted by noise. By definition, noise is random, and so it cannot be predicted. However, the statistical behaviour of noise can be modelled.

A discrete-time noise signal is a realisation[1] of a discrete-time *noise process*, for example[2],

$$w[n] = \{w[n]\} = \{w_0, w_1, w_2, \cdots, w_{N-1}\}. \qquad (10.1)$$

The corresponding noise process is simply a sequence of random variables,

$$W[n] = \{W[n]\} = \{W_0, W_1, W_2, \cdots, W_{N-1}\}. \qquad (10.2)$$

Each random variable is characterised by its probability distribution function, $f_{W_n}(w)$.

[1] Also called sample function.

[2] Note, **both the notations $w[n]$ and $\{w[n]\}$ are used to denote a sequence of values for a discrete-time signal**. $w[n]$ can also denote a single value so $\{w[n]\}$ is less ambiguous but more cumbersome. In practice, $w[n]$ is more common but whether it denotes a sequence or a value must be determined from the context.

1.1 *Additive white Gaussian noise*

The most common noise process is additive white Gaussian noise (AWGN). Here each of the random variables are independent and have the same Gaussian (normal) distribution[3]

$$W_n \sim \mathcal{N}(0, \sigma), \tag{10.3}$$

where σ^2 is the noise variance.

[3] The random variables are said to be i.i.d. (independent and identically distributed). They have zero mean and the same variance.

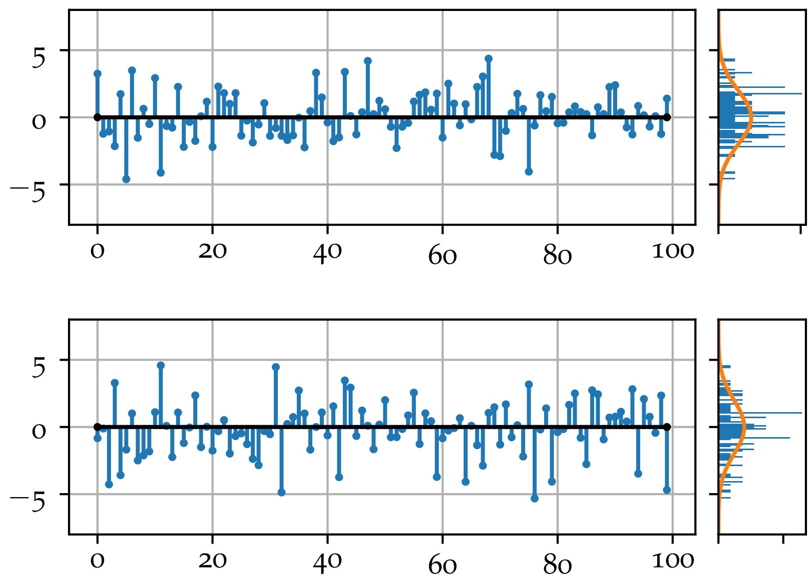


Figure 10.2: Two realisations of a zero-mean Gaussian random process.

2 *Noise filtering example*

Consider the signal shown in Figure 10.3. This is a constant value corrupted by additive white Gaussian noise (AWGN), typical of any sensor output sampled with an ADC. The histogram has the shape of a Gaussian[4].

[4] Most sensor noise has a Gaussian probability density function.

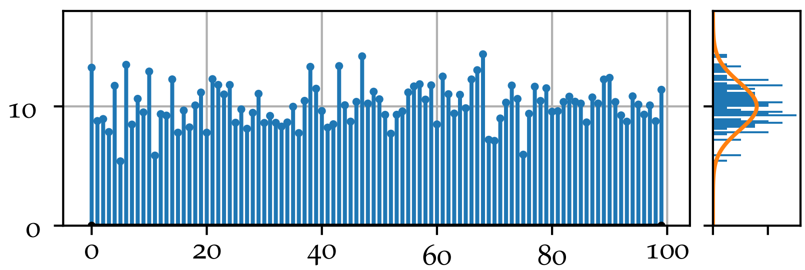


Figure 10.3: A dc signal with additive zero-mean Gaussian noise. Note the discrete histogram due to integer sample values.

3 *Moving average filters*

From the histogram it appears that the underlying sensor value is 10 but how can we obtain this from the signal? The obvious solution is to perform some averaging. But what if the mean value from the sensor is also changing? One solution is a moving average filter.

3.1 *Two sample moving average filter*

Consider a simple moving average formed from two consecutive samples of the input signal, $x[n]$,

$$y[n] = \frac{1}{2}\left(x[n] + x[n-1]\right). \tag{10.4}$$

If this filter is applied to the sequence of samples:

$$\{13.2, 8.8, 8.9, 7.9, 11.7, 5.4, 13.5, 8.5, 10.6, 9.5\}$$

we get a slightly smoother output:

$$\{6.6, 11.0, 8.9, 8.4, 9.8, 8.6, 9.4, 11.0, 9.6, 10.1\}$$

This sequence is plotted in Figure 10.4.

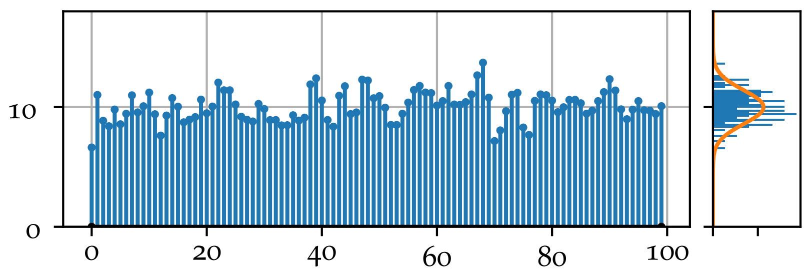


Figure 10.4: A dc signal with additive zero-mean Gaussian noise filtered with a two sample moving average filter.

3.2 *Four sample moving average filter*

Let's now increase the filter size to four samples:

$$y[n] = \frac{1}{4}\left(x[n] + x[n-1] + x[n-2] + x[n-3]\right). \tag{10.5}$$

The resultant sequence is now even smoother:

$$\{3.3, 5.5, 7.7, 9.7, 9.3, 8.5, 9.6, 9.8, 9.5, 10.5\}$$

This sequence is plotted in Figure 10.5.

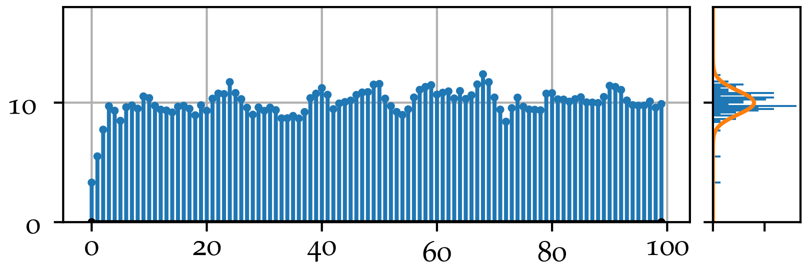


Figure 10.5: A dc signal with additive zero-mean Gaussian noise filtered with a four sample moving average filter.

3.3 *M sample moving average filter*

In general, an M sample moving average filter can be expressed as

$$y[n] = \frac{1}{M} \sum_{m=0}^{M-1} x[n-m]. \tag{10.6}$$

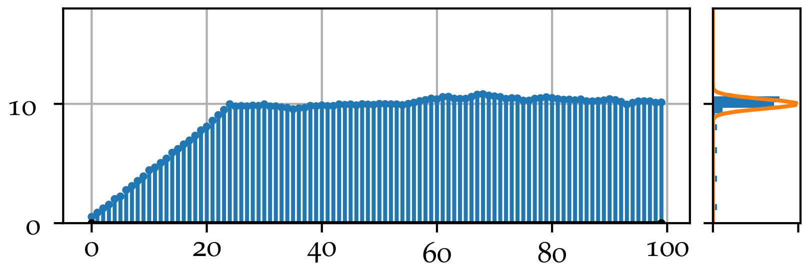


Figure 10.6: A dc signal with additive zero-mean Gaussian noise filtered with a 25 sample moving average filter.

3.4 *Noise reduction*

Consider the average of M independent random variables, X_m, each with a mean[5], μ_X, and variance, σ_X^2. The result, Y, is also a random variable,

[5] The statistics are stationary since both the mean and variance are constant.

$$Y = \frac{1}{M} \sum_{m=0}^{M-1} X_m, \tag{10.7}$$

with a mean

$$\mu_Y = \mu_X, \tag{10.8}$$

and variance

$$\sigma_Y^2 = \frac{1}{M} \sigma_X^2. \tag{10.9}$$

Thus the averaging reduces the noise standard deviation by $\sqrt{M}$.

3.5 *Transient response*

All filters have a delay. For example, a moving average filter takes $M - 1$ samples to reach steady-state.

The transient response[6] also occurs due to a change of the mean signal value. For example, see Figure 10.8.

[6] Of M samples.

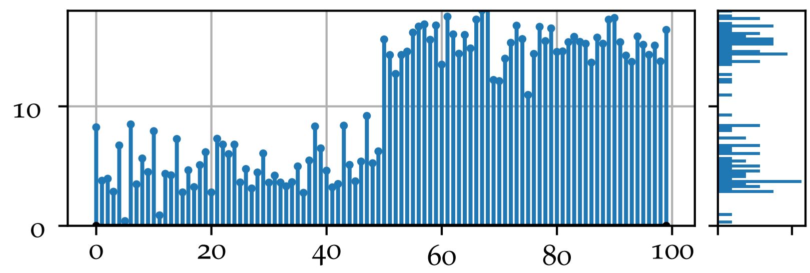


Figure 10.7: A noisy signal with a step change in the mean value.

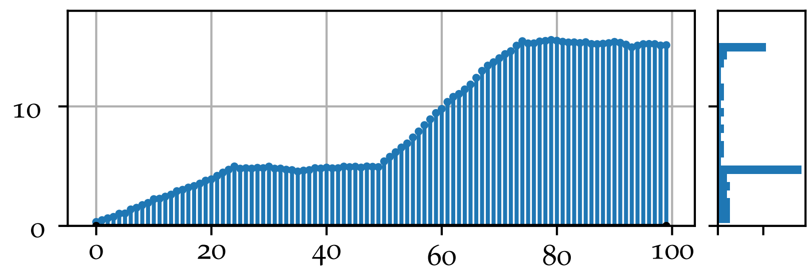


Figure 10.8: A noisy signal filtered with a 25 sample moving average filter.

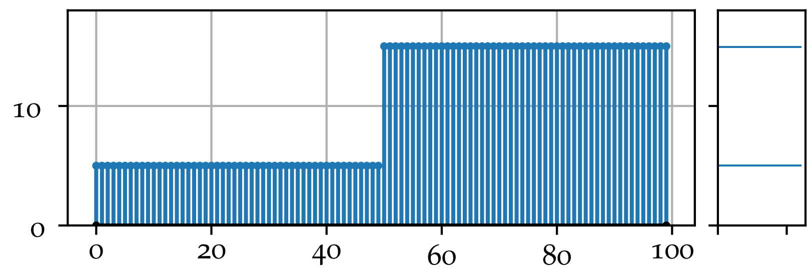


Figure 10.9: A signal with a step change.

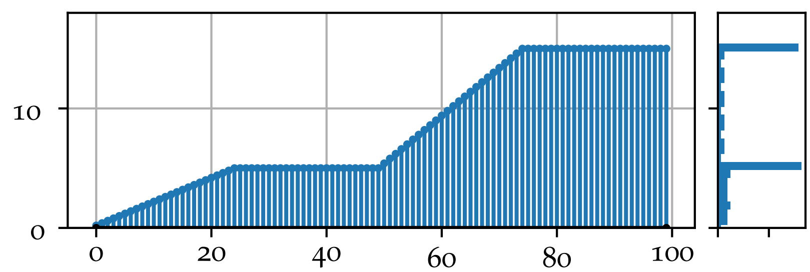


Figure 10.10: The response of a 25 sample moving average filter due to a step change.

4 Recursive filters

A different type of filter to the moving average (MA) filter is the autoregressive (AR) filter (also known as a recursive filter).

4.1 First-order recursive filter

Consider the filter[7] defined by the recurrence relation:

$$y[n] = \alpha y[n-1] + (1-\alpha)x[n]. \qquad (10.10)$$

This is a low-pass filter. α is called the smoothing factor[8]; it sets the break frequency.

The response of this filter to a constant signal corrupted by AWGN is shown in Figure 10.11.

[7] This is sometimes called an exponential filter (or single pole recursive filter).

[8] For a time constant τ, $\alpha = \tau/(\tau + \Delta t)$ where Δt is the sampling interval.

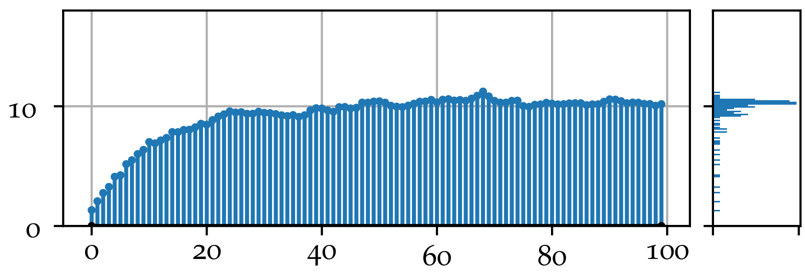


Figure 10.11: A noisy signal and its histogram after being filtered with a first-order recursive filter with $\alpha = 0.9$. Note the transient response due to the initial conditions.

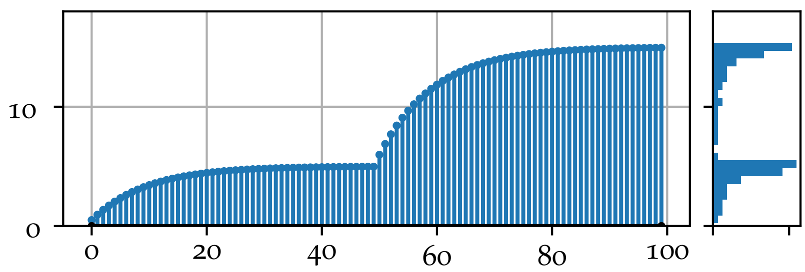


Figure 10.12: The response of a first-order recursive filter ($\alpha = 0.9$) due to a step change. Note the transient response due to the step change. The input signal is $x[n] = 5 + 10u[n-50]$.

5 *Filter implementations*

```c
void setup()
{
    // Initialise sensor
}

void loop()
{
    int value;
    int filtered_value;

    value = analogRead(sensor_pin);

    filtered_value = moving_average(value);

    // Do something with filtered_value ...
}
```

Listing 10.1: Example C code using a filter.

5.1 *Naïve moving average filter*

This example naïvely shifts the previous samples in a buffer and sums the buffer for each new sample.

```c
#define FILTER_SIZE 20

long moving_average_filter(int value)
{
    static int buffer[FILTER_SIZE];
    static byte index = 0;
    long sum = 0;
    int i;

    // Shift and sum previous values in buffer.
    sum = 0
    for (i = FILTER_SIZE - 1; i > 0; i--)
    {
        buffer[i] = buffer[i - 1];
        sum += buffer[i];
    }
    buffer[0] = value;
    sum += value;

    // This uses integer division for speed.
    return sum / FILTER_SIZE;
}
```

Listing 10.2: Example C code to implement a naïve moving average filter.

5.2 *Moving average filter*

This example maintains the current average and then incrementally modifies it for each new value[9].

```
#define FILTER_SIZE 20

int moving_average_filter(int value)
{
    static int buffer[FILTER_SIZE];
    static byte index = 0;
    static byte count = 0;
    static long sum = 0;

    sum -= buffer[index];
    buffer[index] = value;
    sum += value;
    index++;

    if (index >= FILTER_SIZE)
        index = 0;

    if (count < FILTER_SIZE)
        count++;

    // This uses integer division for speed.
    return sum / count;
}
```

Listing 10.3: Example C code to implement a moving average filter.

[9] http://playground.arduino.cc/Main/RunningAverage has an example of an Arduino running average (a.k.a. moving average) library.

5.3 *First-order recursive filter*

```
#define ALPHA 0.9

double first_order_recursive_filter(int value)
{
    static double previous = 0;

    previous = previous * ALPHA + value * (1 - ALPHA);
    return previous;
}
```

Listing 10.4: Example C code to implement a first-order recursive filter. Note, this uses floating-point which is slow on most small microcontrollers.

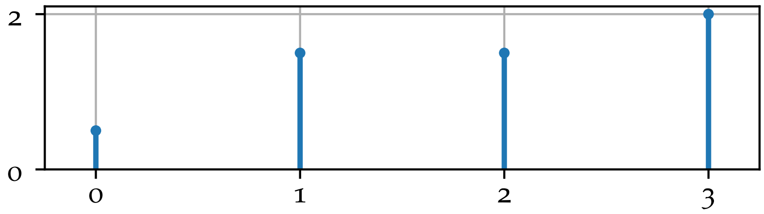
6 Exercises

1. Consider the sequence $\{\underline{1}, 2, 1, 3\}$, i.e., $x[0] = 1$, $x[1] = 2$, $x[2] = 1$, etc., and answer the following questions:

   (a) Sketch the lollipop (stem) plot for this sequence.

   (b) Confirm the lollipop plot below if the sequence is filtered by the filter described by:

$$y[n] = \frac{1}{2}\left(x[n] + x[n-1]\right). \quad (10.11)$$



   (c) How long does this filter take to reach the steady-state response.

   (d) If Gaussian noise[10] with independent samples is presented to the filter, by what factor is the noise standard deviation reduced by?[11]

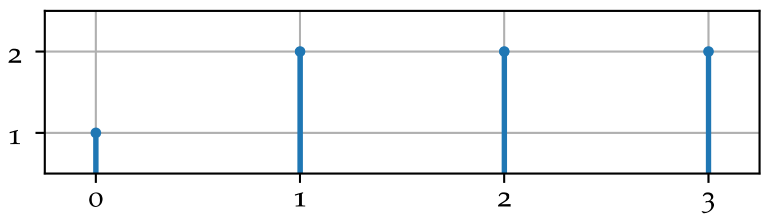
   (e) If a step change is applied to this filter, how many samples does the transient response last for?

2. Consider the sequence $\{\underline{1}, 3, 5, 7\}$ and answer the following questions:

   (a) Sketch the lollipop plot for this sequence.

   (b) Confirm the lollipop plot below if the sequence is filtered by the filter described by:

$$y[n] = x[n] - x[n-1]. \quad (10.12)$$



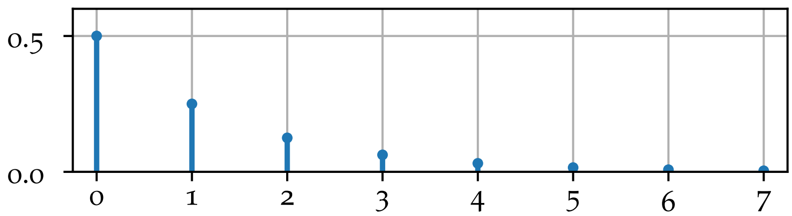
   (c) How long does this filter take to reach the steady-state response.

   (d) What does this filter do?

[10] Noise with a Gaussian (normal) distribution.

[11] Hint: determine the variance of $Z = (X+Y)/2$ where X and Y are random variables with the same variance, σ^2.

3. Consider the sequence $\{\underline{1}, 0, 0, 0, 0, 0, 0, 0\}$ and answer the following questions:
   (a) Sketch the lollipop plot for this sequence.
   (b) Confirm the lollipop plot below if the sequence is filtered by the filter described by:
   $$y[n] = 0.5y[n-1] + 0.5x[n]. \tag{10.13}$$
   Assume $y[n] = 0$ for $n < 0$.



   (c) Why do you think this response is called the impulse response of the filter?
4. The unilateral z-transform of a sequence $x[n]$ is defined by
   $$X(z) = \sum_{n=0}^{\infty} x[n]z^{-n}. \tag{10.14}$$
   (a) Determine the z-transform of $ax[n] + by[n]$.
   (b) Determine the z-transform of $x[n-1]$. *Hint: use a change of variable.*
   (c) Determine the z-transform of $0.7x[n] + 0.3x[n-1]$.
   (d) If $y[n] = 0.7y[n-1] + 0.3x[n]$ show that
   $$Y(z) = \frac{0.3}{1 - 0.7z^{-1}} X(z). \tag{10.15}$$
5. Consider the filter described by:
   $$y[n] = ay[n-1] + 0.5(1+a)x[n] - 0.5(1+a)x[n-1]. \tag{10.16}$$
   (a) What is the response of the filter to a step signal?
   (b) What is the response of the filter to an impulse signal?
   (c) What sort of filter is this?

11
Digital filter frequency response

Filters are used to reduce unwanted signals such as noise and interference from the measured signal leaving the desired signal. However, if we are not careful, a filter will reduce the desired signal. For example, consider Figure 11.1 where the effect of a moving average filter on different frequencies is shown.

There are two things to note:

1. There is a transient response at the start[1] while the filter reaches the steady-state response.
2. The amplitude of the AC signal reduces as the frequency increases but in a non-obvious manner.

[1] Of $M - 1$ samples; in this case 24 samples.

So while a filter may do a good job at reducing the noise, it can also reduce the desired signal! To determine how the filter behaves with frequency, we need to consider the frequency response of the filter. This is found using a discrete-time Fourier transform[2].

[2] Not to be confused with the discrete Fourier transform.

1 Discrete-time frequency transform

Recall that given the s-domain transfer function of a causal system, $H(s)$, the frequency response can be found using

$$H_f(f) = H(s)|_{s=\mathrm{j}2\pi f} \,. \tag{11.1}$$

The discrete-time Fourier transform (DTFT)[3] of a causal discrete-time system is defined in a similar manner:

[3] The subscript indicates that the spectrum is periodic with period $1/\Delta t$.

$$H_{\frac{1}{\Delta t}}(f) = H(z)|_{z=\exp(\mathrm{j}2\pi f \Delta t)} \,. \tag{11.2}$$

Here $H(z)$ is the z-domain transfer function.

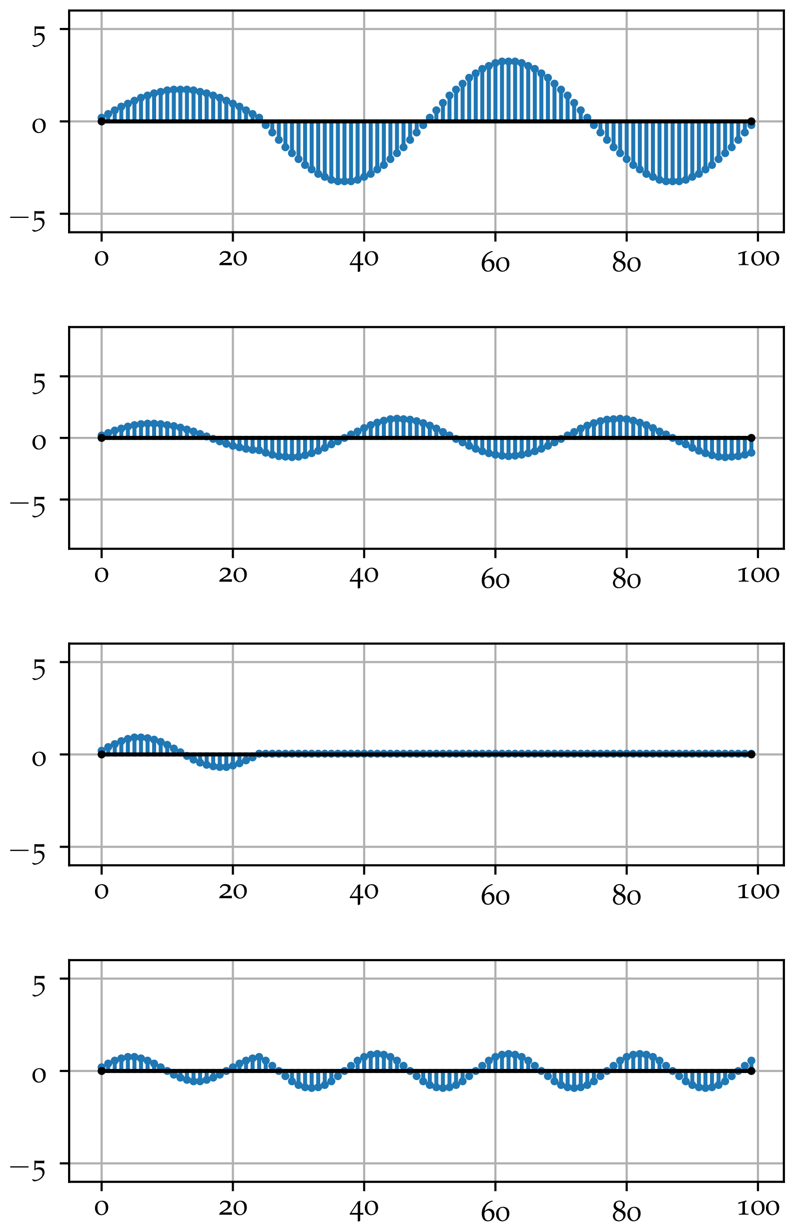


Figure 11.1: AC signals sampled at 1 kHz and filtered by a 25 sample moving average filter. (a) 20 Hz, (b) 30 Hz, (c) 40 Hz, (d) 50 Hz. Note, the AC signals at the start are distorted by the filter transient response.

2 The z-transform

The z-transform is the discrete equivalent to the Laplace transform[4]. Its purpose is to transform a difference equation into an algebraic equation in a similar manner as to how a Laplace transform changes a differential equation into an algebraic equation.

[4] The unilateral Laplace transform is $\int_{0-}^{\infty} h(t)\exp(-st)\,\mathrm{d}t$. Evaluating the integral for $t = n\Delta t$ with $z = \exp(s\Delta t)$ gives the z-transform.

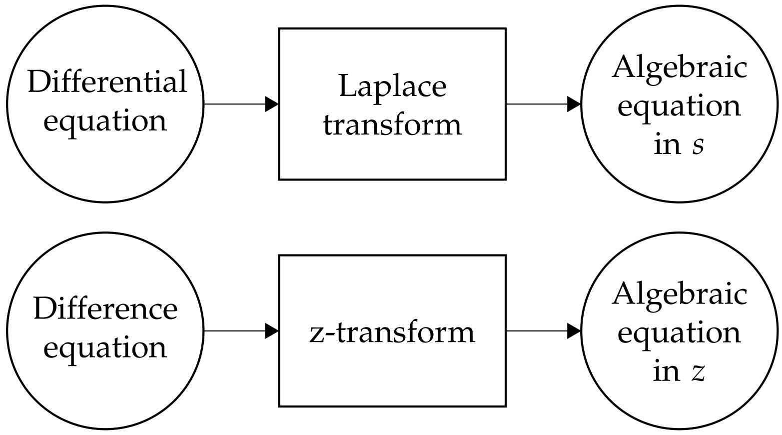


Figure 11.2: Transformation of differential and difference equations into algebraic equations.

The z-transform is defined[5] as:

$$H(z) = \sum_{n=0}^{\infty} h[n] z^{-n}, \tag{11.3}$$

where z is a complex variable, similar to s used for the Laplace transform.

[5] Like the Laplace transform, the z-transform has bilateral and unilateral forms. The unilateral form is used here with the lower limit for n set to 0. The unilateral and bilateral forms give the same answer for causal signals and systems.

2.1 The z-transform theorems

Two of the three important z-transform theorems are[6]:

Linearity

$$a_1 h_1[n] + a_2 h_2[n] \longleftrightarrow a_1 H_1(z) + a_2 H_2(z). \tag{11.4}$$

Time shift

$$h[n-m] \longleftrightarrow H(z) z^{-m}. \tag{11.5}$$

With these two theorems we can simplify the recurrence equations that describe a digital filter.

[6] They are easy to derive by moving constant factors outside the summation in the definition of the z-transform.

3 z-domain transfer function

The z-domain transfer function is defined as

$$H(z) = \left.\frac{Y(z)}{X(z)}\right|_{\mathrm{I.C.}=0}, \tag{11.6}$$

where $X(z)$ is the z-transform of the input signal, $x[n]$, and $Y(z)$ is the z-transform of the output signal, $y[n]$.

3.1 *Example: two sample moving average filter*

A two sample moving average filter is defined as

$$y[n] = \frac{1}{2}\left(x[n] + x[n-1]\right). \tag{11.7}$$

Using the z-transform linearity and time-shift theorems,

$$Y(z) = \frac{1}{2}X(z) + \frac{1}{2}z^{-1}X(z), \tag{11.8}$$

and so the z-domain transfer function is

$$H(z) = \left.\frac{Y(z)}{X(z)}\right|_{\text{I.C.}=0} = \frac{1}{2}\left(1 + z^{-1}\right), \tag{11.9}$$

with a corresponding discrete-time Fourier transform

$$H_{\frac{1}{\Delta t}}(f) = H(z)|_{z=\exp(\mathrm{j}2\pi f\Delta t)} = \frac{1}{2}\left(1 + \exp\left(-\mathrm{j}2\pi f\Delta t\right)\right). \tag{11.10}$$

This can be converted[7] into

$$H_{\frac{1}{\Delta t}}(f) = \exp\left(-\mathrm{j}\pi f\Delta t\right)\cos\left(\pi f\Delta t\right), \tag{11.11}$$

with a magnitude

$$\left|H_{\frac{1}{\Delta t}}(f)\right| = \left|\cos\left(\pi f\Delta t\right)\right|. \tag{11.12}$$

This is plotted in Figure 11.3 and Figure 11.4.

[7] Using $1 + \exp(-\mathrm{j}2\theta) = \exp(-\mathrm{j}\theta)(\exp(\mathrm{j}\theta) + \exp(-\mathrm{j}\theta))$ and $\exp(\mathrm{j}\theta) + \exp(-\mathrm{j}\theta) = 2\cos\theta$.

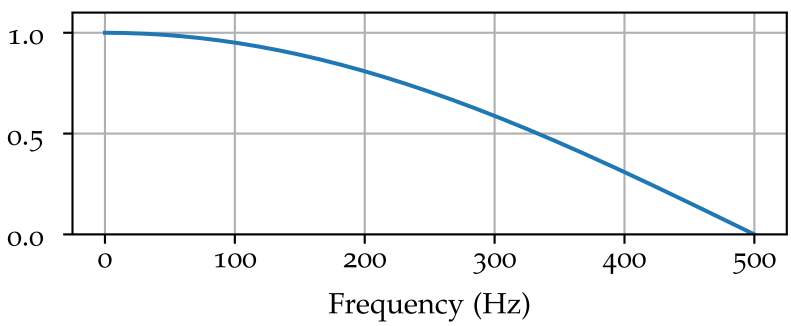


Figure 11.3: Frequency response of a 2 point moving average filter. The sampling frequency is 1 kHz.

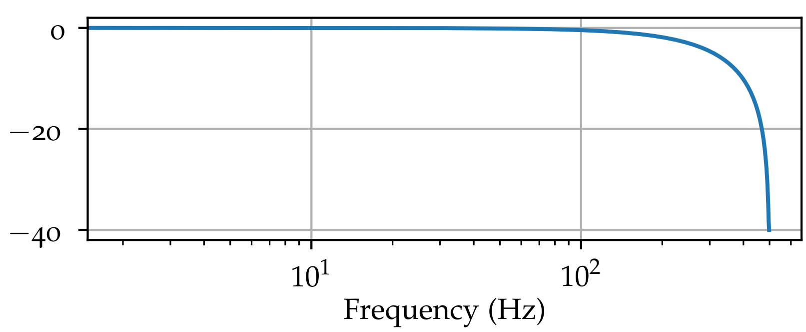


Figure 11.4: Frequency response Bode plot of a 2 point moving average filter. The sampling frequency is 1 kHz.

3.2 *Example: first-order recursive low-pass filter*

A first-order recursive low-pass filter is described by

$$y[n] = \alpha y[n-1] + (1-\alpha)x[n]. \tag{11.13}$$

Using the z-transform linearity and shift theorems,

$$Y(z) = (1-\alpha)X(z) + \alpha z^{-1}Y(z). \tag{11.14}$$

This can be rearranged as

$$\left(1-\alpha z^{-1}\right)Y(z) = (1-\alpha)\,X(z), \tag{11.15}$$

and so the transfer function is

$$H(z) = \left.\frac{Y(z)}{X(z)}\right|_{\text{I.C.}=0} = \frac{1-\alpha}{1-\alpha z^{-1}}, \tag{11.16}$$

with a corresponding discrete-time Fourier transform

$$H_{\frac{1}{\Delta t}}(f) = \frac{1-\alpha}{1-\alpha\exp\left(-j2\pi f\Delta t\right)}. \tag{11.17}$$

The magnitude of the DTFT is

$$\left|H_{\frac{1}{\Delta t}}(f)\right| = \frac{1-\alpha}{\sqrt{1+\alpha^2-2\alpha\cos\left(2\pi f\Delta t\right)}}. \tag{11.18}$$

This is plotted in Figure 11.5 and Figure 11.6.

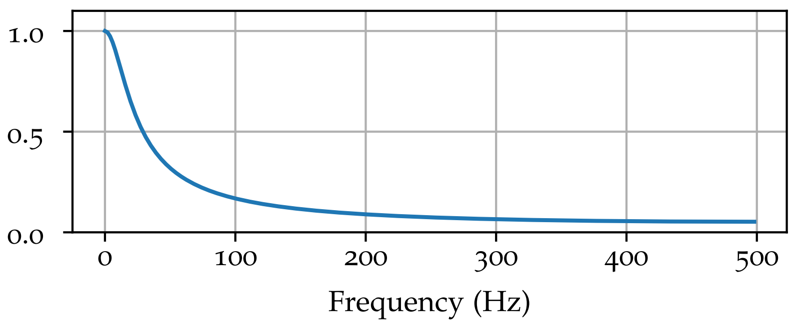


Figure 11.5: Frequency response of first-order low-pass recursive filter. The sampling frequency is 1 kHz.

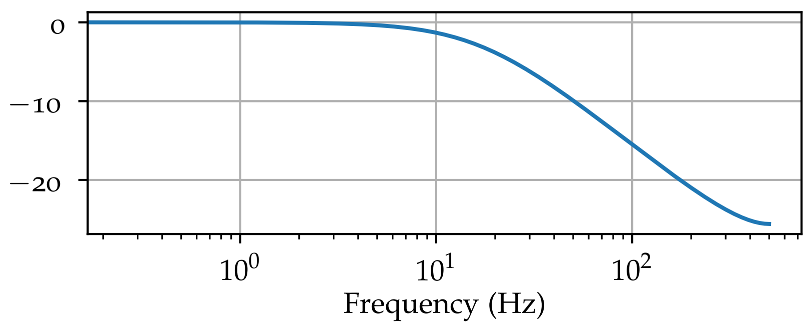


Figure 11.6: Frequency response Bode plot of first-order low-pass recursive filter. The sampling frequency is 1 kHz.

3.3 *Example: differentiator*

A first-order discrete-time differentiator can be implemented using

$$y[n] = \frac{x[n] - x[n-1]}{\Delta t}. \tag{11.19}$$

This has a z-transform,

$$Y(z) = \frac{X(z) - X(z)z^{-1}}{\Delta t}, \tag{11.20}$$

and thus a transform function,

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - z^{-1}}{\Delta t}. \tag{11.21}$$

The resultant DTFT is

$$H_{\frac{1}{\Delta t}}(f) = \frac{1 - \exp\left(-j2\pi f \Delta t\right)}{\Delta t}. \tag{11.22}$$

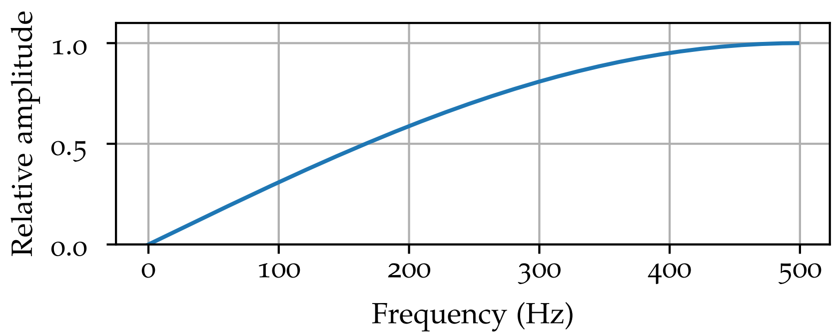


Figure 11.7: Normalised frequency response of differentiator. The sampling frequency is 1 kHz.

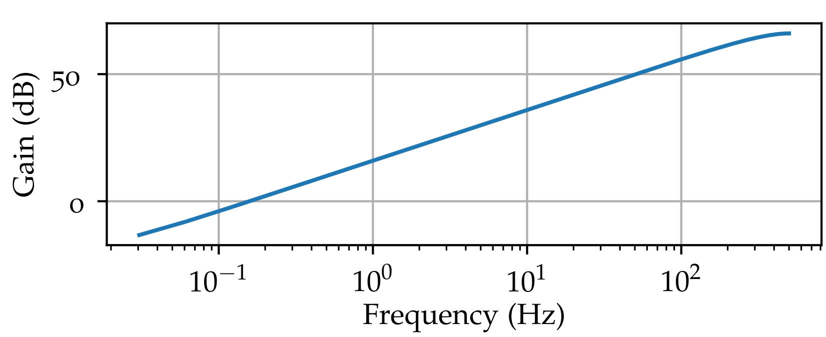


Figure 11.8: Frequency response Bode plot of differentiator. The sampling frequency is 1 kHz.

3.4 *Example: integrator*

A first-order discrete-time integrator can be implemented using

$$y[n] = \sum_{m=0}^{n} x[m]\Delta t. \tag{11.23}$$

This can be expressed by the recursive relation,

$$y[n] = y[n-1] + x[n]\Delta t, \tag{11.24}$$

with a z-transform,

$$Y(z) = Y(z)z^{-1} + X(z)\Delta t, \tag{11.25}$$

and thus a transform function,

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\Delta t}{1 - z^{-1}}. \tag{11.26}$$

The resultant DTFT[8] is

$$H_{\frac{1}{\Delta t}}(f) = \frac{\Delta t}{1 - \exp\left(-\mathrm{j}2\pi f \Delta t\right)}. \tag{11.27}$$

[8] Spot the symmetry with the DTFT for a differentiator.

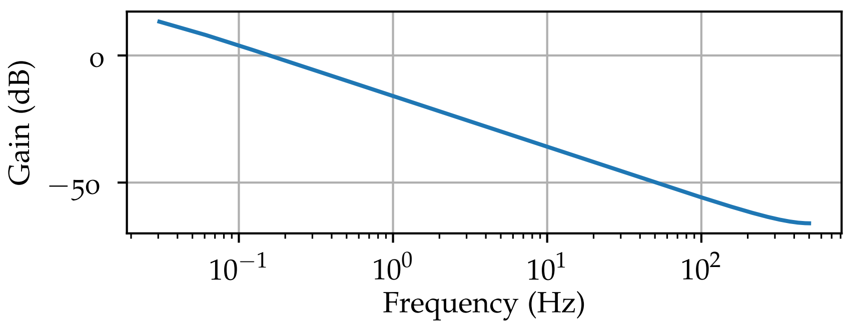


Figure 11.9: Frequency response Bode plot of integrator. The sampling frequency is 1 kHz.

3.5 *Example: M sample moving average filter*

In general, a M sample moving average filter is defined by

$$y[n] = \frac{1}{M}\sum_{m=0}^{M-1} x[n-m] \quad n > M-1. \tag{11.28}$$

This has a z-transform transfer function,

$$H(z) = \frac{1}{M}\sum_{m=0}^{M-1} z^{-m}. \tag{11.29}$$

This is the sum of the first M terms of a geometric series with the known result:

$$H(z) = \frac{1}{M}\frac{1 - z^{-M}}{1 - z^{-1}}. \tag{11.30}$$

From this the frequency response is

$$H_{\frac{1}{\Delta t}}(f) = \frac{1}{M}\frac{1 - \exp\left(-\mathrm{j}2\pi f M\Delta t\right)}{1 - \exp\left(-\mathrm{j}2\pi f\Delta t\right)}. \tag{11.31}$$

The trick to simplifying this is to rewrite it as[9]

[9] With $\theta = \pi f \Delta t$.

$$H_{\frac{1}{\Delta t}}(f) = \frac{1}{M}\frac{\exp\left(-\mathrm{j}M\theta\right)\exp\left(\mathrm{j}M\theta\right) - \exp\left(-\mathrm{j}M\theta\right)\exp\left(-\mathrm{j}M\theta\right)}{\exp\left(-\mathrm{j}\theta\right)\exp\left(\mathrm{j}\theta\right) - \exp\left(-\mathrm{j}\theta\right)\exp\left(-\mathrm{j}\theta\right)}, \tag{11.32}$$

and factoring to give

$$H_{\frac{1}{\Delta t}}(f) = \frac{1}{M}\frac{\exp\left(-\mathrm{j}M\theta\right)\left(\exp\left(\mathrm{j}M\theta\right) - \exp\left(-\mathrm{j}M\theta\right)\right)}{\exp\left(-\mathrm{j}\theta\right)\left(\exp\left(\mathrm{j}\theta\right) - \exp\left(-\mathrm{j}\theta\right)\right)}, \tag{11.33}$$

$$= \frac{1}{M}\exp\left(-\mathrm{j}(M-1)\theta\right)\frac{\sin\left(M\theta\right)}{\sin\left(\theta\right)}. \tag{11.34}$$

The resultant frequency response is

$$H_{\frac{1}{\Delta t}}(f) = \exp\left(-\mathrm{j}\pi(M-1)f\Delta t\right)\frac{\sin\left(\pi M f\Delta t\right)}{M\sin\left(\pi f\Delta t\right)}. \tag{11.35}$$

This magnitude of this is plotted in Figure 11.10[10] and Figure 11.11.

[10] This is similar to a sinc function except that it is periodic.

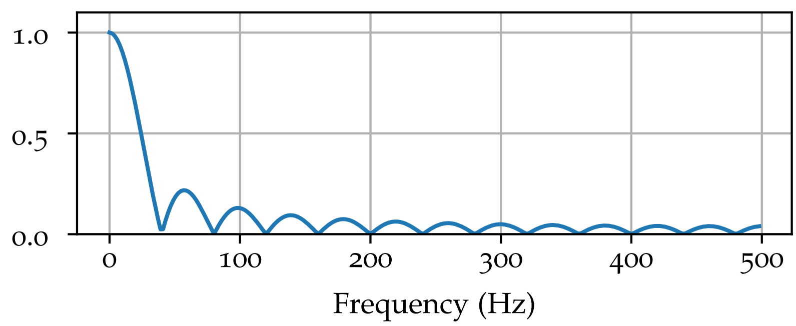


Figure 11.10: Frequency response of a 25 point moving average filter. The sampling frequency is 1 kHz. The nulls are at 40 Hz, 80 Hz, etc.

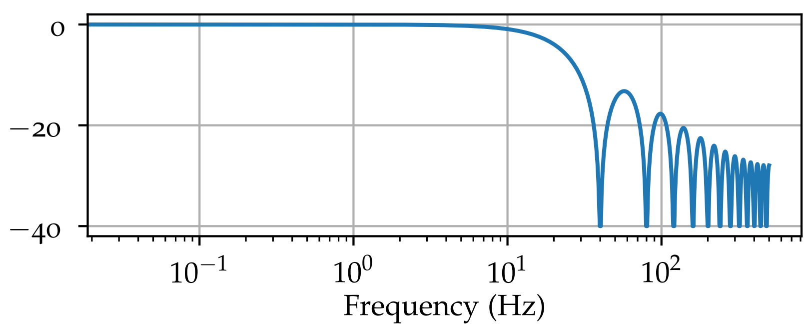


Figure 11.11: Frequency response Bode plot of a 25 point moving average filter. The sampling frequency is 1 kHz.

4 Common digital filters

The difference equations and z-domain transfer functions for some common first-order digital filters are shown in Figure 11.1.

Filter type	Difference equation	Transfer function
low-pass	$y[n] = \alpha y[n-1] + (1-\alpha)x[n]$	$\dfrac{1-\alpha}{1-\alpha z^{-1}}$
high-pass	$y[n] = \alpha y[n-1] + \alpha\left(x[n] - x[n-1]\right)$	$\dfrac{\alpha\left(1-z^{-1}\right)}{1-\alpha z^{-1}}$
differentiator	$y[n] = \dfrac{x[n]-x[n-1]}{\Delta t}$	$\dfrac{1-z^{-1}}{\Delta t}$
integrator	$y[n] = y[n-1] + \Delta t\, x[n]$	$\dfrac{\Delta t}{1-z^{-1}}$

Table 11.1: First-order digital filters.

5 Normalised frequency

Since the frequency response of a filter scales with sampling frequency $f_s = 1/\Delta t$, normalised frequency is often employed to describe the frequency response of a digital filter,

$$H_{\frac{1}{\Delta t}}(F) = H\left(\exp\left(\mathrm{j}2\pi F\right)\right), \tag{11.36}$$

where $F = f\Delta t = f/f_s$. The frequency response is usually only plotted for $-0.5 \leq F < 0.5$ since it is periodic.

6 Periodic sinc function

With digital signal processing the following function often pops up[11],

$$\frac{\sin(M\pi x)}{\sin(\pi x)}. \tag{11.37}$$

This is similar to a sinc function but it is periodic. It is called many names: periodic sinc, psinc, aliased sinc, Dirichlet function. It has a maximum amplitude of M when x is an integer, see Figure 11.12–Figure 11.14[12].

[11] It is the DTFT of a sequence of M ones scaled by $1/M$.

[12] Sometimes it is normalised. SciPy has the `diric(x,M)` function defined as $\sin\left(Mx/2\right)/\left(M\sin\left(x/2\right)\right)$.

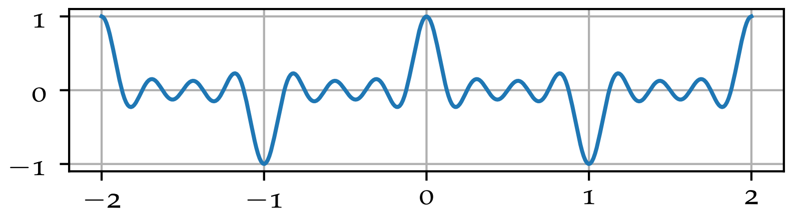


Figure 11.12: Periodic sinc for $M = 8$.

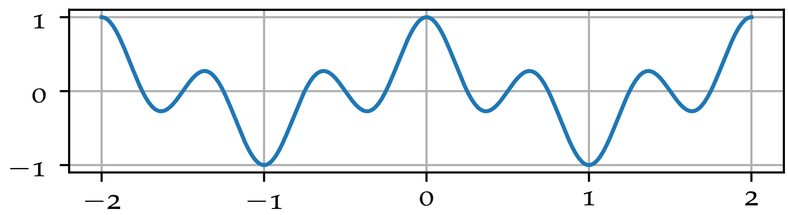


Figure 11.13: Periodic sinc for $M = 4$.

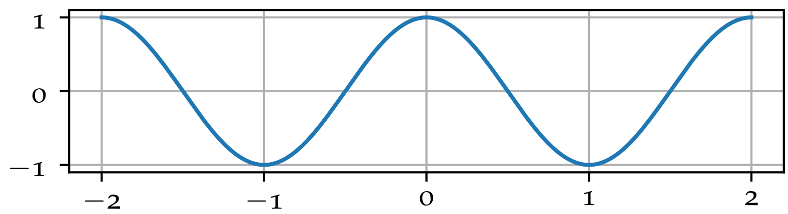


Figure 11.14: Periodic sinc for $M = 2$.

7 Initial conditions

(11.7) and (11.13) ignored the effect of the initial conditions. This section now considers the response due to these initial conditions.

7.1 Two sample moving average filter

The two sample moving average filter can be described by

$$y[n] = \begin{cases} x[0] & n = 0, \\ \frac{1}{2}\left(x[n] + x[n-1]\right) & n > 0. \end{cases} \tag{11.38}$$

Assuming that $x[n] = 0$ for $n < 0$, the z-transform of the filter output is

$$Y(z) = \sum_{n=-\infty}^{\infty} y[n]z^{-n}, \tag{11.39}$$

$$= x[0] + \sum_{n=1}^{\infty} \frac{1}{2}\left(x[n] + x[n-1]\right) z^{-n}, \tag{11.40}$$

$$= x[0] + \frac{1}{2}\sum_{n=1}^{\infty} x[n]z^{-n} + \frac{1}{2}\sum_{n=1}^{\infty} x[n-1]z^{-n}, \tag{11.41}$$

$$= \frac{1}{2}x[0] + \frac{1}{2}\sum_{n=0}^{\infty} x[n]z^{-n} + \frac{1}{2}\sum_{m=0}^{\infty} x[m]z^{-(m+1)}, \tag{11.42}$$

$$= \frac{1}{2}x[0] + \frac{1}{2}\sum_{n=0}^{\infty} x[n]z^{-n} + \frac{1}{2}z^{-1}\sum_{m=0}^{\infty} x[m]z^{-m}, \tag{11.43}$$

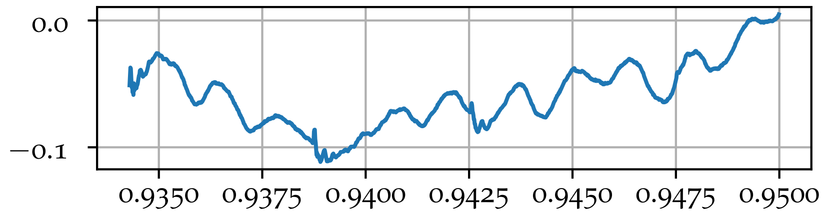
$$= \frac{1}{2}x[0] + \frac{1}{2}X(z) + \frac{1}{2}z^{-1}X(z), \tag{11.44}$$

$$= \frac{1}{2}x[0] + \frac{1}{2}\left(1 + z^{-1}\right) X(z). \tag{11.45}$$

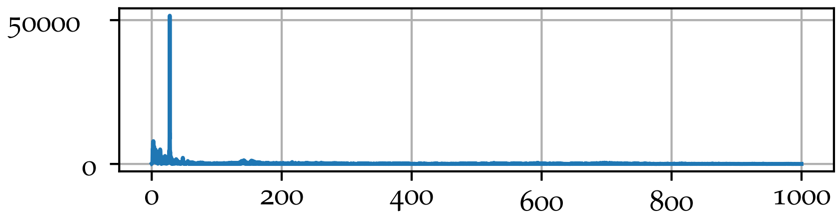
The first term is the single sample transient response due to the initial condition.

8 *Listening to the beetles*

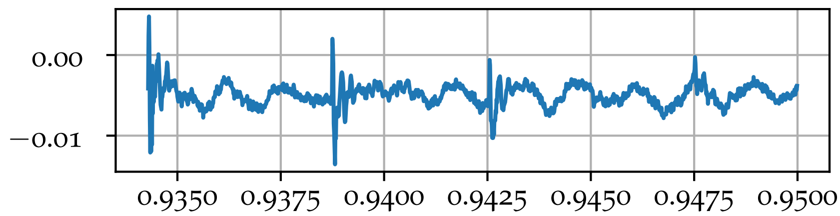
Figure 11.15 shows plots of the faint sounds produced by beetles. Unfortunately, the signal is swamped by 27 Hz, 700 Hz, and 5 Hz interference. These tones have been reduced by notch filters revealing the underlying clicks produced by the beetles.



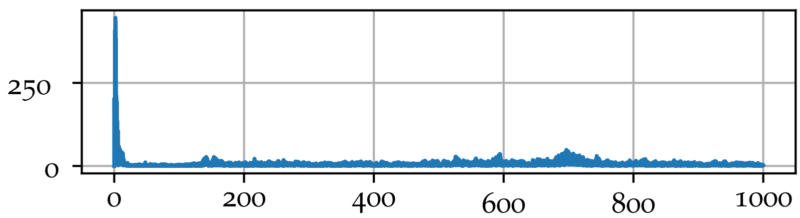
(a)



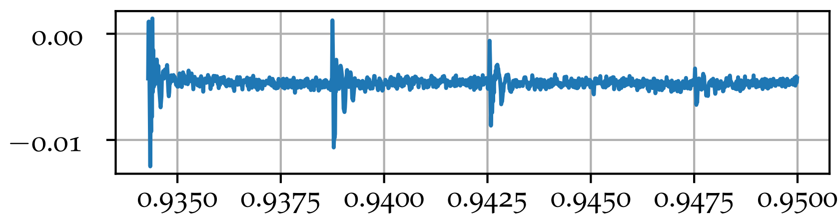
(b)



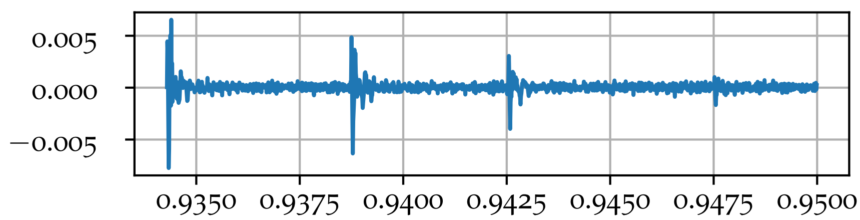
(c)



(d)



(e)



(f)

Figure 11.15: Beetle signals: (a) measured signal, (b) measured signal spectrum showing 27 Hz interference, (c) signal filtered by 27 Hz notch filter, (d) spectrum of filtered signal, (e) signal also filtered by 700 Hz notch filter.

9 Exercises

1. (a) Determine the z-transform of $ax[n] + by[n]$.
   (b) Determine the z-transform of $x[n-1]$.
   (c) Determine the z-transform of $0.7x[n] + 0.3x[n-1]$.
   (d) If $y[n] = 0.7y[n-1] + 0.3x[n]$ show that

$$Y(z) = \frac{0.3}{1 - 0.7z^{-1}} X(z).$$

2. Consider a filter with the z-domain transfer function $H(z) = 0.5 + 0.5z^{-1}$.

   (a) If $H(z) = Y(z)/X(z)$ determine the difference equation for $y[n]$.
   (b) Show that the frequency response is

$$H_{\frac{1}{\Delta t}}(f) = \exp(-\mathrm{j}\pi f \Delta t) \cos(\pi f \Delta t).$$

   Hint: $1 + \exp(\mathrm{j}2\theta) = \exp(\mathrm{j}\theta)(\exp(-\mathrm{j}\theta) + \exp(\mathrm{j}\theta))$.
   (c) Determine the gain of the filter at DC.
   (d) Determine the gain of the filter at $f = f_s/4$ where f_s is the sampling frequency.
   (e) Sketch the magnitude of the frequency response assuming the sampling frequency is 1 kHz for frequencies between DC and 2 kHz.
   (f) What sort of filter is this?

3. Consider a filter with the z-domain transfer function $H(z) = \left(1 - z^{-1}\right)/\Delta t$.

   (a) If $H(z) = Y(z)/X(z)$ determine the difference equation for $y[n]$.
   (b) Show that the expression for the frequency response is

$$H_{\frac{1}{\Delta t}}(f) = \frac{2\mathrm{j}}{\Delta t} \exp(-\mathrm{j}\pi f \Delta t) \sin(\pi f \Delta t).$$

   (c) Determine the gain of the filter at DC.
   (d) Determine the gain of the filter at $f = f_s/4$ where f_s is the sampling frequency.
   (e) Sketch the magnitude of the frequency response assuming the sampling frequency is 1 kHz for frequencies between DC and 2 kHz.
   (f) What sort of filter is this?

4. A gyroscope signal is filtered by first passing it through a first-order integrator with a z-domain transfer function $H_{\mathrm{INT}}(z) = \frac{\Delta t}{1-z^{-1}}$ and then through a first-order high-pass filter with a z-domain transfer function $H_{\mathrm{HPF}}(z) = \frac{\alpha(1-z^{-1})}{1-\alpha z^{-1}}$.
   (a) Determine the difference equation for the integrator.
   (b) Determine the difference equation for the high-pass filter.
   (c) Determine the overall z-domain transfer function for the combined response of the two filters.
   (d) Determine the overall difference equation for the combined filters.
   (e) Determine the DC gain of the combined filter assuming the sampling frequency is 1 kHz and $\alpha = 0.5$.

5. The discrete time Fourier transform of a digital filter has the form,

$$H_{\frac{1}{\Delta t}}(f) = \frac{0.2}{1 - 0.8\exp\left(-\mathrm{j}2\pi f \Delta t\right)}. \tag{11.46}$$

   Determine the z-transform transfer function of the filter, $H(z)$.

12
Digital filter impulse response

There is a tradeoff between the time and frequency response of a filter. A narrowband filter has a long transient response and vice-versa. This lecture considers a special transient response called the impulse response.

1 Impulse response from a transfer function

Recall that the transfer function of a digital filter in the z-domain is found from

$$H(z) = \left.\frac{Y(z)}{X(z)}\right|_{\text{I.C.}=0}. \tag{12.1}$$

From this the impulse response can be found from an inverse z-transform:

$$h[n] = \mathcal{Z}^{-1}\left\{H(z)\right\}. \tag{12.2}$$

2 Inverse z-transform

Solving the inverse z-transform is not easy[1]. So instead we use tricks and tables; one of the tricks is partial fraction expansion.

The key z-transform pair is

$$a^n u[n] \longleftrightarrow \frac{1}{1 - az^{-1}}. \tag{12.3}$$

This is analogous to the Laplace transform pair,

$$\exp(\alpha t)\, u(t) \longleftrightarrow \frac{1}{s - \alpha}. \tag{12.4}$$

[1] This requires solving the Cauchy integral. Like the Bromwich integral used to solve the inverse Laplace transform, this is an integral of a complex variable.

$x[n]$	$X(z)$
$\delta[n]$	1
$a^n u[n]$	$\frac{1}{1-az^{-1}}$
$u[n]$	$\frac{1}{1-z^{-1}}$
$\delta[n-1]$	z^{-1}

Table 12.1: Table of common z-transforms.

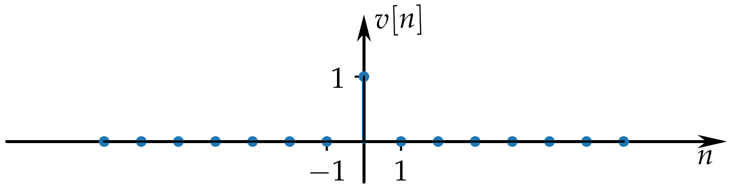
2.1 *Unit impulse*

The unit impulse (unit sample or discrete-time impulse[2]) is defined as

$$\delta[n] = \begin{cases} 1 & n = 0, \\ 0 & n \neq 0. \end{cases} \tag{12.5}$$

This corresponds to the sequence: $\{\cdots, 0, 0, 0, \underline{1}, 0, 0, 0, \cdots\}$. It has a rather simple z-transform

$$\sum_{n=-\infty}^{\infty} \delta[n] z^{-n} = 1. \tag{12.6}$$



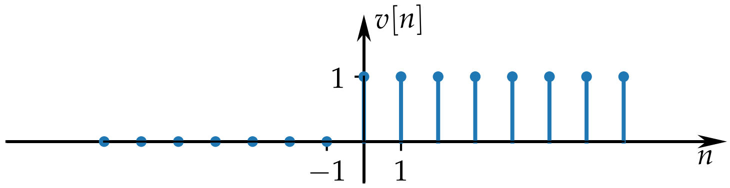
2.2 *Unit step*

The unit step is defined as

$$u[n] = \begin{cases} 0 & n < 0, \\ 1 & n \geq 0. \end{cases} \tag{12.7}$$

This corresponds to the sequence: $\{\cdots, 0, 0, 0, \underline{1}, 1, 1, 1, \cdots\}$. Its z-transform is

$$\sum_{n=-\infty}^{\infty} u[n] z^{-n} = \sum_{n=0}^{\infty} z^{-n} = \frac{1}{1 - z^{-1}}. \tag{12.8}$$



[2] It has a similar purpose to the Dirac delta.

2.3 *Example: impulse response of first-order recursive filter*

Consider the first-order recursive filter with zero initial conditions defined by

$$y[n] = ay[n-1] + bx[n]. \tag{12.9}$$

In the z-domain this can be expressed as

$$Y(z) = az^{-1}Y(z) + bX(z), \tag{12.10}$$

and thus the transfer function is

$$H(z) = \left.\frac{Y(z)}{X(z)}\right|_{\text{I.C.}=0} = \frac{b}{1-az^{-1}}. \tag{12.11}$$

Using the table of z-transforms Table 12.1, the corresponding impulse response is

$$h[n] = ba^n u[n]. \tag{12.12}$$

This result can be obtained by inputting a unit impulse, $x[n] = \delta[n]$, into the filter defined by (12.9). After a few iterations the output sequence can be found to be

$$y[n] = \{\underline{b}, ab, a^2b, a^3b, \cdots\}. \tag{12.13}$$

This can be generalised as[3]:

[3] $u[n]$ is the unit step sequence.

$$h[n] = ba^n u[n]. \tag{12.14}$$

There are three things to note:

1. The impulse response is causal since $h[n] = 0$ for $n < 0$.
2. The impulse response has an infinite number of terms and is called an infinite impulse response (IIR).
3. If $a \geq 1$, the impulse response does not decay and thus the filter is not stable.

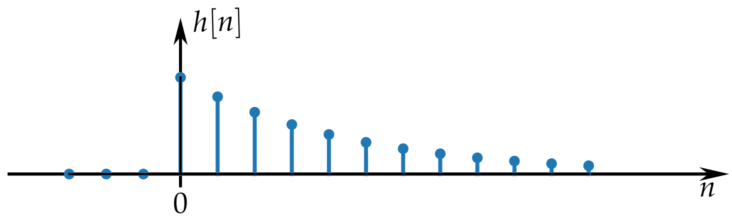


Figure 12.1: First-order recursive low-pass filter impulse response with $\alpha = 0.8$.

2.4 *Example: two sample moving average filter*

The two sample moving average filter is described by:

$$y[n] = \frac{1}{2}x[n] + \frac{1}{2}x[n-1], \tag{12.15}$$

with a discrete-time transfer function

$$H(z) = \frac{1}{2} + \frac{1}{2}z^{-1}. \tag{12.16}$$

Using the table of z-transforms and the shift-theorem, the discrete-time impulse response for this filter is

$$h[n] = \frac{1}{2}\delta[n] + \frac{1}{2}\delta[n-1]. \tag{12.17}$$

This can be expressed as the sequence $h[n] = \{\underline{0.5}, 0.5\}$. Since $h[n]$ has a finite number of non-zero terms, the filter is called a finite impulse response (FIR) filter.

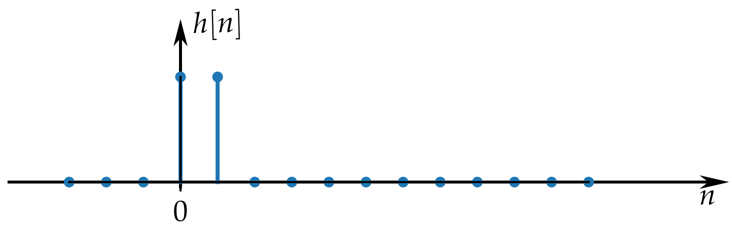


Figure 12.2: Two sample moving average low-pass filter impulse response.

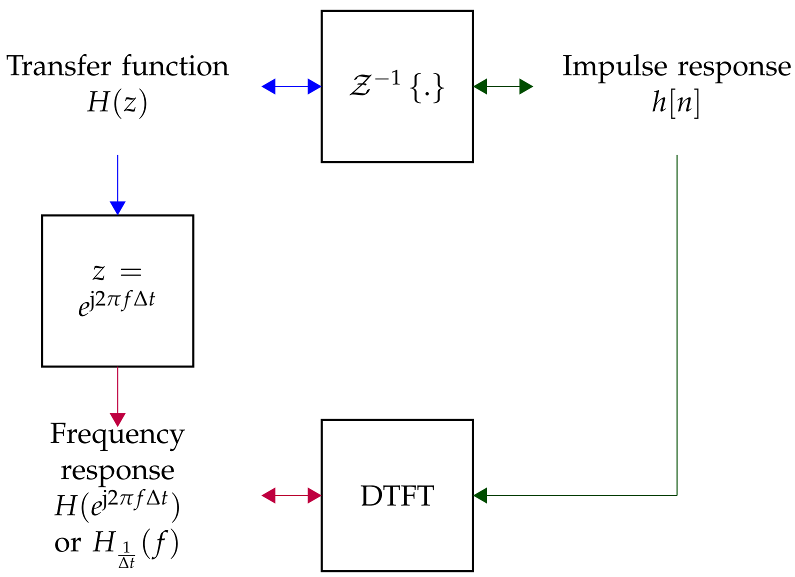


Figure 12.3: Relationship between transfer function, impulse response, and frequency response for a discrete-time causal system where $h[n] = 0$ for $n < 0$.

3 Exercises

1. For the filter described by $y[n] = x[n] - x[n-1]$ and the input sequence $x[n] = \{\underline{1}, 1\}$:

   (a) Determine the impulse response, $h[n]$, i.e., the response due to the signal $\delta[n] = \{\underline{1}\}$.

   (b) Determine the response of the filter, $y[n]$, assuming zero initial conditions.

   (c) Determine the z-transform, $H(z)$, of the filter impulse response, $h[n]$.

   (d) Determine the z-transform, $X(z)$, of the input sequence, $x[n]$.

   (e) Determine the z-transform, $Y(z) = H(z)X(z)$, of the filter response, $y[n]$.

   (f) From the z-transform of the filter response, $Y(z)$, determine the filter response $y[n]$.

2. For the filter described by $y[n] = x[n] + 2x[n-1] + x[n-2]$ and the input sequence $x[n] = \{\underline{1}, -1\}$:

   (a) Determine the impulse response, $h[n]$, i.e., the response due to the signal $\delta[n] = \{\underline{1}\}$.

   (b) Determine the response of the filter, $y[n]$, assuming zero initial conditions.

   (c) Determine the z-transform, $H(z)$, of the filter impulse response, $h[n]$.

   (d) Determine the z-transform, $X(z)$, of the input sequence, $x[n]$.

   (e) Determine the z-transform, $Y(z) = H(z)X(z)$, of the filter response, $y[n]$.

   (f) From the z-transform of the filter response, $Y(z)$, determine the filter response $y[n]$.

3. For the filter described by $y[n] = 0.8y[n-1] - 0.2x[n]$:

   (a) Determine the first four non-zero samples of the impulse response of the filter, i.e., the response due to the signal $\delta[n] = \{\underline{1}\}$.

   (b) What is the value of $y[100]$ when $x[n] = \delta[n]$? *Hint, generalise the previous answer.*

   (c) Determine the impulse response, $h[n]$, of the filter for all n.

   (d) Determine the z-transform, $Y(z)$, of the filter response, $y[n]$.

(e) Determine the z-transform, $H(z)$, of the filter impulse response, $h[n]$.

(f) Determine the impulse response, $h[n]$, from the transfer function, $H(z)$.

4. (a) What is the impulse response of the filter defined by $y[n] = 0.75x[n] - 0.25x[n-1]$? Is it a FIR filter?

   (b) What is the impulse response of the filter defined by $y[n] = 0.75y[n-1] - 0.25x[n]$? Is it a FIR filter?

13
Discrete convolution

There are three ways that the output of an LTI discrete-time filter can be found, given an input signal, $x[n]$:

1. Evaluate the difference equation for each time step n.
2. Determine the z-transform of the input signal, multiply by the filter transfer function, $H(z)$, and take the inverse z-transform:
$$y[n] = \mathcal{Z}^{-1}\left\{X(z)H(z)\right\}. \tag{13.1}$$
3. Convolve the input signal with the filter impulse response, $h[n]$:
$$y[n] = x[n] * h[n]. \tag{13.2}$$

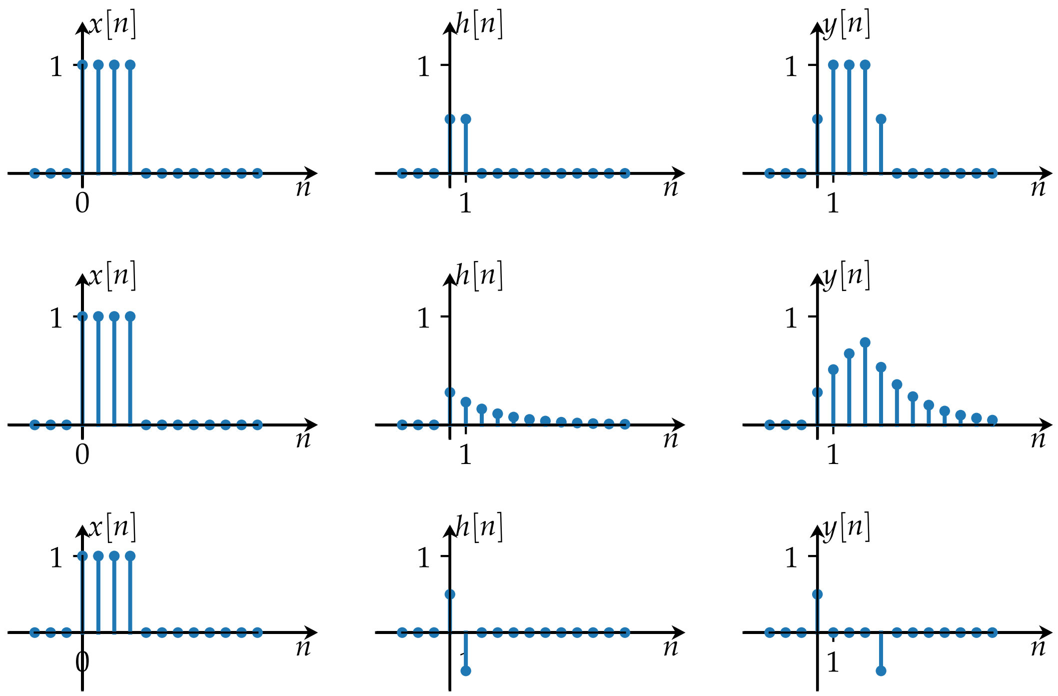


Figure 13.1: Example filter responses. Guess the filters!

1 Discrete-time convolution

The convolution integral for discrete time signals is

$$y[n] = h[n] * x[n] = \sum_{m=0}^{M-1} h[m]x[n-m]. \tag{13.3}$$

Here M denotes the extent[1] of the impulse response, $h[n]$. Expanding the summation yields:

$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + \cdots \tag{13.4}$$

[1] This is infinite for a recursive filter.

1.1 Extent

The extent of a discrete-time signal is the span over the non-zero samples. For example,

Extent $\{1, 2, 4\} = 3$

Extent $\{1, 2, 0, 4\} = 4$

Extent $\{0, 1, 2, 0, 4, 0, 0\} = 4$

1.2 FIR and IIR

FIR Moving average filters have a finite impulse response (FIR), i.e., Extent $\{h[n]\}$ is finite.

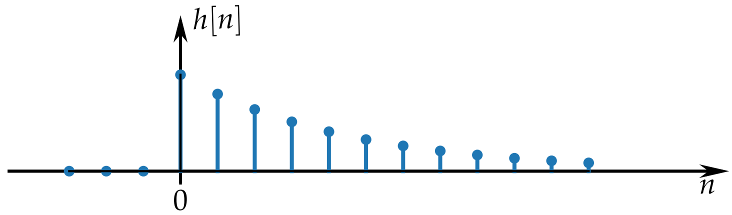


Figure 13.2: Example IIR.

IIR Recursive filters have an infinite impulse response (FIR), i.e., Extent $\{h[n]\}$ is infinite.

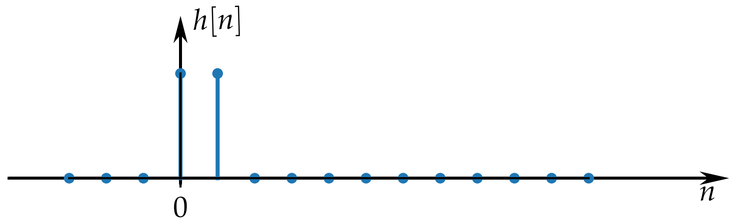


Figure 13.3: Example FIR.

1.3 Convolution extent

Convolution broadens the extent of a signal. For example, given

$$y[n] = h[n] * x[n], \tag{13.5}$$

then

$$\text{Extent}\,\{y[n]\} = \text{Extent}\,\{h[n]\} + \text{Extent}\,\{x[n]\} - 1. \tag{13.6}$$

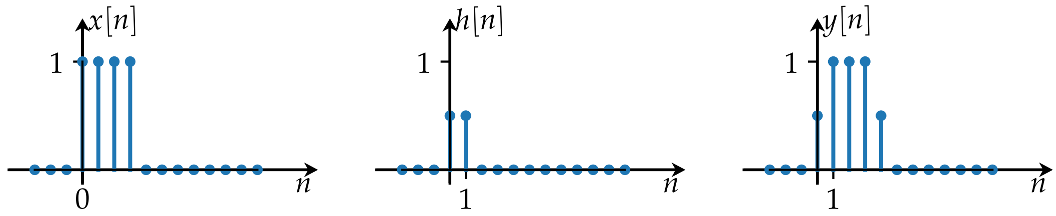


Figure 13.4: Convolution increases the signal extent.

1.4 Convolution example

Consider a FIR filter with an impulse response $h[n] = \{\underline{h_0}, h_1\}$ and an input sequence $x[n] = \{\underline{x_0}, x_1, x_2\}$. The output extent is $2 + 3 - 1 = 4$ and thus the output signal can be denoted $y[n] = \{\underline{y_0}, y_1, y_2, y_3\}$, where

$$y_0 = h_0 x_0 + 0, \tag{13.7}$$
$$y_1 = h_0 x_1 + h_1 x_0, \tag{13.8}$$
$$y_2 = h_0 x_2 + h_1 x_1, \tag{13.9}$$
$$y_3 = 0 + h_1 x_2. \tag{13.10}$$

This can be written in matrix form as

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} h_0 & 0 & 0 \\ h_1 & h_0 & 0 \\ 0 & h_1 & h_0 \\ 0 & 0 & h_1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}. \tag{13.11}$$

2 Convolution theorem

One of the advantages of the z-transform is that it avoids convolutions. The discrete-time convolution theorem is

$$\sum_{m=-\infty}^{\infty} h[m]x[n-m] \longleftrightarrow X(z)H(z). \tag{13.12}$$

This is usually denoted as

$$x[n] * h[n] \longleftrightarrow X(z)H(z). \tag{13.13}$$

For example,

$$\delta[n] * h[n] \longleftrightarrow 1 \times H(z). \qquad (13.14)$$

This confirms $\delta[n] * h[n] = h[n]$.

2.1 *Causality*

If the filter is causal then the impulse response $h[n] = 0$ for $n < 0$ and so the convolution (13.12) simplifies to

$$y[n] = \sum_{m=0}^{\infty} h[m]x[n-m]. \qquad (13.15)$$

2.2 *Finite extent*

If the causal impulse response remains zero after M terms, then the convolution can be further simplified to

$$y[n] = \sum_{m=0}^{M-1} h[m]x[n-m]. \qquad (13.16)$$

2.3 *Step response*

The step response is the cumulative sum of the impulse response. This can be described as the convolution of the filter's impulse response with a unit step, $h[n] * u[n]$. In the z-domain, this is a multiplication,

$$\sum_{m=0}^{n} h[n] = h[n] * u[n] \longleftrightarrow H(z)\frac{1}{1-z^{-1}}. \qquad (13.17)$$

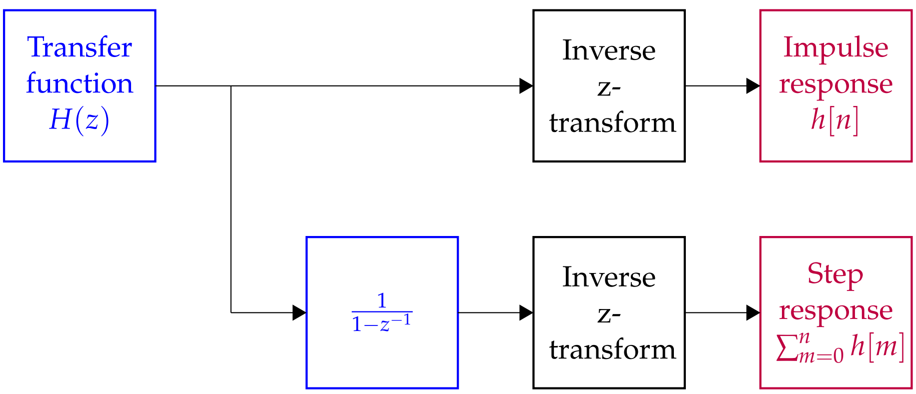


Figure 13.5: Relationship of transfer function to impulse and step response.

3 Convolution as polynomial multiplication

Consider a signal, $x[n]$, and an impulse-response, $h[n]$, where

$$x[n] = \{\underline{x_0}, x_1, x_2\}, \tag{13.18}$$

$$h[n] = \{\underline{h_0}, h_1\}. \tag{13.19}$$

These sequences can be represented as

$$x[n] = x_0\delta[n] + x_1\delta[n-1] + x_2\delta[n-2], \tag{13.20}$$

$$h[n] = h_0\delta[n] + h_1\delta[n-1]. \tag{13.21}$$

The discrete convolution of $x[n]$ and $h[n]$ is given by

$$y[n] = \sum_{m=0}^{1} h[m]x[n-m]. \tag{13.22}$$

The non-zero terms of $y[n]$ are

$$y[0] = h[0]x[0] = h_0x_0, \tag{13.23}$$

$$y[1] = h[0]x[1] + h[1]x[0] = h_0x_1 + h_1x_0, \tag{13.24}$$

$$y[2] = h[0]x[2] + h[1]x[1] = h_0x_2 + h_1x_1. \tag{13.25}$$

Thus the result can be expressed as:

$$y[n] = h_0x_0\delta[n] + (h_0x_1 + h_1x_0)\,\delta[n-1] + (h_0x_2 + h_1x_1)\,\delta[n-2]. \tag{13.26}$$

In the z-domain,

$$X(z) = x_0 + x_1z^{-1} + x_2z^{-2}, \tag{13.27}$$

$$H(z) = h_0 + h_1z^{-1}. \tag{13.28}$$

The product of these two z-transforms is

$$Y(z) = H(z)X(z), \tag{13.29}$$

$$= h_0x_0 + (h_0x_1 + h_1x_0)\,z^{-1} + (h_0x_2 + h_1x_1)\,z^{-2}, \tag{13.30}$$

with an inverse z-transform

$$y[n] = h_0x_0\delta[n] + (h_0x_1 + h_1x_0)\,\delta[n-1] + (h_0x_2 + h_1x_1)\,\delta[n-2]. \tag{13.31}$$

Comparing (13.26) and (13.31), the same answer is achieved as expected by the convolution theorem. This shows that discrete convolution is equivalent to polynomial multiplication.

4 *Convolution using the DFT*

Convolution is expensive to compute numerically. If the impulse response has an extent of M samples, then each sample $y[n]$ requires M multiplications and additions.

When M is large it is more efficient to compute convolutions with the FFT (fast fourier transform). The FFT is equivalent to the DFT but much fast to compute. The convolution is performed using two forward FFTs and an inverse FFT:

$$\mathrm{FFT}^{-1}\left\{\mathrm{FFT}\left\{x[n]\right\}\mathrm{FFT}\left\{h[n]\right\}\right\} = x[n] \circledast h[n]. \quad (13.32)$$

Often this is denoted as:

$$X[k]H[k] \leftrightarrow x[n] \circledast h[n]. \quad (13.33)$$

There is a subtle difference when using the FFT to compute a convolution since the result is a circular convolution; zero padding is required to ensure that the result is equivalent to a convolution.

5 Exercises

1. (a) Write the sequence $\{\underline{1}, 2, 3\}$ as a weighted sum of impulses.
   (b) Write the sequence $\{1, \underline{2}, 3\}$ as a weighted sum of impulses.
   (c) Write $1\delta[n] - 3\delta[n-2] + 5\delta[n-4]$ as a sequence.
   (d) Determine the convolution of the sequences $x[n] = \{\underline{1}, 1, 1\}$ and $h[n] = \{\underline{1}, 1, 1\}$?
   (e) Determine the convolution of the sequences $x[n] = \{\underline{1}, 2, 3\}$ and $h[n] = \{\underline{1}, 1\}$?
   (f) What is the extent of the convolution of the sequences $\{\underline{1}, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and $\{\underline{8}, 7, 6, 5, 4, 3, 2, 1\}$?
   (g) How many multiplications are required for the convolution in the previous question?
   (h) If sequences of extents 100 and 50 are convolved, what is the extent of the result?
   (i) If sequences of extents 100 and 50 are convolved, how many multiplications are required?

2. Consider the sequence $x[n] = \{\underline{1}, 2, 1, 3\}$[2] to be convolved with a sequence $h[n] = \{\underline{0.5}, 0.5\}$.
   (a) Express the sequence $x[n]$ as a weighted sum of delayed impulses[3].
   (b) Sketch the lollipop (stem) plot for $x[n]$.
   (c) Express the sequence $h[n]$ as a weighted sum of delayed impulses.
   (d) Sketch the lollipop (stem) plot for $h[n]$.
   (e) Sketch the lollipop plot for $x[n]$ convolved with $h[n]$. Hint: expand the convolution of $x[n]$ with $h[n]$ given by

$$y[n] = \sum_{m=0}^{M-1} h[m]x[n-m]. \tag{13.34}$$

   (f) What is the z-transform of $h[n]$?
   (g) What sort of filter does this convolution perform?

[2] i.e., $x[-1] = 0, x[0] = 1, x[1] = 2, x[2] = 1, x[3] = 3, x[4] = 0$.

[3] $\{\underline{a}, b, c\} = a\delta[n] + b\delta[n-1] + c\delta[n-2]$.

14 Filter design

Filters are straightforward to analyse but hard to design since there are many tradeoffs to consider, for example, frequency response, time response, and computation time. Most people use filter design software.

1 Linear filters

In general, the steady-state difference equation of a linear filter can be generalised as

$$\sum_{i=0}^{p} a_i y[n-i] = \sum_{m=0}^{q} b_m x[n-m]. \tag{14.1}$$

Taking z-transforms of both sides gives

$$Y(z) \sum_{i=0}^{p} a_i z^{-i} = X(z) \sum_{m=0}^{q} b_m z^{-m}. \tag{14.2}$$

Thus the z-domain transfer function[1] is

$$H(z) = \left.\frac{Y(z)}{X(z)}\right|_{\text{I.C.s}=0} = \frac{\sum_{m=0}^{q} b_m z^{-m}}{\sum_{i=0}^{p} a_i z^{-i}}. \tag{14.3}$$

p is the number of poles and q is the number of zeros; the filter order is $\max(p, q)$.

[1] Note that if $p = 0$ and $a_0 = 1$ then the resulting filter is a weighted moving average filter with a transfer function $H(z) = \sum_{m=0}^{q} b_m z^{-m}$ and impulse response $h[n] = \sum_{m=0}^{q} b_m \delta[n-m]$.

1.1 Example: first-order recursive filter

Consider the first-order recursive filter[2] with the steady-state difference equation:

$$a_0 y[n] = b_0 x[n] + b_1 x[n-1] - a_1 y[n-1]. \tag{14.4}$$

This can be rearranged as

$$a_0 y[n] + a_1 y[n-1] = b_0 x[n] + b_1 x[n-1]. \tag{14.5}$$

The resultant transfer function is

$$H(z) = \left.\frac{Y(z)}{X(z)}\right|_{\text{I.C.s}=0} = \frac{b_0 + b_1 z^{-1}}{a_0 + a_1 z^{-1}}. \tag{14.6}$$

[2] With one pole and one zero.

2 Filter types

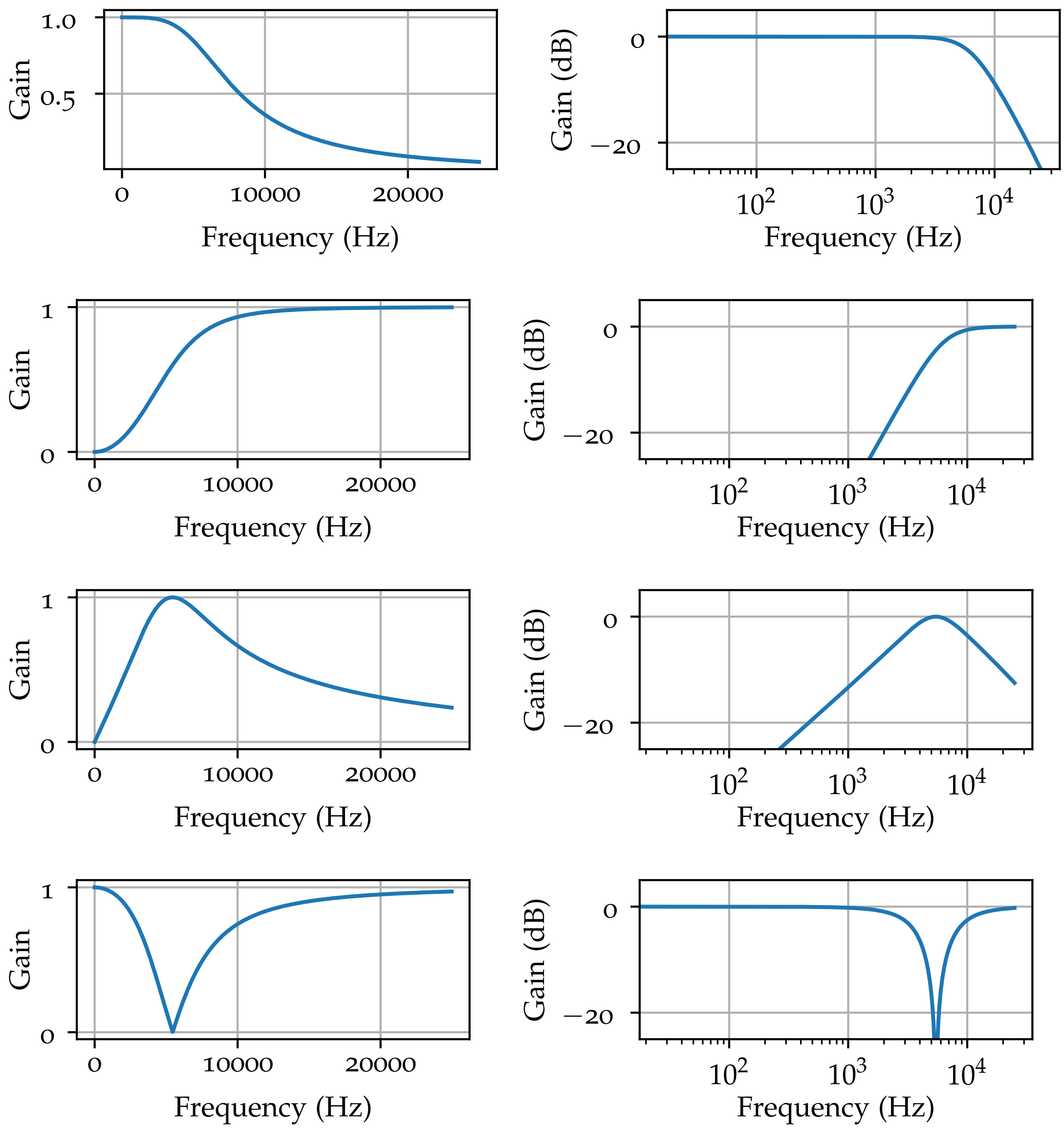


Figure 14.1: Examples of second-order digital Butterworth filters: low-pass, high-pass, band-pass, and band-stop.

3 *Numerical linear filter design*

Matlab and SciPy (Python) have functions for designing a filter. You specify the filter type, the filter order, and the normalised break frequencies[3] and they return vectors of the numerator and denominator coefficients.

[3] Normalised to half the sampling frequency.

```
from numpy.random import standard_normal
from numpy import arange, cos, pi
from scipy.signal import butter, lfilter

# Sampling frequency
fs = 50e3

# Break frequency
fb = 2e3

# Design 2nd-order Butterworth low-pass IIR filter
b, a = butter(2, 2 * fb / fs, 'lowpass')

# Create cosine signal
N = 1000
n = arange(N)
x = 5 + 2 * (1 + cos(2 * pi * n * 2 / N))

# Add zero-mean, unit variance Gaussian noise
x = x + standard_normal(N)

# Add dropout
x[100:104] = 0

# Apply filter
y = lfilter(b, a, x)
```

Listing 14.1: Example script to apply a second-order Butterworth low-pass digital filter.

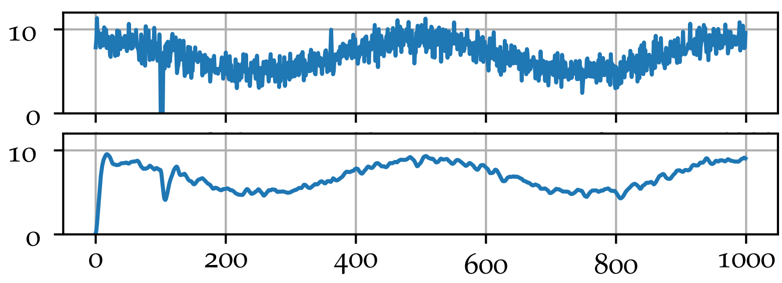


Figure 14.2: Operation of second-order Butterworth low-pass filter to reduce noise.

3.1 *Frequency response*

The frequency response can be determined with the `freqz` function.

```
from scipy.signal import butter, freqz
from numpy import arange
from matplotlib.pyplot import figure, show

# Sampling frequency
fs = 50e3

# Break frequency
fb = 2e3

# Design 2nd-order Butterworth low-pass IIR filter
b, a = butter(2, 2 * fb / fs, 'lowpass')

# Calculate frequency response
N = 1000
f = arange(N) * fs / (2 * N)
w, H = freqz(b, a, 2 * f / fs)

# Plot frequency response
fig = figure()
ax = fig.add_subplot(111)
ax.plot(f, abs(H))
ax.set_xlabel('Frequency (Hz)')
ax.set_ylabel('Gain')
ax.grid(True)
```

Listing 14.2: Example script to plot frequency response of second-order Butterworth low-pass filter.

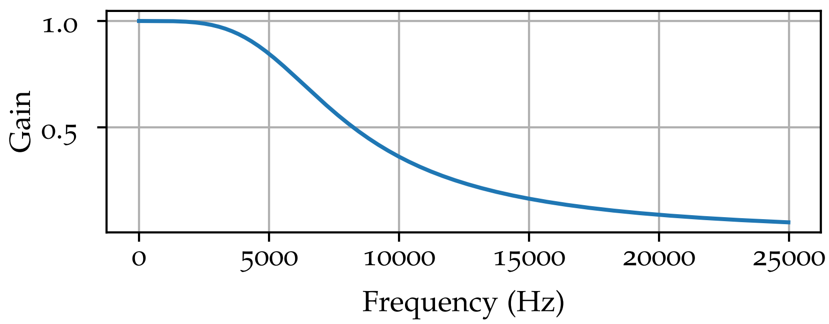


Figure 14.3: Second-order Butterworth low-pass filter frequency response.

3.2 *Impulse response*

The impulse response can be found by inputting a unit impulse to a filter:

```
from scipy.signal import butter, lfilter
from numpy import arange, zeros
from matplotlib.pyplot import figure, show
from matplotlib.pyplot import style

style.use('../etc/signal.mplstyle')

# Sampling frequency
fs = 50e3

# Break frequency
fb = 2e3

# Design 2nd-order Butterworth filter
b, a = butter(2, 2 * fb / fs, 'lowpass')

# Calculate impulse response
N = 50
x = zeros(N)
x[0] = 1
h = lfilter(b, a, x)

# Plot impulse response
fig = figure()
ax = fig.add_subplot(111)
ax.stem(h, basefmt=' ')
ax.set_xlabel('Sample')
ax.grid(True)
```

Listing 14.3: Example script to plot impulse response of second-order Butterworth low-pass filter.

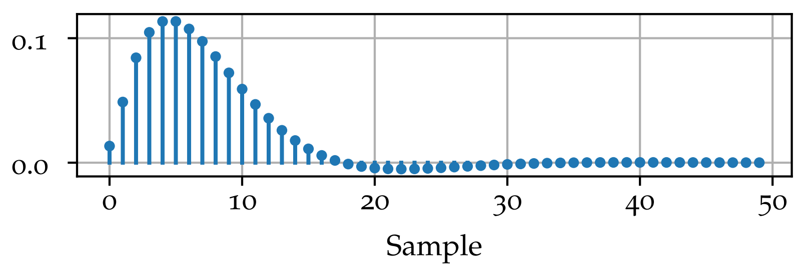


Figure 14.4: Second-order Butterworth low-pass filter impulse response.

3.3 *FIR versus IIR filters*

A FIR filter has a transfer function with no poles. If it has a symmetrical impulse response, then its phase response phase varies linearly with frequency (this is called *linear phase*) and its group delay[4] is constant for all frequencies. In comparison, non-symmetrical FIR filters and all IIR filters have a non-linear phase.

FIR filters are better for high order filters although they require greater computation that IIR filters. Since they do not use recursion there are no stability problems[5].

If in doubt use a FIR filter; it has better properties. Only use an IIR filter when you need a quick and dirty filter. If you do use an IIR filter, use a low order filter to ensure stability.

[4] All filters have a delay.

[5] High order IIR filters can go unstable due to floating point quantisation. They are best achieved by cascading second (and first) order IIR filters together.

Aspect	FIR	IIR
Computation	High	Low
Memory	High	Low
Stability	Stable	Can be unstable‡
Phase	Linear†	Non-linear
Group delay	$(M-1)/2$ samples†	Frequency dependent

Table 14.1: Comparison of FIR and IIR filters. † assuming a symmetrical impulse response. ‡ for filters higher than second-order.

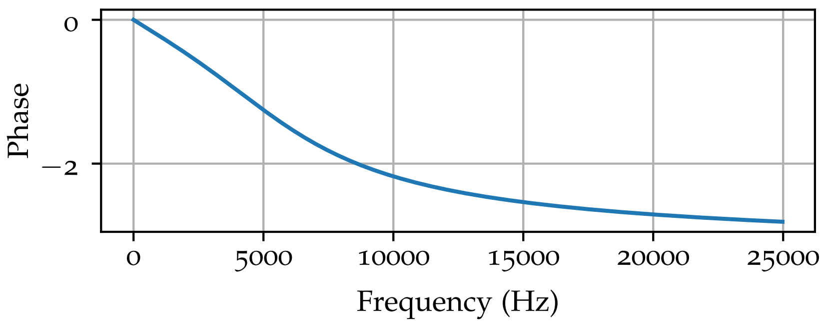


Figure 14.5: Phase response for a second order Butterworth (IIR) low-pass filter. Note, it is not linear with frequency.

4 *Non-linear filters*

Linear filters do not work well if there is impulsive noise; say due to sensor drop outs. In this case it is better to use a non-linear filter.

4.1 *Median filter*

One common non-linear filter is the median filter. This considers a sliding window of samples and selects the median value from the window as the output value.

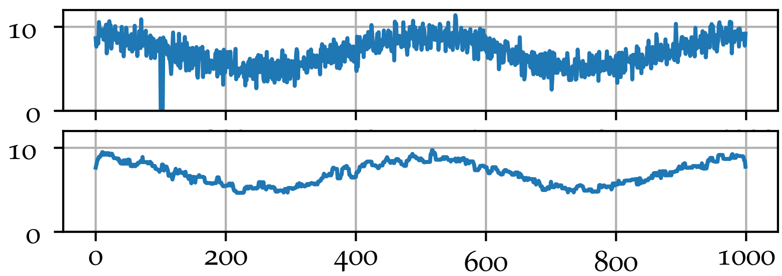


Figure 14.6: Operation of 11 sample median filter to reduce effect of drop out.

```
from numpy.random import standard_normal
from numpy import cos, pi, arange
from scipy.signal import medfilt

# Create cosine signal
N = 1000
n = arange(N)
x = 5 + 2 * (1 + cos(2 * pi * n * 2 / N))

# Add zero-mean, unit variance Gaussian noise
x = x + standard_normal(N)

# Add dropout
x[100:104] = 0

# Apply median filter over 11 sample window
y = medfilt(x, 11)
```

Listing 14.4: Example script to apply a median filter to a cosine contaminated by a drop-out.

5 *Filter example: automatic double tracking*

Automatic double tracking (ADT[6] is a technique used to make sounds "thicker" and "fuller" by adding a signal delayed by a short interval to the original signal. It can

[6] A technique first used by The Beatles.

be described by the impulse response

$$h[n] = b_0\delta[n] + b_M\delta[n-M], \tag{14.7}$$

with a transfer function

$$H(z) = b_0 + b_M z^{-M}. \tag{14.8}$$

The frequency response is

$$H_f(f) = H\left(\exp\left(\mathrm{j}2\pi f\Delta t\right)\right), \tag{14.9}$$
$$= b_0 + b_M \exp\left(-\mathrm{j}2\pi f M\Delta t\right), \tag{14.10}$$
$$= b_0 - b_M + b_M \exp\left(-\mathrm{j}\theta\right)\exp\left(\mathrm{j}\theta\right) + b_M \exp\left(-\mathrm{j}\theta\right)\exp\left(-\mathrm{j}\theta\right), \tag{14.11}$$
$$= b_0 - b_M + b_M \exp\left(-\mathrm{j}\theta\right)\cos\theta, \tag{14.12}$$
$$= b_0 - b_M + b_M \exp\left(-\mathrm{j}\pi f M\Delta t\right)\cos\left(\pi f M\Delta t\right), \tag{14.13}$$

where $\theta = \pi f M\Delta t$. The frequency response is plotted in Figure 14.7.

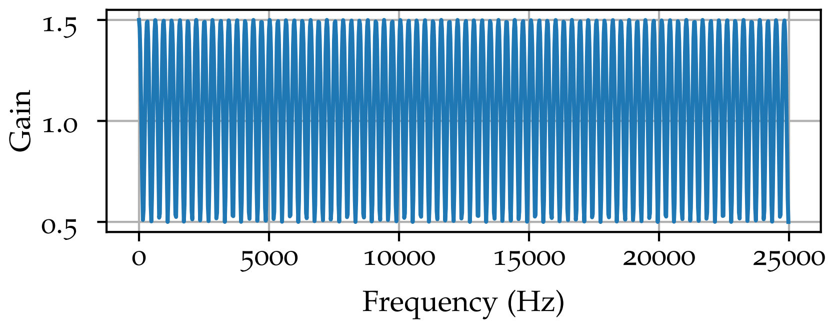


Figure 14.7: Frequency response of ADT with a delay of 10 ms, a sampling frequency of 50 kHz, $b_0 = 1$, and $b_M = 0.5$.

6 *Filter example: reverberation modelling*

Convolution can be used to alter a sound recording. For example, it can add a delayed echo. It can also be used to change the reverberation. For example, if we wanted to hear what a Dalek sounded like in a cathedral, we could measure the impulse response of the cathedral's acoustics, and then convolve the Dalek's speech with the measured impulse response, see Figure 14.8.

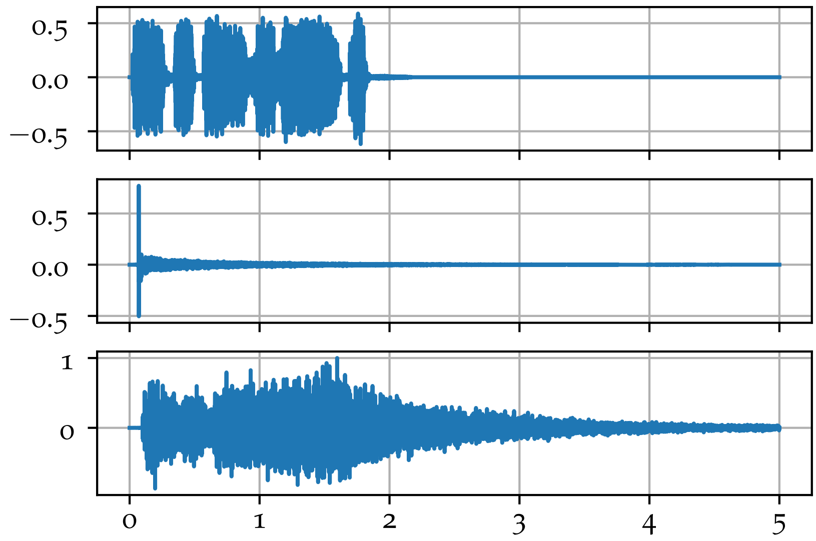
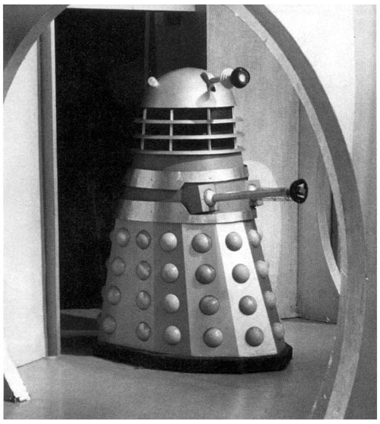


Figure 14.8: (a) Dalek saying Exterminate!, (b) impulse response of York Minster Cathedral, (c) Dalek saying Exterminate! in York Minster Cathedral.



7 *Approximating an analogue filter*

A digital filter can be derived from an analogue filter with $z = \exp(-s\Delta t)$ or $s = \frac{1}{\Delta t}\ln z$. An approximation is

$$s = \frac{1}{\Delta t}\ln z \approx \frac{2}{\Delta t}\frac{z-1}{z+1}. \tag{14.14}$$

However, substituting (14.14) into a s-domain transfer function produces much tedious maths to wade through.

8 *Biquad filters*

IIR filters are sensitive to quantisation of the coefficients and high order filters can become unstable. A compromise is to chain second-order filters[7] (biquad filters) to achieve the desired filter order.

In general, a second-order filter has a transfer function of the form:

$$H(z) = K\frac{(z-z_1)(s-z_2)}{(z-p_1)(z-p_2)}, \tag{14.15}$$

where z_1 and z_2 are the zeros and p_1 and p_2 are the

[7] And perhaps a first order filter.

poles. (14.15) can be expressed as

$$H(z) = K\frac{1-(z_1+z_2)z^{-1}+z_1z_2z^{-2}}{1-(p_1+p_2)z^{-1}+p_1p_2z^{-2}}, \quad (14.16)$$

$$= \frac{b_0+b_1z^{-1}+b_2z^{-2}}{a_0+a_1z^{-1}+a_2z^{-2}}. \quad (14.17)$$

By selection of K, z_1, z_2, p_1, p_2 or equivalently $a_0, a_1, a_2, b_0, b_1, b_2$ it is possible to construct a low-pass, high-pass, band-pass, or band-stop filter.

8.1 Second-order notch filter

A notch filter is useful to remove an interfering frequency component, say 50 Hz mains hum. It is a band-stop filter with a narrow bandwidth. To create a notch at a frequency f_0 requires the numerator of (14.15) to be zero at f_0. This requires the zeros to be at $z_1 = \exp(-\mathrm{j}2\pi f_0\Delta t)$ and $z_2 = \exp(\mathrm{j}2\pi f_0\Delta t)$. Defining $\theta_0 = 2\pi f_0\Delta_t$ then the zeros are at $\exp(\pm\mathrm{j}\theta_0)$. Thus the numerator of (14.15) is

$$N(z) = (z-\exp(-\mathrm{j}\theta_0))(z-\exp(\mathrm{j}\theta_0)) = 1-2\cos\theta_0 z^{-1}+z^{-2}. \quad (14.18)$$

The poles need to be placed close to the zeros; the closer they are the narrower the notch. They are chosen as a complex conjugate pair at $z = r\exp(\pm\mathrm{j}\phi)$, where

$$\cos\phi = \frac{\beta(1+\alpha)}{2\sqrt{\alpha}}, \quad (14.19)$$

$r = \sqrt{\alpha}$ and $\beta = \cos\theta_0$. The resulting transfer function is

$$H(z) = \frac{1-2\beta z^{-1}+z^{-2}}{1-\beta(1+\alpha)z^{-1}+\alpha z^{-2}}. \quad (14.20)$$

This does not have unity gain at DC since

$$H(z) = \frac{2-2\beta}{1-\beta(1+\alpha)+\alpha} = \frac{2}{1+\alpha}. \quad (14.21)$$

Thus to ensure a unity gain at DC, the transfer function is

$$H(z) = \frac{1+\alpha}{2}\frac{1-2\beta z^{-1}+z^{-2}}{1-\beta(1+\alpha)z^{-1}+\alpha z^{-2}}. \quad (14.22)$$

Denoting $K = (1+\alpha)/2$ then

$$H(z) = \frac{K-2K\beta z^{-1}+Kz^{-2}}{1-2K\beta z^{-1}+(2K-1)z^{-2}}. \quad (14.23)$$

The 3 dB bandwidth is controlled by the parameter α,

$$B = \frac{1}{2\pi\Delta t}\cos^{-1}\left(\frac{2\alpha}{1+\alpha^2}\right) \approx \frac{(1-\alpha)}{2\pi\Delta t}, \tag{14.24}$$

and thus the Q is

$$Q = \frac{f_0}{B} \approx \frac{2\pi f_0 \Delta t}{1-\alpha}. \tag{14.25}$$

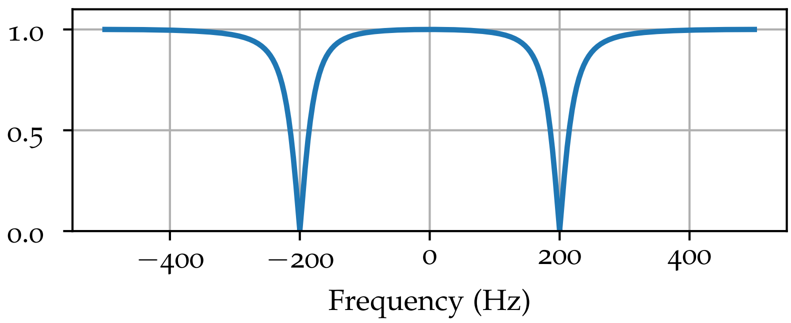


Figure 14.9: Notch filter with $f_0 = 200\,\text{Hz}$, $\alpha = 0.73$, $f_s = 1\,\text{kHz}$ giving a 3 dB bandwidth of 50 Hz and a Q of 4.

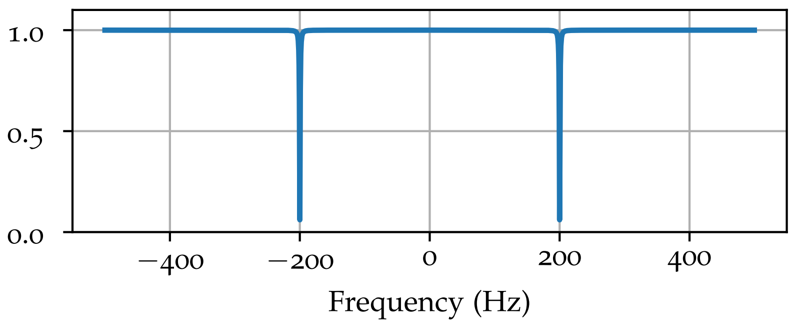


Figure 14.10: Notch filter with $f_0 = 200\,\text{Hz}$ and $\alpha = 0.99$ giving a 3 dB bandwidth of 1.6 Hz and a Q of 125. $a_0 = a_2 = 0.995$, $a_1 = b_1 = -0.615$, $b_0 = 1$, $b_2 = 0.99$.

A second-order recursive notch filter is also described by

$$H(z) = \frac{1 - 2\beta z^{-1} + z^{-2}}{1 - 2r\beta z^{-1} + r^2 z^{-2}}. \tag{14.26}$$

This is an approximation to (14.20) that is valid when r is close to 1, since $r = \sqrt{\alpha}$ and $2r \approx 1 + \alpha$.

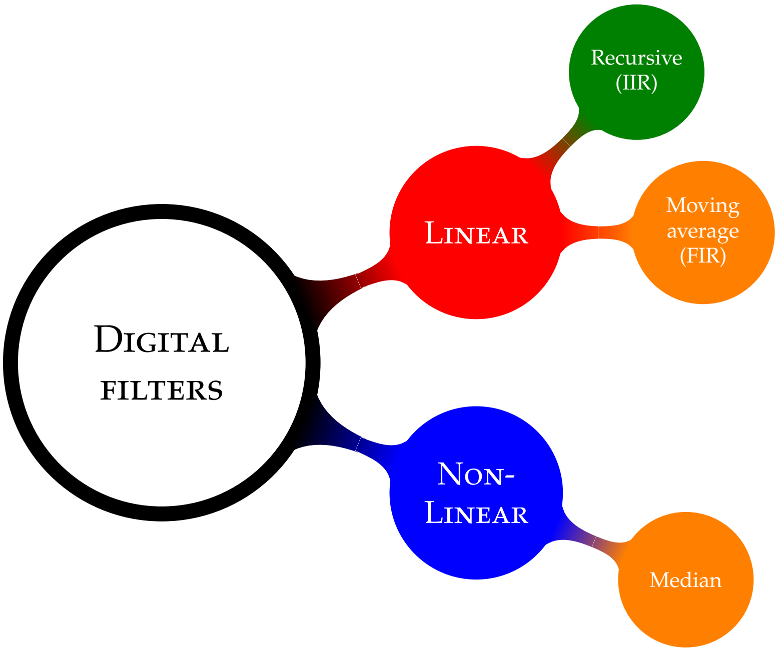


Figure 14.11: Categories of digital filters. There are many other filters such as adaptive filters that modify the filter weights.

15 Sampling

A continuous-time signal is converted to a discrete-time signal by an analogue to digital converter (ADC). For example, consider a continuous-time AC signal,

$$x(t) = A\cos\left(2\pi f_0 t\right), \qquad (15.1)$$

sampled at a frequency $f_s = 1/\Delta t$ where Δt is the sampling interval. The discrete-time signal is

$$x[n] = x(n\Delta t) = A\cos\left(2\pi f_0 n\Delta t\right). \qquad (15.2)$$

1 Sampling theorem

Intuitively, the sampling operation leads to a loss of information when compared with the original analogue signal $x(t)$. The smaller the sampling interval, Δt, the less this loss of information. However, if the signal $x(t)$ is band-limited to some finite frequency f_{max}, there is no loss of information, provided the sampling interval is sufficiently small. This is the essence of Shannon's sampling theorem.

The question is how small does Δt have to be to avoid a loss of information? This can be answered by considering the spectrum of the sampled signal.

1.1 Relationship between the continuous and discrete-time Fourier transforms

Consider the Fourier transform of a continuous signal:

$$X(f) = \int_{-\infty}^{\infty} x(t)\exp\left(-\mathrm{j}2\pi f t\right)\mathrm{d}t. \qquad (15.3)$$

The equivalent discrete-time Fourier transform is[1]

$$X_{\frac{1}{\Delta t}}(f) = \sum_{n=-\infty}^{\infty} x[n]\exp\left(-\mathrm{j}2\pi f n\Delta t\right). \qquad (15.4)$$

[1] This is equivalent to sampling the z-transform, $X(z)$, of a causal signal at $z = \exp\left(\mathrm{j}2\pi f\Delta t\right)$.

It is not straightforward to show, but the two transforms are related by

$$\boxed{X_{\frac{1}{\Delta t}}(f) = \frac{1}{\Delta t}\sum_{n=-\infty}^{\infty} X\left(f - \frac{n}{\Delta t}\right).} \tag{15.5}$$

This is a key result. It says that the DTFT is the sum of shifted copies (images) of the continuous Fourier transform, spaced by the sampling rate $f_s = 1/\Delta t$.

To recover $X(f)$ from $X_{\frac{1}{\Delta t}}(f)$ requires that $X(f) = 0$ for $|f| \geq 1/(2\Delta t)$ otherwise aliasing occurs. This is a frequency domain interpretation of the sampling theorem, see Figure 15.1.

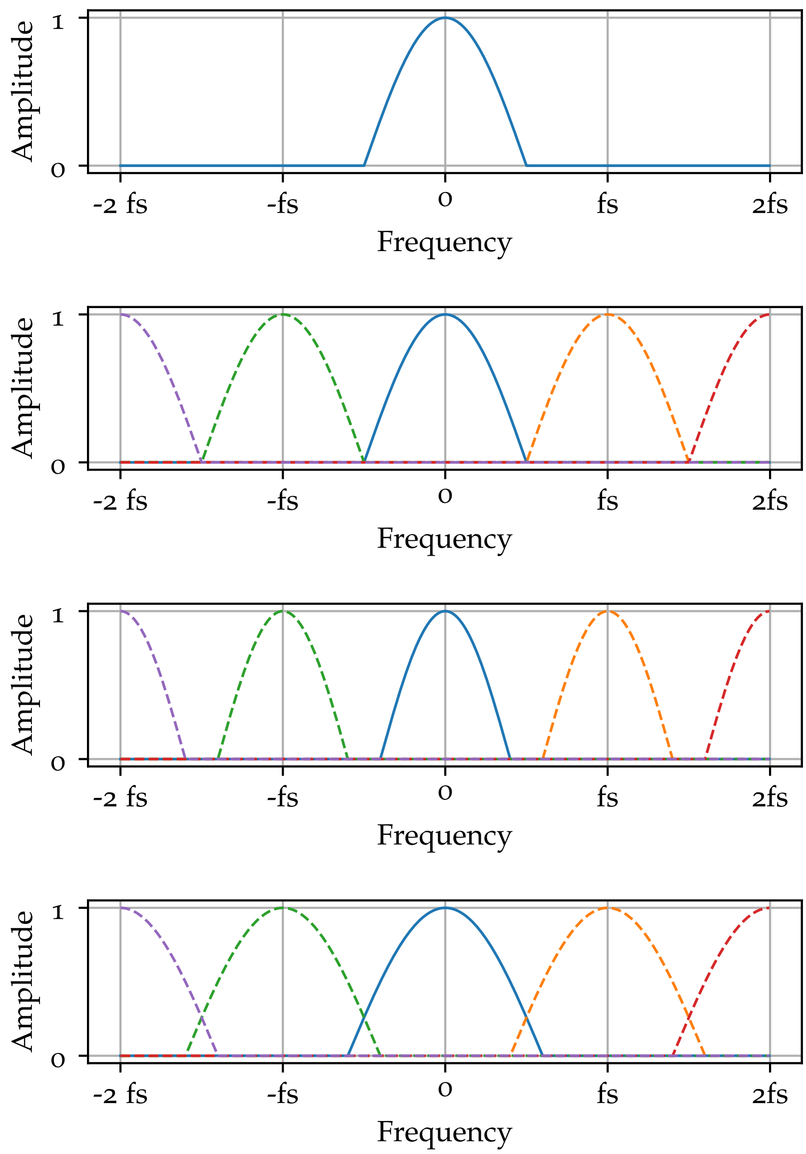


Figure 15.1: (a) Fourier transform $X(f)/\Delta t$, (b) DTFT Nyquist sampled, (c) DTFT oversampled, (d) DTFT undersampled. The undersampled spectrum cannot be recovered. Note, this example is contrived since a spectrum is never going to be bandlimited.

1.2 Nyquist rate

From Figure 15.1 it can be seen that when $f_s > 2f_{\max}$ then there is no overlap of the images. Thus $X(f)$ can be extracted by filtering the central image of $X_{\frac{1}{\Delta t}}(f)$ to recover $x(t)$ exactly. The critical sampling rate $f_s = 2f_{\max}$ is called the *Nyquist rate*. Sampling below this rate leads to an unrecoverable overlap of the images of $X(f)$, a process called *aliasing*, and thus $x(t)$ cannot be recovered[2].

[2] Without any prior information.

1.3 Anti-aliasing filter

Bandlimited continuous time signals can only be achieved with a brick-wall filter. However, since a brick-wall filter is not realisable, analogue signals are never strictly band-limited. Thus there is always a level of aliasing, see Figure 15.2.

In practice, *signals are usually sampled well above the Nyquist limit*, by a factor of 10 or more[3]. This allows the use of less sharp anti-aliasing filters and thus a filter with fewer poles giving less phase distortion.

[3] They can then be digitally filtered and decimated to give a sampling rate closer to the Nyquist limit.

For a control system, it is important to minimise any controller lag. Thus it is common to oversample by a factor of at least twenty.

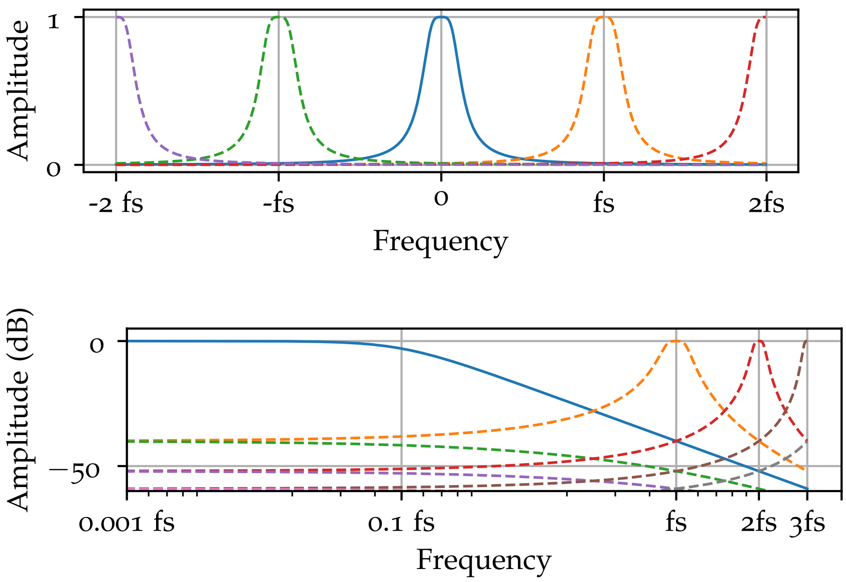
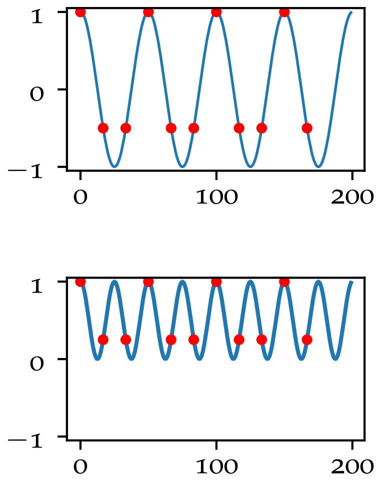


Figure 15.2: Spectrum with a second-order Butterworth response showing aliasing.



1.4 *Aliasing recovery*

Once a signal is aliased it cannot be recovered without some prior knowledge of the signal. For example, consider squaring a bandlimited signal $x(t)$ with maximum frequency $f_{\max}$ sampled just above the Nyquist rate:

$$y(t) = x^2(t). \tag{15.6}$$

The resultant signal, $y(t)$, has frequency components up $2f_{\max}$ and thus $y(t)$ will not be adequately sampled. However, we can recover $x(t)$ from $y(t)$ (provided $x(t)$ is positive to avoid any ambiguity when performing the square root operation).

1.5 *Sample-hold*

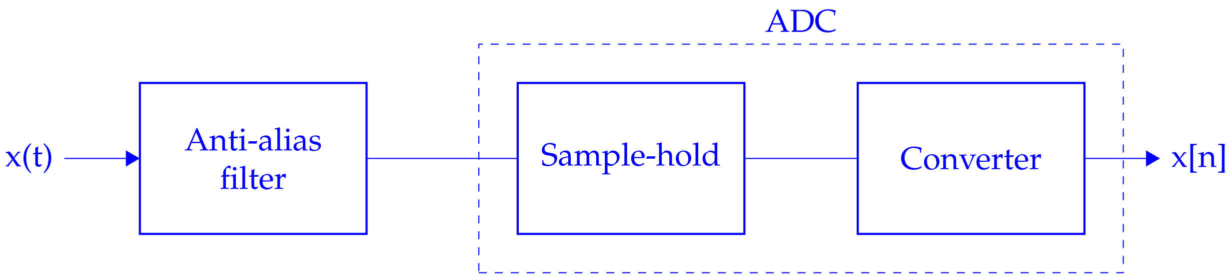


Figure 15.3: Signal sampling using an anti-alias filter and an ADC.

It is not possible for an ADC to instantaneously sample a signal; it takes time to perform the conversion. During this time the signal can change. To reduce this effect a sample-hold circuit is used prior to the ADC. But even this circuit cannot instantaneously sample a signal; it essentially averages the signal over a short period called the aperture time. This averaging can be described by

$$x_s[n] = x_s(n\Delta t) = \frac{1}{T_A}\int_0^{T_A} x(n\Delta t - t)\mathrm{d}t, \tag{15.7}$$

where T_A is the aperture time. This integral can be expressed as a convolution

$$x_s[n] = \int_{-\infty}^{\infty} h(t)x(n\Delta t - t)\mathrm{d}t, \tag{15.8}$$

where the impulse response is

$$h(t) = \frac{1}{T_A}\,\mathrm{rect}\left(\frac{t - \frac{T_A}{2}}{T_A}\right). \tag{15.9}$$

Using the convolution theorem, the averaging is equivalent to a filter, where

$$X_s(f) = H(f)X(f), \tag{15.10}$$
$$= \operatorname{sinc}(fT_A)\exp(-j\pi f T_A)\,X(f). \tag{15.11}$$

The effect of this filtering is reduced with a short aperture time.

2 Signal reconstruction

Recovering a continuous time signal, $x(t)$, from a discrete-time signal, $x[n]$, requires sinc interpolation[4]:

[4] This equation is called the Whittker-Shannon formula.

$$x(t) = \sum_{n=-\infty}^{\infty} x[n]\operatorname{sinc}\left(\frac{t - n\Delta t}{\Delta t}\right). \tag{15.12}$$

In the frequency domain, this is equivalent to brick-wall filtering.

The reconstruction can be performed using a digital to analogue converter (DAC) followed by a reconstruction filter[5]. The reconstruction filter should be a brick-wall filter but this is not realisable. Instead, the signal should be oversampled to minimise reconstruction errors with a realisable analogue filter.

[5] A compensating filter is also required for zero-order-hold sampling.

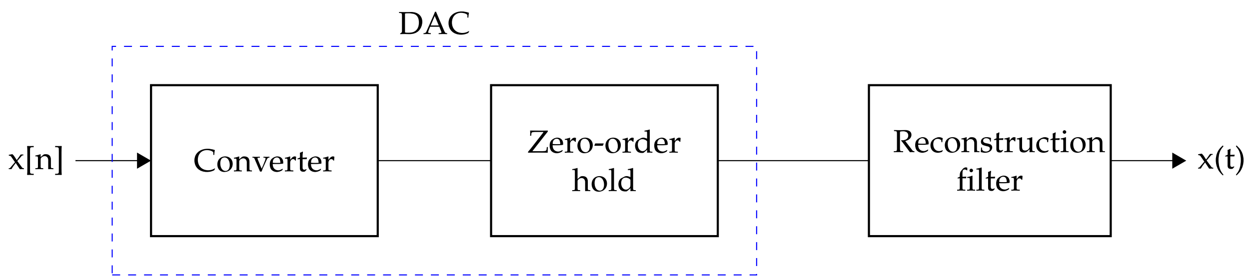


Figure 15.4: Signal reconstruction from its samples using a DAC and reconstruction filter.

2.1 Reconstruction filter

The goal of the reconstruction filter is to recover $X(f)$ from $X_{\frac{1}{\Delta t}}(f)$ by removing the spectral images. This requires an analogue filter. Like an anti-aliasing filter, this cannot do a perfect job since it is impossible to build a brick-wall filter. Again, oversampling is important to simplify the requirements for this filter.

2.2 *Zero-order-hold sampling*

A DAC produces a staircase representation of the desired analogue signal using a *zero-order-hold* device. The zero-order-hold device holds the sample value constant until the next sample is output Δt later.

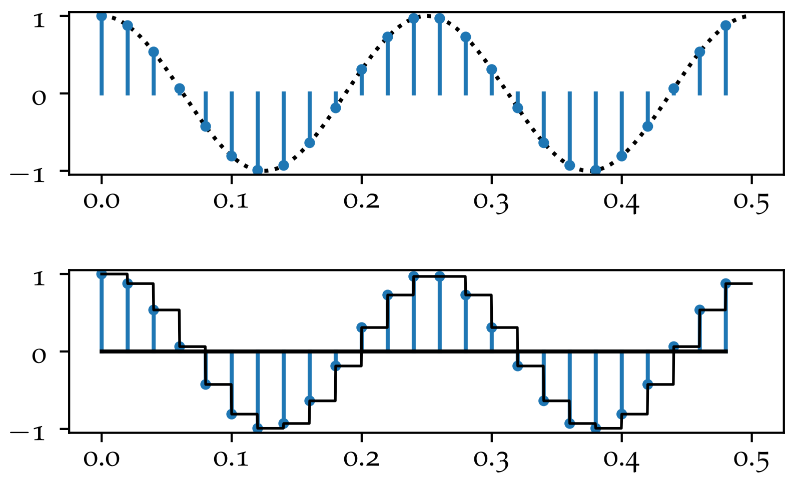


Figure 15.5: (a) Lollipop plot of sampled sinewave. (b) Staircase representation of a signal produced by a DAC followed by a zero-order hold.

The effect of the staircase approximation is to tweak the spectrum by

$$X_{\mathrm{ZOH}}(f) = \exp\left(-\mathrm{j}\pi f \Delta t\right) \operatorname{sinc}(f\Delta t) X_{\frac{1}{\Delta t}}(f). \tag{15.13}$$

To recover $X_{\frac{1}{\Delta t}}(f)$ without distortion, a *compensating filter* is used to correct for the $\operatorname{sinc}(f\Delta t)$ factor.

3 *Ideal sampling*

The discrete Fourier transform is the infinite sum of shifted copies of the continuous Fourier transform

$$X_{\frac{1}{\Delta t}}(f) = \sum_{n=-\infty}^{\infty} X\left(f - \frac{n}{\Delta t}\right) \tag{15.14}$$

This can be represented by the convolution of the Fourier transform with a comb of Dirac deltas,

$$X_{\frac{1}{\Delta t}}(f) = X(f) * \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{\Delta t}\right). \tag{15.15}$$

The Fourier transform of a comb of Dirac deltas is another comb of Dirac deltas,

$$\Delta t \sum_{n=-\infty}^{\infty} \delta\left(t - n\Delta t\right) \longleftrightarrow \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{\Delta t}\right), \tag{15.16}$$

and thus using the convolution theorem, the inverse Fourier transform of (15.15) is

$$x_I(t) = \mathcal{F}^{-1}\left\{X_{\frac{1}{\Delta t}}(f)\right\} = \Delta t \sum_{n=-\infty}^{\infty} x(n\Delta t)\delta(t - n\Delta t), \tag{15.17}$$

$x_I(t)$ is a weighted sum of Dirac deltas and is called the ideal sampled signal. Note, this is just a *mathematical signal representation*; it is not physically realisable and does not approximate the analogue signal. Moreover, it has infinite energy (a squared impulse has infinite energy).

Since an impulse is zero is zero except when its argument is zero, (15.17) can be expressed as a product of the analogue signal with a Dirac comb (or impulse train),

$$x_I(t) = x(t)\Delta t \sum_{n=-\infty}^{\infty} \delta(t - n\Delta t). \tag{15.18}$$

Texts use this expression to say that an ideal sampled signal is produced by multiplying the analogue signal with a Dirac comb. However, this is not how a sampler works in practice. Nevertheless, the ideal sampled signal is useful since the Fourier transform of $x_I(t)$ gives the same result as the DTFT of $x[n]$.

4 *Aperture time*

Ideal sampling requires an infinitesimally short time during which the signal is sampled. Not surprising, this is unachievable in practice. The finite time required to sample a signal is called the the *aperture time*. The greater the aperture time the greater the amplitude uncertainty.

The maximum aperture time, $\Delta\tau$, that is tolerable to sample a sinewave of frequency $f_{\max}$ to N bits of resolution is given by

$$\Delta\tau < \frac{1}{2\pi f_{\max} 2^N}. \tag{15.19}$$

For example, to digitise a 1 kHz signal to 10 bits (0.1%), requires

$$\Delta\tau < \frac{1}{2\pi \times 1000 \times 2^{10}} = 155\,\text{ns}. \tag{15.20}$$

Even moderate resolution appears to require a very fast ADC. However, this can be overcome by using a fast sample and hold circuit that feeds a slower ADC.

5 Derivation of the DTFT

The discrete-time Fourier transform (DTFT) is defined as

$$X_{\frac{1}{\Delta t}}(f) = \sum_{n=-\infty}^{\infty} x[n] \exp\left(-\mathrm{j}2\pi f n\Delta t\right), \tag{15.21}$$

$$= \sum_{n=-\infty}^{\infty} x(n\Delta t) \exp\left(-\mathrm{j}2\pi f n\Delta t\right). \tag{15.22}$$

To relate this to the continuous Fourier transform of $x(t)$ we can represent $x(t)$ with an inverse Fourier transform,

$$x(t) = \int_{-\infty}^{\infty} X(f) \exp\left(\mathrm{j}2\pi f t\right) \mathrm{d}f. \tag{15.23}$$

Substituting this into (15.22) gives

$$X_{\frac{1}{\Delta t}}(f) = \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} X(\nu) \exp\left(\mathrm{j}2\pi \nu n\Delta t\right) \mathrm{d}\nu \exp\left(-\mathrm{j}2\pi f n\Delta t\right). \tag{15.24}$$

Now some magic is needed using Dirac combs.

6 Zero-order-hold sampling

The zero-order-hold sampled signal can be expressed as

$$x_{\mathrm{ZOH}}(t) = \sum_{n=-\infty}^{\infty} x[n] \operatorname{rect}\left(\frac{t - \frac{\Delta t}{2} - n\Delta t}{\Delta t}\right). \tag{15.25}$$

Using a continuous Fourier transform, this has a spectrum,

$$X_{\mathrm{ZOH}}(t) = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[n] \operatorname{rect}\left(\frac{t - \frac{\Delta t}{2} - n\Delta t}{\Delta t}\right) \exp\left(-\mathrm{j}2\pi f t\right) \mathrm{d}t. \tag{15.26}$$

Exchanging the order of summation and integration gives,

$$X_{\mathrm{ZOH}}(t) = \sum_{n=-\infty}^{\infty} x[n] \int_{-\infty}^{\infty} \operatorname{rect}\left(\frac{t - \frac{\Delta t}{2} - n\Delta t}{\Delta t}\right) \exp\left(-\mathrm{j}2\pi f t\right) \mathrm{d}t, \tag{15.27}$$

$$= \Delta t \operatorname{sinc}(f\Delta t) \exp\left(-\mathrm{j}\pi f \Delta t\right) \sum_{n=-\infty}^{\infty} x[n] \exp\left(-\mathrm{j}2\pi f n\Delta t\right), \tag{15.28}$$

$$= \Delta t \operatorname{sinc}(f\Delta t) \exp\left(-\mathrm{j}\pi f \Delta t\right) X_{\frac{1}{\Delta t}}(f). \tag{15.29}$$

Alternatively, the zero-order-hold sampled signal can be regarded as the convolution of an ideally sampled signal with a rectangle function of width Δt,

$$x_{\mathrm{ZOH}}(t) = x_I(t) * \mathrm{rect}\left(\frac{t-\frac{\Delta t}{2}}{\Delta t}\right), \tag{15.30}$$

$$= \sum_{n=-\infty}^{\infty} x(n\Delta t)\,\mathrm{rect}\left(\frac{t-\frac{\Delta t}{2}-n\Delta t}{\Delta t}\right). \tag{15.31}$$

$$\tag{15.32}$$

This has a spectrum,

$$X_{\mathrm{ZOH}}(f) = \exp\left(-\mathrm{j}\pi f\Delta t\right)\mathrm{sinc}(f\Delta t)X_I(f), \tag{15.33}$$

which can be seen to distort the spectrum of $X_I(f)$. To recover $X_I(f)$ without distortion, a *compensating filter* is used to correct for the $\mathrm{sinc}(f\Delta t)$ factor.

16
Discrete Fourier Transform

The algorithm to compute a spectrum on a computer is the discrete Fourier transform (DFT). This produces a sampled spectrum $X[k]$, where k is the frequency sample index from a discrete signal $x[n]$ where n is the time sample index. The spectrum at other frequencies can be found using interpolation.

The relationship of the DFT to the Fourier transforms is shown in Figure 16.1. The key thing to note is that a sampled quantity transforms to a periodic quantity.

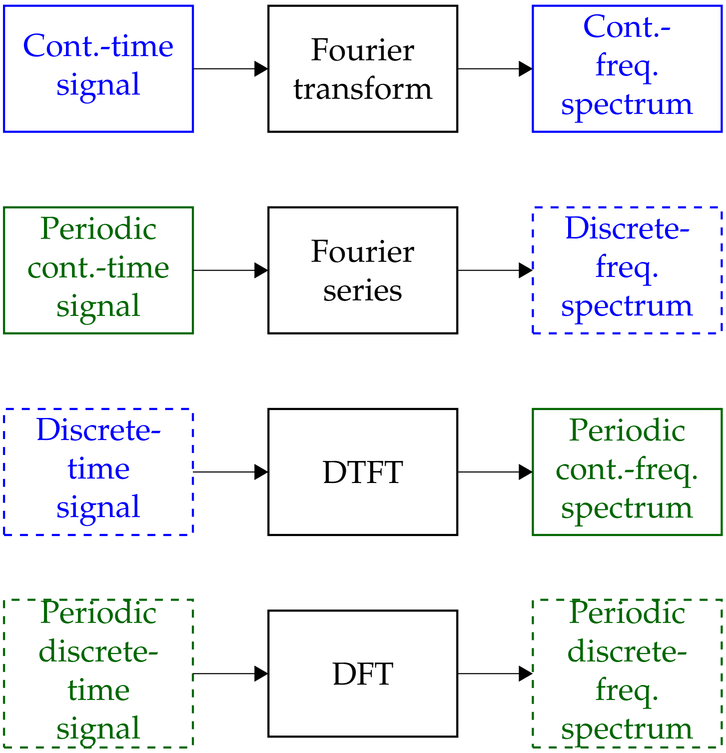


Figure 16.1: Comparison of Fourier transforms.

1 *The discrete Fourier transform (DFT)*

To represent a spectrum on a computer we need to sample the continuous spectrum[1]. Fortunately, if the signal has a finite extent and if its spectrum is adequately sampled, the continuous spectrum can be recovered from the discrete spectrum using interpolation.

[1] In addition to sampling the signal in the time-domain.

The DFT takes N signal samples and produces N frequency samples, using[2]

[2] This is a coordinate transform.

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(-j2\pi\frac{nk}{N}\right). \tag{16.1}$$

Here $x[n]$ is the discrete-time signal and $X[k]$ is its discrete-frequency spectrum. In comparison, the DTFT is

$$X_{\frac{1}{\Delta t}}(t) = \sum_{n=-\infty}^{\infty} x[n] \exp\left(-j2\pi f n \Delta t\right). \tag{16.2}$$

The DFT differs from the DTFT in two ways:

1. It truncates the signal to a duration of $T = N\Delta t$.
2. It samples the spectrum with a spacing $\Delta f = 1/(N\Delta t)$.

The discrete frequencies are evenly spaced over the band $[-f_s/2, f_s/2)$ where $f_s = 1/\Delta t$. Thus the frequency sample index, k, corresponds to a frequency, f by

$$f = k\Delta f = k\frac{1}{N\Delta t}. \tag{16.3}$$

2 *The fast Fourier transform (FFT)*

The DFT requires N complex multiplications per frequency sample k. With N frequency samples, the DFT requires N^2 complex multiplications. Thus the DFT becomes infeasible to compute for large N.

In practice, the DFT is computed with an efficient algorithm called the Fast Fourier Transform (FFT). This requires approximately $N \log_2 N$ complex multiplications. The most efficient FFT algorithms require N to be a power of two[3].

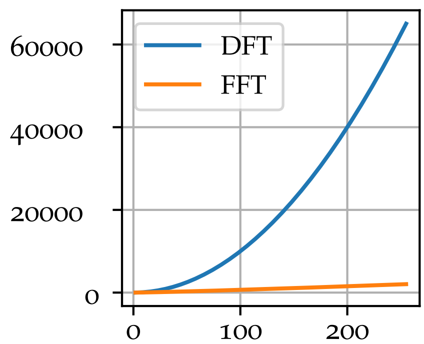


Figure 16.2: Computation comparison between DFT and FFT.

[3] An FFT can be calculated efficiently if the prime factors of the size N are small.

3 Calculating a spectrum with the FFT

Consider the discrete-signal described by

$$s[n] = \cos(2\pi f_0 n \Delta t) \quad n = 0, 1, \cdots 127, \qquad (16.4)$$

where $f_0 = 20\,\text{Hz}$ and $\Delta t = 2\,\text{ms}$. This is plotted in Figure 16.3 along with the magnitude of the positive part of its spectrum calculated with a 50 point FFT.

The sampling frequency is $f_s = 1/\Delta t = 500\,\text{Hz}$ and thus the maximum frequency is $f_s/2 = 250\,\text{Hz}$. The spacing between the frequency samples is $f_s/N = 500/50 = 10\,\text{Hz}$.

Figure 16.4 shows a similar signal but with a frequency of $f_0 = 25\,\text{Hz}$. The peak of its spectrum falls between two frequency samples but notice all the other frequency components that have appeared.

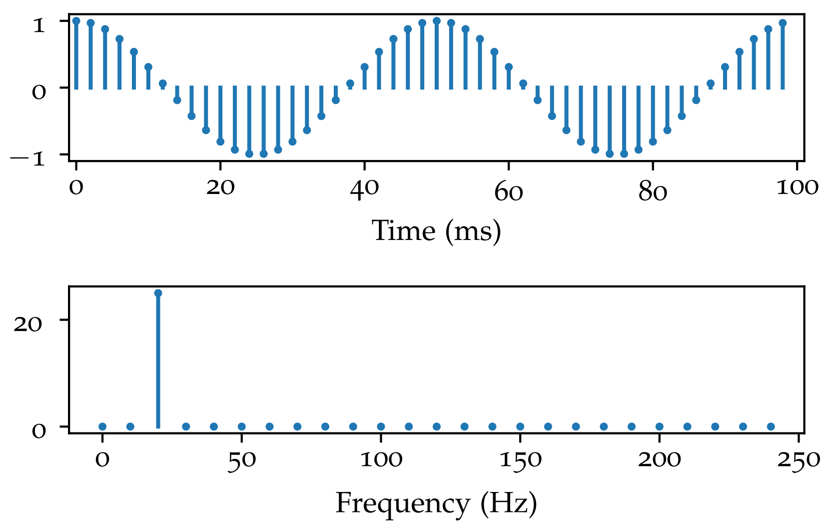


Figure 16.3: Lollipop plot of a 20 Hz signal sampled at 500 Hz and the magnitude of its 50 point DFT.

4 Zero-padding

If a signal is sampled at a frequency f_s for N samples, the DFT constrains the spectral frequencies to a spacing of f_s/N. If we want the spectrum at a finer spacing, we can zero-pad the sequence to a length M. The resulting frequency spacing is now f_s/M.

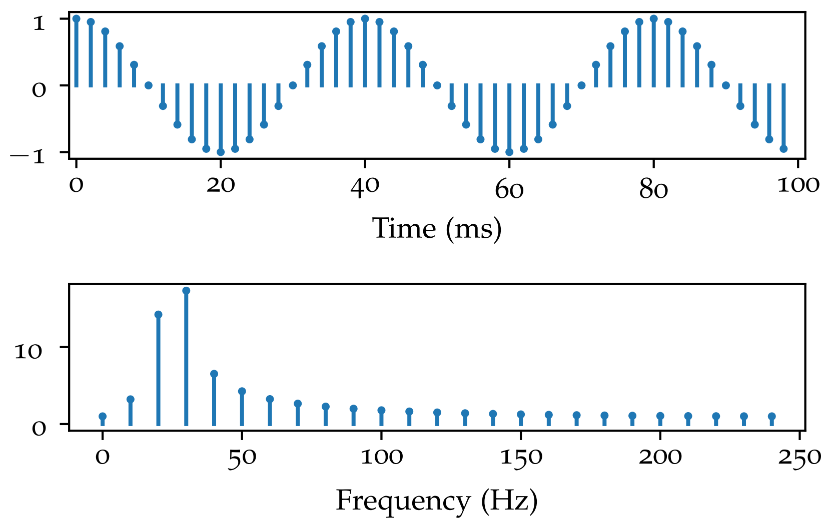


Figure 16.4: Lollipop plot of a 25 Hz signal sampled at 500 Hz and the magnitude of its 50 point DFT. Note all the non-zero frequency components.

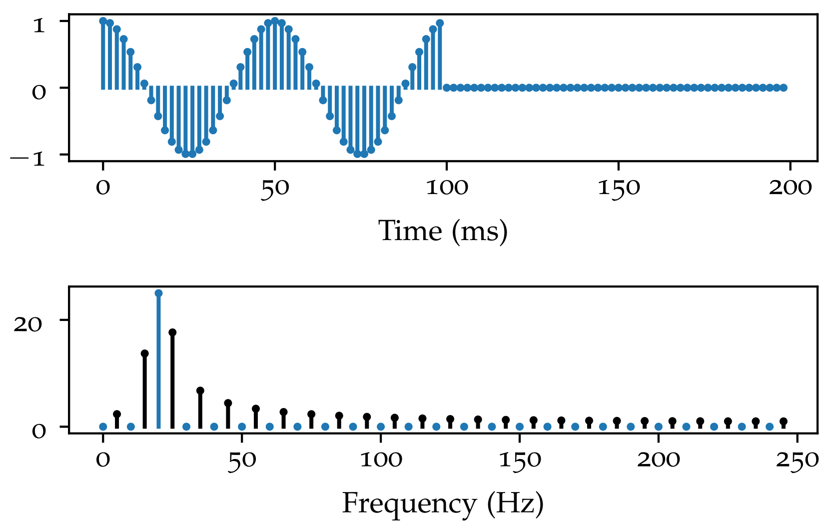


Figure 16.5: Lollipop plot of a 20 Hz signal sampled at 500 Hz, zero-padded by a factor of 2, and the magnitude of its 100 point DFT. The black samples are the additional samples not visible in Figure 16.3.

5 *Windowing*

The DFT considers a truncated portion of the signal of interest. Mathematically, this can be represented by[4]

$$g[n] = s[n]w[n]. \tag{16.5}$$

[4] The equivalent continuous time operation is $g(t) = s(t)w(t)$.

Here $s[n]$ denotes the untruncated discrete-time signal, $g[n]$ denotes the truncated signal, and $w[n]$ denotes the truncation function[5]. This is shown in Figure 16.8.

[5] Usually called a window.

The truncation of the signal produces a blurred spectrum. Using the convolution theorem, the DTFT of the truncated signal can be represented as[6]

$$G_{\frac{1}{\Delta t}}(f) = S_{\frac{1}{\Delta t}}(f) * W_{\frac{1}{\Delta t}}(f). \tag{16.6}$$

[6] The equivalent continuous time operation is $G(f) = S(f) * W(f)$.

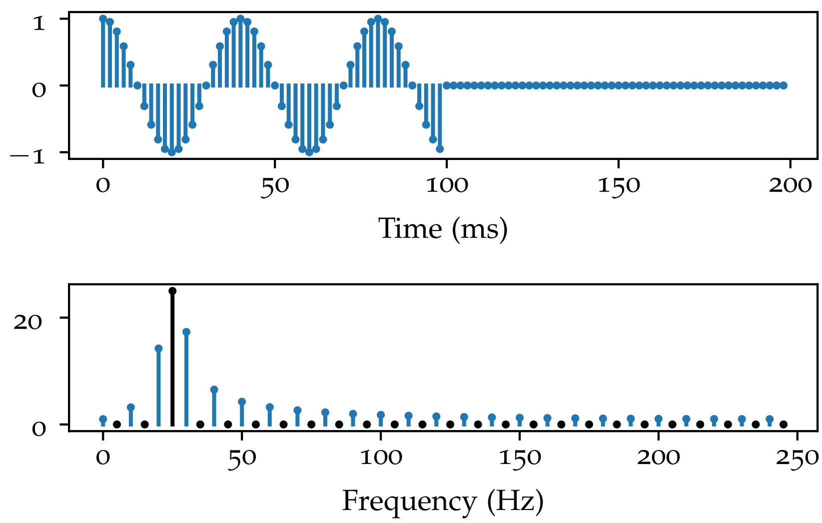


Figure 16.6: Lollipop plot of a 25 Hz signal sampled at 500 Hz, zero-padded by a factor of 2, and the magnitude of its 100 point DFT. Note this is similar to Figure 16.5 but shifted 5 Hz.

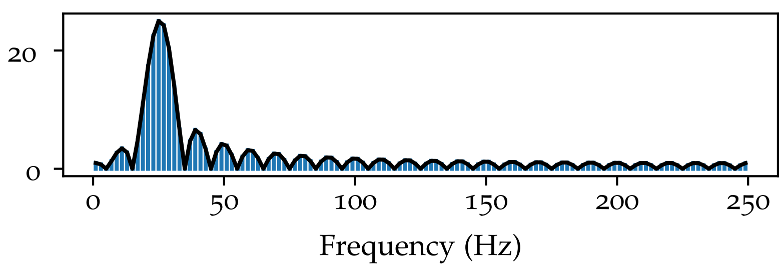


Figure 16.7: Lollipop plot of spectrum of a 20 Hz signal sampled at 500 Hz and zero-padded by a factor of 10.

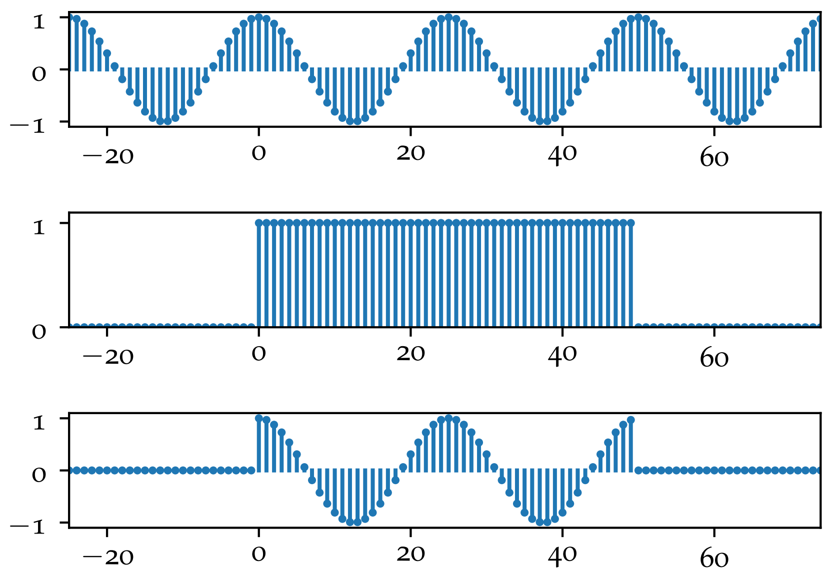


Figure 16.8: Truncation of a signal with a rectangular window.

Thus the desired spectrum has been convolved with the DTFT of the truncation function.

The most common truncation function is a rectangular function[7],

$$w[n] = \mathrm{rect}\left(\frac{n}{N} - 0.5\right). \tag{16.7}$$

[7] The equivalent continuous time signal is $w(t) = \mathrm{rect}(t/T - 0.5)$.

This has the DTFT

$$W_{\frac{1}{\Delta t}}(f) = \frac{\sin(N\pi f \Delta t)}{\sin(\pi f \Delta t)}. \tag{16.8}$$

Note, to get a narrower width and thus more frequency resolution, more samples are required to increase the duration, T.

While the sinc function has the narrowest mainlobe, it many sidelobes that slowly decay away. These sidelobes can mask other small frequency components of interest. To reduce the sidelobe level it is necessary to use a different truncation function that tapers more slowly to zero at both ends. Such a function is called a *window*. The down-side is that the spectral mainlobe becomes wider: the slower the taper, the lower the sidelobe level but the larger the spectral mainlobe width.

A window with a reasonable compromise between spectral mainlobe width and sidelobe level is the Hamming window[8],

[8] There are many window functions commonly used.

$$w[n] = 0.54 - 0.46\cos\left(2\pi\frac{n}{N}\right) \qquad n = 0, 1, \cdots N-1. \tag{16.9}$$

This window is plotted in Figure 16.9 with a rectangular and triangular window for comparison.

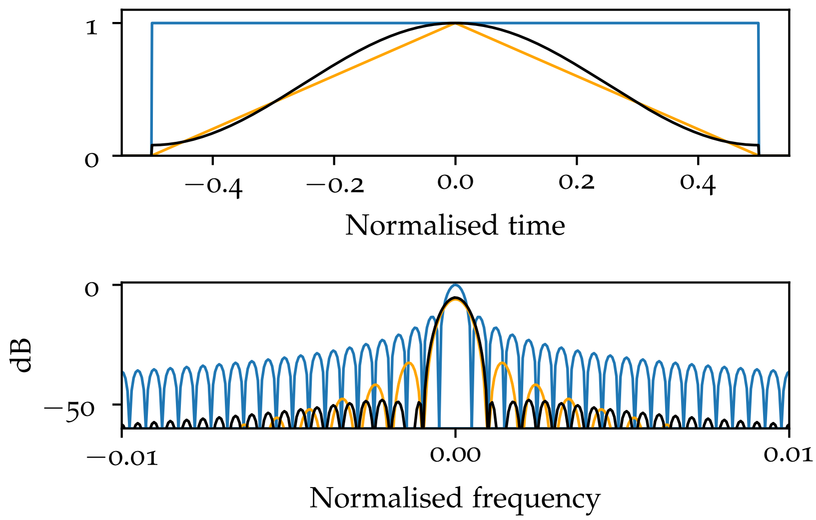


Figure 16.9: Rectangular, triangular, and Hamming windows with the magnitude of their DFTs. Note how a smoother window causes the sidelobes to decay more rapidly.

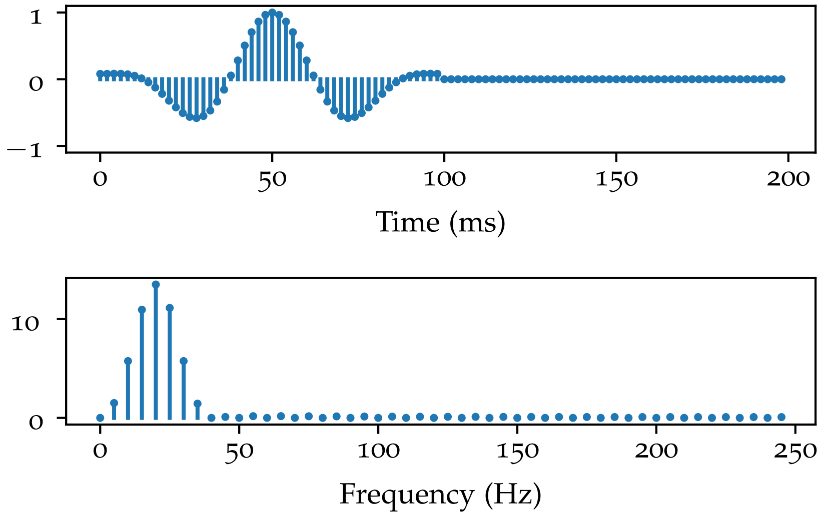


Figure 16.10: Lollipop plot of a 20 Hz signal sampled at 500 Hz, Hamming windowed, zero-padded by a factor of 2, and the magnitude of its 100 point DFT. Note, the window suppresses the sidelobes but increases the width of the mainlobe.

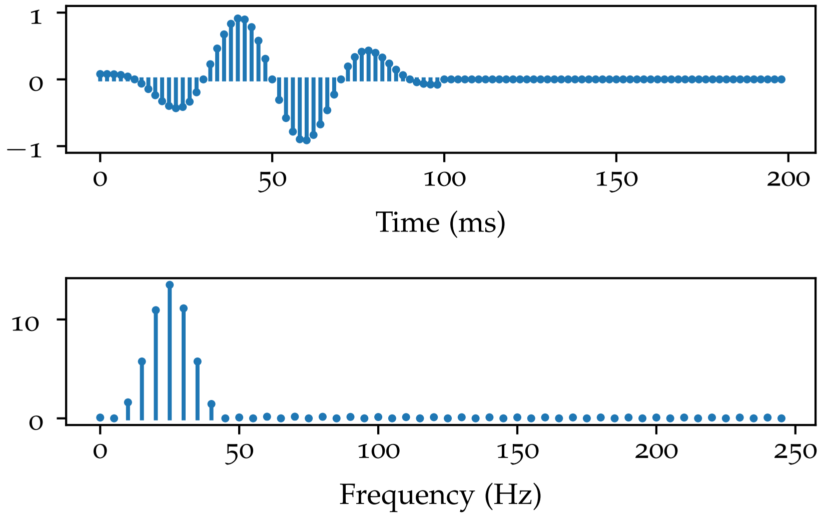


Figure 16.11: Lollipop plot of a 25 Hz signal sampled at 500 Hz, Hamming windowed, zero-padded by a factor of 2, and the magnitude of its 100 point DFT.

5.1 Convolution with the FFT

For a large convolution size, N, convolutions can be computed much more efficiently using FFTs. From the convolution theorem,

$$x(t) * h(t) \longleftrightarrow X(f)H(f), \tag{16.10}$$

the convolution of two signals can be found from the inverse Fourier transform of the product of their Fourier transforms. The same applies to discrete signals but with a twist:

$$x[n] \circledast h[n] \longleftrightarrow X[k]H[k]. \tag{16.11}$$

The twist is that a *circular convolution*[9] is performed. If the extent of $x[n]$ is N_x samples and the extent of $h[n]$ is N_h samples then the convolution has a length $N_x + N_h - 1$. Thus to avoid circular affects the FFT size must satisfy $N \geq N_x + N_h - 1$. This is achieved by zero-padding both $x[n]$ and $h[n]$.

[9] This is why the convolution symbol is inside a circle.

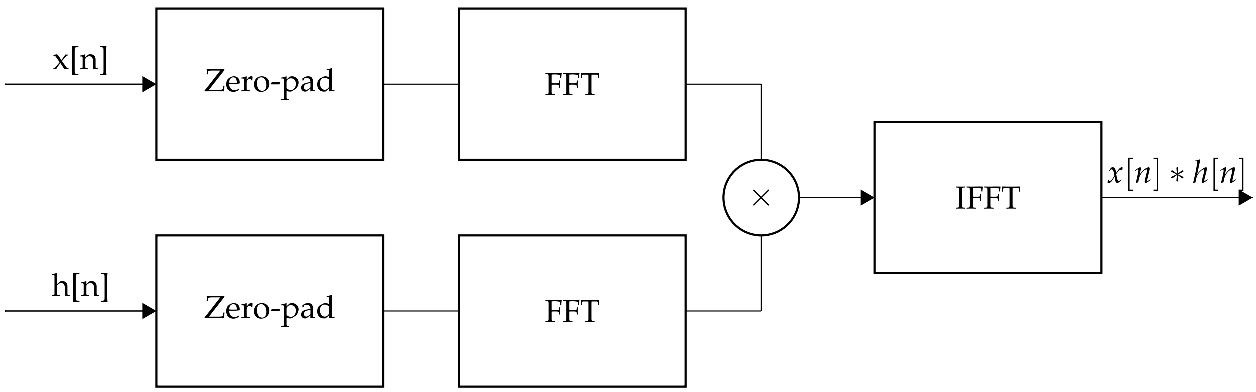


Figure 16.12: Efficient convolution using FFTs.

6 Inverse DFT

The inverse DFT is similar to the forward DFT

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \exp\left(\mathrm{j}2\pi \frac{nk}{N}\right). \tag{16.12}$$

The difference is a scale factor $1/N$ and a change of sign for the complex exponential.

```
from __future__ import division
from numpy.fft import fft
from numpy import arange, cos, pi
from matplotlib.pyplot import subplots, show

# Data points
N = 50
# Sampling frequency
fs = 500
# Sampling interval
dt = 1 / fs

# Signal frequency
f0 = 20

n = arange(N)
t = n * dt
s = cos(2 * pi * f0 * t)

S = fft(s)
# Consider positive frequencies only
S = S[0: N // 2]
f = arange(N // 2) * fs / N

# Plot signal and its spectrum
fig, axes = subplots(2)
axes[0].stem(t * 1e3, s)
axes[0].set_xlabel('Time (ms)')
axes[0].set_xlim(-1, 101)
axes[0].set_ylim(-1.1, 1.1)

axes[1].stem(f, abs(S))
axes[1].set_xlabel('Frequency (Hz)')
```

Listing 16.1: Example script for DFT of a cosine.

```python
from __future__ import division
from numpy.fft import fft
from numpy import arange, cos, pi, zeros
from matplotlib.pyplot import subplots, show

# Signal data points
Ns = 50
# Zeropad factor
M = 2
# Sampling frequency
fs = 500
# Sampling interval
dt = 1 / fs

# Signal frequency
f0 = 20

# FFT data points
N = Ns * M

# Create signal of Ns samples and zeropad to N samples
s = zeros(N)
n = arange(N)
t = n * dt
s[0:Ns] = cos(2 * pi * f0 * t[0:Ns])

# Create and apply Hamming window
w = 0.54 - 0.46 * cos(2 * pi * n / Ns)
y = s * w

Y = fft(y)
# Consider positive frequencies only
Y = Y[0: N // 2]
f = arange(N // 2) * fs / N

# Plot signal and its spectrum
fig, axes = subplots(2)
axes[0].stem(t * 1e3, y)
axes[0].set_xlabel('Time (ms)')
axes[0].set_ylim(-1.1, 1.1)

axes[1].stem(f, abs(Y))
axes[1].set_xlabel('Frequency (Hz)')
```

Listing 16.2: Example script for DFT of a cosine with Hamming window, zero-padded by a factor of 2.

7 Sampling

The DFT (and FFT) converts a discrete-time signal to a discrete-frequency spectrum; thus both the time and frequency domains are sampled.

7.1 Time domain sampling

The sampling theorem states that a bandlimited signal can be perfectly reconstructed from its samples provided the sampling rate is greater than twice the signal bandwidth. A bandlimited signal implies that the spectrum $S(f)$ of a signal $s(t)$ is zero for frequencies outside a band, i.e.,

$$S(f) = \begin{cases} 0 & f < f_{\text{min}}, \\ 0 & f > f_{\text{max}}. \end{cases} \tag{16.13}$$

In this case the bandwidth is $B = f_{\text{max}} - f_{\text{min}}$ in which case the sampling theorem[10] requires that $f_s > 2B$. Since $f_s = 1/\Delta t$, the sampling requirement is

[10] Usually $f_{\text{min}} = 0$ and thus $f_s > 2f_{\text{max}}$.

$$\Delta t < \frac{1}{2B}. \tag{16.14}$$

In theory, the original signal can be perfectly recovered using interpolation. However, in practice, a signal is not bandlimited and there is some aliasing that prevents perfect reconstruction.

7.2 Frequency domain sampling

Analogous to time domain sampling a spectrum can be sampled. In this case we require that the signal $s(t)$ is limited in duration, i.e.,

$$s(t) = \begin{cases} 0 & t < t_{\text{min}}, \\ 0 & t > t_{\text{max}}. \end{cases} \tag{16.15}$$

The duration of the signal is $T = t_{\text{max}} - t_{\text{min}}$. To adequately sample the spectrum requires that

$$\Delta f < \frac{1}{2T}. \tag{16.16}$$

7.3 Time and frequency domain sampling

In practice, a signal cannot be both band and duration limited. Thus there is always some undersampling in the time and/or frequency domains. For example, consider an ideal band-limited spectrum

$$S(f) = \mathrm{rect}\left(\frac{f}{B}\right). \tag{16.17}$$

From an inverse Fourier transform, the signal is a sinc function

$$s(t) = B\,\mathrm{sinc}(Bt). \tag{16.18}$$

This function gets smaller but never goes to zero and thus is not duration limited.

8 DFT as a matrix multiply

The DFT

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(-\mathrm{j}2\pi\frac{nk}{N}\right), \tag{16.19}$$

can be written as

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N kn \tag{16.20}$$

where

$$W_N^{kn} = \exp\left(-\mathrm{j}2\pi\frac{nk}{N}\right) \tag{16.21}$$

is called the twiddle factor. Dropping the subscript on the twiddle factor for clarity, the DFT can be written as a matrix multiplication,

$$\begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} W^{00} & W^{10} & \dots & W^{N-1,0} \\ W^{01} & W^{11} & \dots & W^{N-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ W^{0,N-1} & W^{1,N-1} & \dots & W^{N-1,N-1} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} \tag{16.22}$$

Thus a DFT requires N^2 complex multiplications.

9 DFT of a complex exponential

Consider a complex exponential of unit amplitude and frequency f_0,

$$s(t) = \exp\left(\mathrm{j}2\pi f_0 t\right). \tag{16.23}$$

If this is sampled at an interval Δt, the equivalent discrete-time signal is

$$s[n] = \exp\left(\mathrm{j}2\pi f_0 n \Delta t\right). \tag{16.24}$$

The DFT is given by

$$
\begin{aligned}
S[k] &= \frac{1}{N}\sum_{n=0}^{N-1} s[n] \exp\left(-\mathrm{j}2\pi\frac{nk}{N}\right), && (16.25)\\
&= \frac{1}{N}\sum_{n=0}^{N-1} \exp\left(\mathrm{j}2\pi f_0 n\Delta t\right) \exp\left(-\mathrm{j}2\pi\frac{nk}{N}\right), && (16.26)\\
&= \frac{1}{N}\sum_{n=0}^{N-1} \exp\left(-\mathrm{j}2\pi n\left(\frac{k}{N} - f_0\Delta t\right)\right) && (16.27)
\end{aligned}
$$

This is the sum of a geometric series. This can be seen by writing it as

$$S[k] = \sum_{n=0}^{N-1} r^n, \tag{16.28}$$

where

$$r = \exp\left(-\mathrm{j}2\pi\left(\frac{k}{N} - f_0\Delta t\right)\right). \tag{16.29}$$

Provided $r \neq 1$, the sum of the geometric series is

$$S[k] = \frac{1-r^n}{1-r}. \tag{16.30}$$

Substituting $r = \exp\left(-\mathrm{j}2a\right)$, where $a = \pi\left(k/N - f_0\Delta t\right)$, gives

$$
\begin{aligned}
S[k] &= \sum_{n=0}^{N-1} \exp\left(-\mathrm{j}2an\right), && (16.31)\\
&= \frac{1-\exp\left(-\mathrm{j}2aN\right)}{1-\exp\left(-\mathrm{j}2a\right)}, && (16.32)\\
&= \frac{\exp\left(-\mathrm{j}aN\right)\exp\left(\mathrm{j}aN\right) - \exp\left(-\mathrm{j}aN\right)\exp\left(-\mathrm{j}aN\right)}{\exp\left(-\mathrm{j}a\right)\exp\left(\mathrm{j}a\right) - \exp\left(-\mathrm{j}a\right)\exp\left(-\mathrm{j}a\right)}, && (16.33)\\
&= \frac{\exp\left(-\mathrm{j}aN\right)\left(\exp\left(\mathrm{j}aN\right) - \exp\left(-\mathrm{j}aN\right)\right)}{\exp\left(-\mathrm{j}a\right)\left(\exp\left(\mathrm{j}a\right) - \exp\left(-\mathrm{j}a\right)\right)}, && (16.34)\\
&= \exp\left(-\mathrm{j}(N-1)a\right)\frac{\sin\left(Na\right)}{\sin a}, && (16.35)
\end{aligned}
$$

Finally, substituting for a gives

$$S[k] = \exp\left(-\mathrm{j}\pi(N-1)\left(\frac{k}{N} - f_0\Delta t\right)\right)\frac{\sin\left(N\pi\left(\frac{k}{N} - f_0\Delta t\right)\right)}{\sin\left(\pi\left(\frac{k}{N} - f_0\Delta t\right)\right)}. \tag{16.36}$$

10 DFT of a cosinusoid

Consider a cosinusoid of amplitude A and frequency f_0,

$$s(t) = A\cos(2\pi f_0 t)\,. \tag{16.37}$$

If this is sampled at an interval Δt, the equivalent discrete-time signal is

$$s[n] = A\cos(2\pi f_0 n\Delta t)\,. \tag{16.38}$$

The DFT is given by

$$\begin{aligned} S[k] &= \frac{1}{N}\sum_{n=0}^{N-1} s[n]\exp\left(-\mathrm{j}2\pi\frac{nk}{N}\right), && (16.39)\\ &= \frac{1}{N}\sum_{n=0}^{N-1} A\cos(2\pi f_0 n\Delta t)\exp\left(-\mathrm{j}2\pi\frac{nk}{N}\right). && (16.40)\end{aligned}$$

We want to convert this to a closed-form solution. The trick is to split the cosine into exponential functions using Euler's theorem,

$$2\cos\theta = \exp(-\mathrm{j}\theta) + \exp(\mathrm{j}\theta)\,, \tag{16.41}$$

and thus

$$s[n] = \frac{A}{2}\left(s_+[n] + s_-[n]\right), \tag{16.42}$$

where

$$\begin{aligned} s_+[n] &= \exp(-\mathrm{j}2\pi f_0 n\Delta t)\,, && (16.43)\\ s_-[n] &= \exp(\mathrm{j}2\pi f_0 n\Delta t)\,. && (16.44)\end{aligned}$$

Using the principle of linearity,

$$S[k] = \frac{A}{2}\left(S_+[k] + S_-[k]\right), \tag{16.45}$$

where

$$\begin{aligned} S_+[k] &= \sum_{n=0}^{N-1} s_+[n]\exp\left(-\mathrm{j}2\pi\frac{nk}{N}\right), && (16.46)\\ S_-[k] &= \sum_{n=0}^{N-1} s_-[n]\exp\left(-\mathrm{j}2\pi\frac{nk}{N}\right). && (16.47)\end{aligned}$$

$S_+[k]$ is given by (16.36) and from symmetry, $S_-[k]$ can be found to be

$$S_-[k] = \exp\left(-\mathrm{j}\pi(N-1)\left(\frac{k}{N} + f_0\Delta t\right)\right)\frac{\sin\left(N\pi\left(\frac{k}{N} + f_0\Delta t\right)\right)}{\sin\left(\pi\left(\frac{k}{N} + f_0\Delta t\right)\right)}. \tag{16.48}$$

Finally, combining (16.48) and (16.36) gives the DFT for a cosine to be

$$S[k] = \frac{A}{2N} \exp\left(-\mathrm{j}\pi(N-1)\left(\frac{k}{N} + f_0\Delta t\right)\right) \frac{\sin\left(N\pi\left(\frac{k}{N} + f_0\Delta t\right)\right)}{\sin\left(\pi\left(\frac{k}{N} + f_0\Delta t\right)\right)} \tag{16.49}$$

$$+ \frac{A}{2N} \exp\left(-\mathrm{j}\pi(N-1)\left(\frac{k}{N} - f_0\Delta t\right)\right) \frac{\sin\left(N\pi\left(\frac{k}{N} - f_0\Delta t\right)\right)}{\sin\left(\pi\left(\frac{k}{N} - f_0\Delta t\right)\right)}. \tag{16.50}$$

11 *DFT of an arbitrary sinusoid*

Let's consider a discrete-time sinusoid with arbitrary phase ϕ,

$$s[n] = A\cos\left(2\pi f_0 n\Delta t + \phi\right). \tag{16.51}$$

When $\phi = 0$ we have a cosinewave and when $\phi = \pi/2$ we have a sinewave.

$$S[k] = \frac{A}{2N} \exp\left(-\mathrm{j}\phi\right) \exp\left(-\mathrm{j}\pi(N-1)\left(\frac{k}{N} + f_0\Delta t\right)\right) \frac{\sin\left(N\pi\left(\frac{k}{N} + f_0\Delta t\right)\right)}{\sin\left(\pi\left(\frac{k}{N} + f_0\Delta t\right)\right)} \tag{16.52}$$

$$+ \frac{A}{2N} \exp\left(\mathrm{j}\phi\right) \exp\left(-\mathrm{j}\pi(N-1)\left(\frac{k}{N} - f_0\Delta t\right)\right) \frac{\sin\left(N\pi\left(\frac{k}{N} - f_0\Delta t\right)\right)}{\sin\left(\pi\left(\frac{k}{N} - f_0\Delta t\right)\right)}. \tag{16.53}$$

When $f_0\Delta t = k/(MN)$ then

$$S[k] = \frac{A}{2} \exp\left(\mathrm{j}\phi\right). \tag{16.54}$$

12 *DFT of an arbitrary sinusoid with zero padding*

Consider an arbitrary sinusoid of N samples. If we add $(M-1)N$ zeros to increase the signal length to MN samples then we increase the number of frequency samples by a factor of M so that $0 \le k < MN$.

The DFT of a zero-padded signal is

$$S[k] = \frac{1}{MN} \sum_{n=0}^{N-1} s[n] \exp\left(-\mathrm{j}2\pi \frac{nk}{MN}\right). \tag{16.55}$$

For the case of an arbitrary sinusoid, its DFT is

$$S[k] = \frac{A}{2N} \exp(-\mathrm{j}\phi) \exp\left(-\mathrm{j}\pi(N-1)\left(\frac{k}{MN} + f_0 \Delta t\right)\right) \frac{\sin\left(N\pi\left(\frac{k}{MN} + f_0\Delta t\right)\right)}{\sin\left(\pi\left(\frac{k}{MN} + f_0\Delta t\right)\right)} \tag{16.56}$$

$$+ \frac{A}{2N} \exp(\mathrm{j}\phi) \exp\left(-\mathrm{j}\pi(N-1)\left(\frac{k}{MN} - f_0 \Delta t\right)\right) \frac{\sin\left(N\pi\left(\frac{k}{MN} - f_0\Delta t\right)\right)}{\sin\left(\pi\left(\frac{k}{MN} - f_0\Delta t\right)\right)}. \tag{16.57}$$

Note, zero padding does not improve the frequency resolution but samples the spectrum more finely.

12.1 *Correlation with the FFT*

The FFT can also be used to efficiently compute a correlation:

$$x_1[n] \star x_2[n] \longleftrightarrow X_1[k] X_2^*[k]. \tag{16.58}$$

Again zero padding may be needed to avoid circular effects.

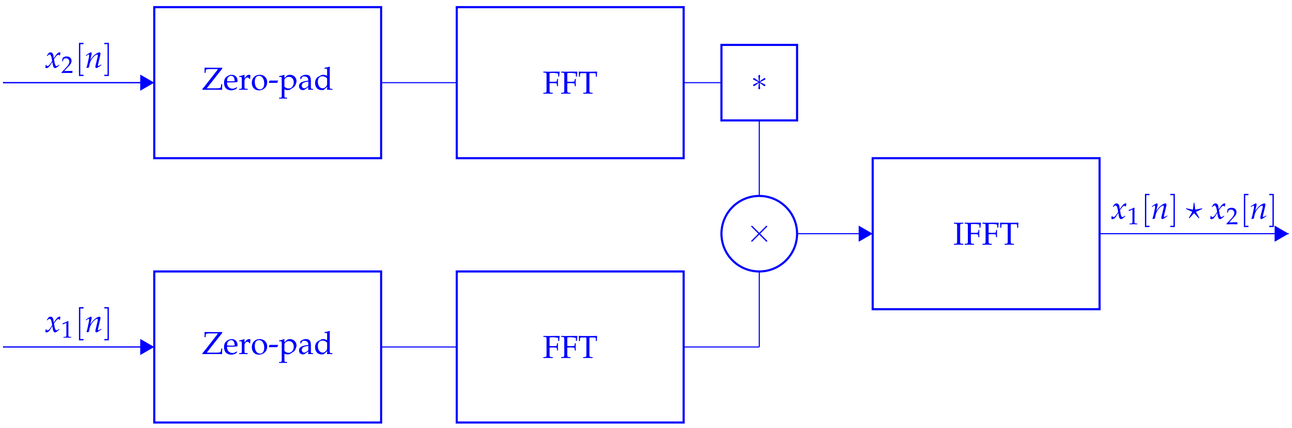


Figure 16.13: Efficient correlation using FFTs. The asterisk $*$ denotes a complex conjugate and a star $\star$ denotes correlation.

13 Summary

- The DFT transforms a discrete-time signal to a discrete-frequency spectrum.
- The FFT is an efficient algorithm for the DFT.
- The frequency resolution of a DFT is $1/T$ where T is the measurement period.
- The frequency spacing of a DFT is $1/(N\Delta t)$ where N is the DFT size.
- The spectrum can be interpolated by zero padding the signal before a DFT.
- An interpolated spectrum reveals the sidelobes due to time-domain truncation of the signal.
- The sidelobes in the spectrum can be reduced in amplitude (but increased) in width by applying a window function to the time-domain samples.
- Zeropadding to provide a guard band is required to avoid circular effects when using the FFT for efficient convolution.

14 Exercises

1. A 1024 point FFT is calculated for a signal sampled at 1 kHz. What is the spacing between frequency samples[11]?
2. A signal is sampled at 1 kHz for 1024 samples, zero-padded to a length 4096, and it spectrum calculated with a FFT. What is the spacing between frequency samples?
3. A discrete spectrum calculated with a FFT has 512 frequency samples spaced by 1 Hz. What is the spacing of the samples of the corresponding discrete-time signal?
4. What is the purpose of a guard band when performing convolution with FFTs?
5. A 100 sample sequence is to be convolved with a 150 sample sequence using FFTs. Determine the length of the output sequence and suggest a suitable FFT size.
6. Calculate the DFT of the sequence $\{\underline{1}, 2, 2, 1\}$.
7. What is the purpose of applying a window function before using an FFT?

[11] For some reason, samples in the frequency domain are often called bins.

17
Sensor fusion I

Sensor fusion is commonly used in mobile robotics to combine (fuse) measurements from different sensors. The fused measurement has a lower variance than the individual measurements. An example, is the fusing of measurements from an inertial measurement unit[1] (IMU) with a global positioning system (GPS). To understand how a better estimate is obtained, it is necessary to consider some statistics, in particular, estimation theory.

[1] These contain accelerometers, gyroscopes, and magnetometers.

1 A little problem

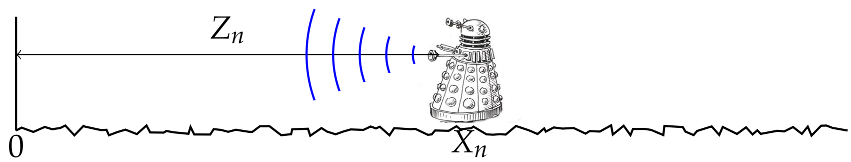


Figure 17.1: Robot position estimation problem.

Consider a robot that has been instructed to move at a speed of 3 m/s for 3 s. When it has stopped, an ultrasonic range sensor says that it moved 8 m, however, a laser range finder says that it moved 11 m. So how do we decide how far the robot moved?

1. Do we ignore the sensors and say that the robot moved 9 m since that is what we told it to do?
2. Do we choose the laser scanner measurement of 11 m since laser scanners are usually more accurate than ultrasonic range sensors?
3. Do we choose the ultrasonic measurement of 8 m since it is closest to what we expect?
4. Do we average the two measurements? If so, do we give each measurement equal weighting?

2 *Weighted averaging*

Let's say that the laser scanner measures distance with a standard deviation $\sigma_{Z_1} = 1\,\text{m}$ and the ultrasonic sensor measures distance with a standard deviation $\sigma_{Z_2} = 2\,\text{m}$? Does this extra information allow us to make a better estimate of the robot's position?

Clearly the laser scanner gives a more reliable measurement so if we use averaging we should give it a higher weighting. To determine what the weighting should be, we need a little statistics.

Let's assume that the distance measured by a laser range finder is described by a random variable Z_1 and the distance measured by the ultrasonic sensor is described by a random variable Z_2. Let's also assume that the errors are uncorrelated with variances $\text{Var}\left\{Z_1\right\}$ and $\text{Var}\left\{Z_2\right\}$.

A linear weighted average estimator for the distance is

$$\hat{Z} = w_1 Z_1 + w_2 Z_2. \tag{17.1}$$

To ensure the estimator is unbiased, the weights must sum to unity, and thus

$$\hat{Z} = w_1 Z_1 + (1 - w_1) Z_2. \tag{17.2}$$

If the sensor errors are uncorrelated, the variance of the estimator can be shown to be

$$\text{Var}\left\{\hat{Z}\right\} = w_1^2 \,\text{Var}\left\{Z_1\right\} + (1 - w_1)^2 \,\text{Var}\left\{Z_2\right\}, \tag{17.3}$$

and thus the best linear unbiased (BLU) estimator can be found to occur where the weights are chosen inversely proportional to the variances[2]. This yields

[2] Provided the measurements are uncorrelated.

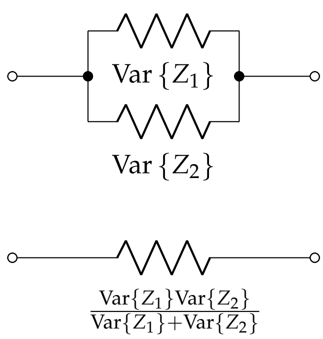
$$\hat{Z} = \frac{\frac{1}{\text{Var}\{Z_1\}} Z_1 + \frac{1}{\text{Var}\{Z_2\}} Z_2}{\frac{1}{\text{Var}\{Z_1\}} + \frac{1}{\text{Var}\{Z_2\}}}, \tag{17.4}$$

with a variance [3]

[3] This result is analogous to two resistors in parallel.

$$\text{Var}\left\{\hat{Z}\right\} = \frac{1}{\frac{1}{\text{Var}\{Z_1\}} + \frac{1}{\text{Var}\{Z_2\}}}, \tag{17.5}$$

$$= \frac{\text{Var}\left\{Z_1\right\} \text{Var}\left\{Z_2\right\}}{\text{Var}\left\{Z_1\right\} + \text{Var}\left\{Z_2\right\}}. \tag{17.6}$$



Note that this is smaller than $\text{Var}\left\{Z_1\right\}$ and $\text{Var}\left\{Z_2\right\}$.

2.1 Weighted average example

Let's return to the problem at the start and assume that the laser is described by Z_1 with $\mathrm{Var}\{Z_1\} = 1$ and the ultrasound sensor is described by Z_2 with $\mathrm{Var}\{Z_2\} = 4$. The BLU estimator using these sensors is

$$\hat{Z} = \frac{\frac{1}{\mathrm{Var}\{Z_1\}}Z_1 + \frac{1}{\mathrm{Var}\{Z_2\}}Z_2}{\frac{1}{\mathrm{Var}\{Z_1\}} + \frac{1}{\mathrm{Var}\{Z_2\}}}, \tag{17.7}$$

$$= \frac{4Z_1 + Z_2}{5}, \tag{17.8}$$

$$= 0.8Z_1 + 0.2Z_2, \tag{17.9}$$

with a variance

$$\mathrm{Var}\{\hat{Z}\} = \frac{1}{\frac{1}{\mathrm{Var}\{Z_1\}} + \frac{1}{\mathrm{Var}\{Z_2\}}}, \tag{17.10}$$

$$= \frac{1}{\frac{1}{1} + \frac{1}{4}}, \tag{17.11}$$

$$= 0.8, \tag{17.12}$$

and a standard deviation

$$\sigma_{\hat{Z}} = \sqrt{\mathrm{Var}\{\hat{Z}\}}, \tag{17.13}$$

$$= \sqrt{0.8}, \tag{17.14}$$

$$= 0.89. \tag{17.15}$$

So with a measured value $z_1 = 11$ for Z_1 and $z_2 = 8$ for Z_2, then the BLU estimate is

$$\hat{z} = \frac{\frac{1}{\mathrm{Var}\{Z_1\}}z_1 + \frac{1}{\mathrm{Var}\{Z_2\}}z_2}{\frac{1}{\mathrm{Var}\{Z_1\}} + \frac{1}{\mathrm{Var}\{Z_2\}}}, \tag{17.16}$$

$$= 0.8 \times 11 + 0.2 \times 8, \tag{17.17}$$

$$= 10.4\,\mathrm{m}. \tag{17.18}$$

Note, the estimate may sometimes result in a poorer value than one of the measured values but on average it will be better.

3 *Bias and noise*

Consider an ideal speed sensor with a time-varying output signal denoted by $s(t)$. The real measured signal is related to the ideal sensor signal by

$$x(t) = s(t) + b(t) + n(t). \tag{17.19}$$

Here

$b(t)$ denotes a slowly varying additive bias (offset),

$n(t)$ is a rapidly varying additive noise signal.

The bias can come from a mis-calibration or from an amplifier offset. The noise is generated by any sensor or amplifier resistance. An example of a sinusoidal signal with a bias and noise is shown in Figure 17.2.

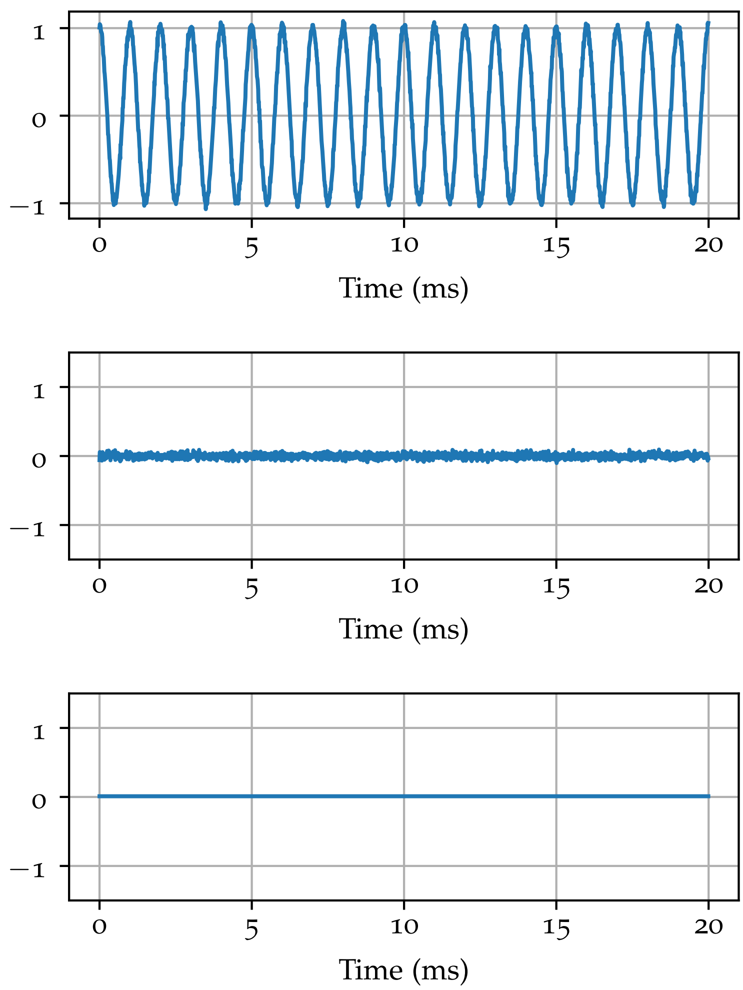


Figure 17.2: Signal plus bias plus noise; noise; bias.

For a control system we may need the derivative of the measured signal. Assuming that $b(t)$ is slowly varying, then its derivative can be neglected and thus the

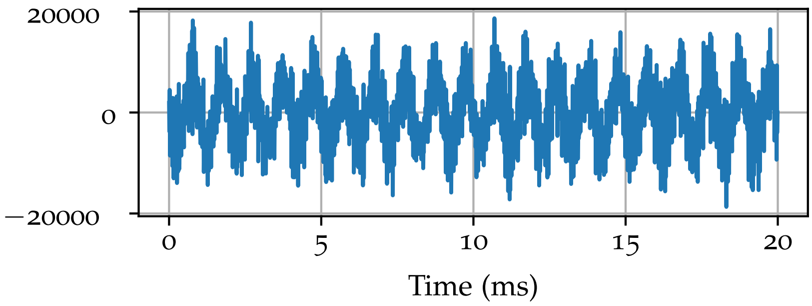


Figure 17.3: Derivative (first-order difference) of signal plus bias plus noise. Notice how the noise is more apparent since a differentiator is a high-pass filter.

derivative of the measured signal can be approximated by

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} \approx \left(\frac{\mathrm{d}s(t)}{\mathrm{d}t} + \frac{\mathrm{d}n(t)}{\mathrm{d}t}\right). \tag{17.20}$$

Note, we cannot neglect the derivative of the noise since it is wideband and thus rapidly varying. Indeed, the faster it changes, the greater its derivative. Thus the high-frequency noise components are amplified[4] more than the low-frequency signal as can be seen by Figure 17.3.

[4] The frequency response of a differentiator is $H(f) = \mathrm{j}2\pi f$.

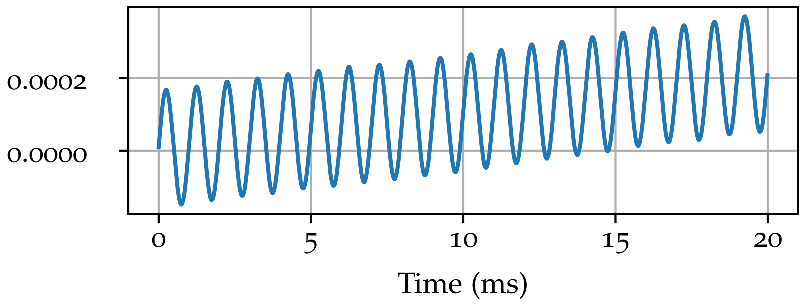


Figure 17.4: Integral (cumulative sum) of signal plus bias plus noise. Notice how the small bias integrates to a large drift.

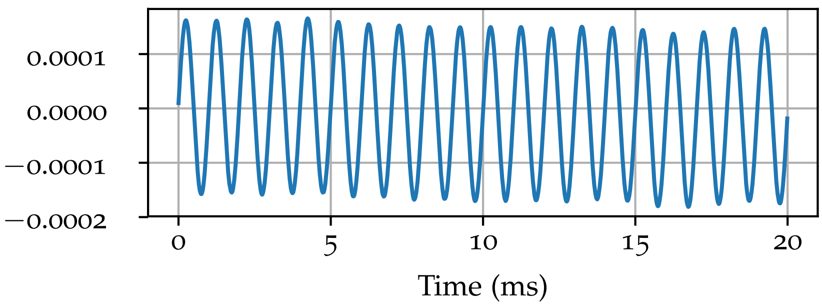


Figure 17.5: Integral (cumulative sum) of signal plus noise. There is no obvious high frequency noise, however, even without a bias, the low frequency noise produces a random walk (shown by the wandering peak amplitude).

If we integrate the measured signal, the high-frequency noise is rapidly attenuated[5]. However, low frequency

[5] The frequency response of an integrator is $H(f) = 1/\,(\mathrm{j}2\pi f)$.

noise produces a random walk. Worse still, a bias will produce a drift, see Figure 17.4. For example, let's assume that $b(t)$ is a constant and thus

$$\int_0^t x(t)\mathrm{d}t \approx \int_0^t s(t)\mathrm{d}t + \int_0^t b(t)\mathrm{d}t, \tag{17.21}$$

$$\approx \int_0^t s(t)\mathrm{d}t + bt. \tag{17.22}$$

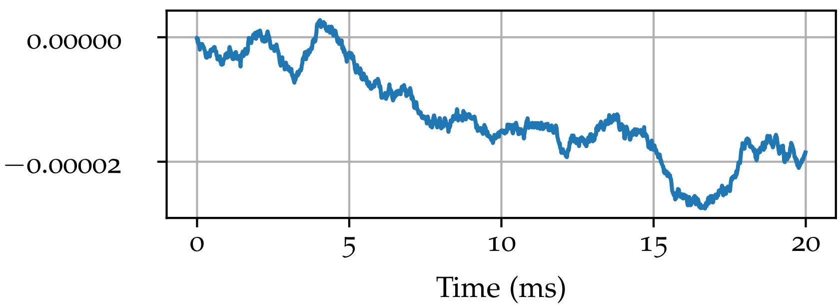


Figure 17.6: Cumulative sum of noise showing a random-walk.

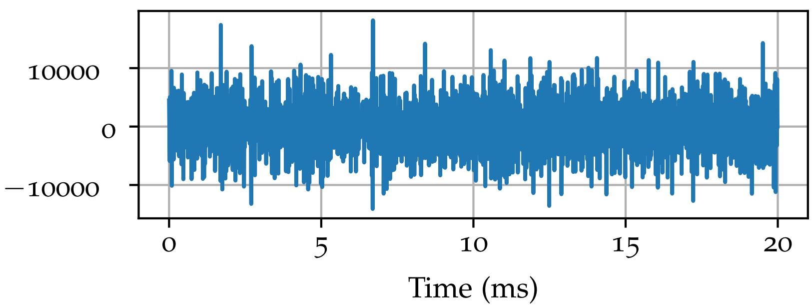


Figure 17.7: Noise difference.

Instead of differentiating a speed sensor to get a noisy acceleration and integrating it to get a drifting position, what about using separate speed, acceleration, and displacement sensors? But since the noise is independent for each sensor can we combine the three sensors together to reduce the noise for each estimate of speed, acceleration, and displacement? Or, what about just combining displacement and acceleration sensors? The answer is yes and the process is called sensor fusion.

4 Exercises

1. An ultrasonic sensor with a variance of $0.01\,\text{m}^2$ measures the range to a wall to be 1.2 m. A lidar with a variance $0.01\,\text{m}^2$ measures the range to the wall to be 1.1 m.
   (a) If the measurements are to be fused using a linear weighted sum, what should the weightings be?
   (b) If the measurements are to be fused using a linear weighted sum, what is the resulting variance?
2. An ultrasonic sensor with a variance of $0.04\,\text{m}^2$ measures the range to a wall to be 1.2 m. A lidar with a variance $0.01\,\text{m}^2$ measures the range to the wall to be 1.1 m.
   (a) If the measurements are to be fused using a linear weighted sum, what should the weightings be?
   (b) If the measurements are to be fused using a linear weighted sum, what is the resulting variance?
3. If measurements from range sensor have a standard deviation of 0.001 m what is the standard deviation of the result if 100 values are averaged?
4. Consider two range sensors: one that samples at 1 Hz with a variance of $0.01\,\text{m}^2$ and another that samples at 10 Hz with a variance of $0.05\,\text{m}^2$. Which would you choose and why?

18
Estimation introduction

Measurements from sensors are corrupted by noise. However, if we have a model of the sensor and the noise we can apply estimation techniques to get a better answer.

Estimation techniques (estimators) include averaging and filtering. The answer they give is called an estimate. Ideally an estimator should not have a bias and its variance should be as small as possible.

The best possible estimator is called a minimum variance unbiased estimator (MVUE). This has a variance given by the Cramér-Rao lower bound (CRLB)[1].

Bayesian estimators use prior information that biases the result. Thus they can achieve a variance lower than the CRLB.

[1] No unbiased estimators can achieve a lower variance than the CRLB.

1 Observation model

Estimators require an observation model. A typical observation model with additive noise is

$$Z_n = f(\boldsymbol{\beta}) + V_n \quad \text{for } n = 0, 1, \cdots N-1, \tag{18.1}$$

where Z_n is the n-th random variable of a random process describing the measurements, V_n is the n-th random variable of a random process describing additive noise, and $f(\boldsymbol{\beta})$ is a function of some parameters, $\boldsymbol{\beta}$, that we wish to estimate[2]. For example, consider a DC signal corrupted by additive noise. The observation model is

$$Z_n = a + V_n, \tag{18.2}$$

where a is a constant but unknown parameter that we wish to estimate. This observation model is linear in terms of the parameter, a.

A second example is determining the parameters of an AC signal corrupted by additive noise,

$$Z_n = a\cos(2\pi f_0 T + \phi) + V_n. \tag{18.3}$$

[2] The noise process also has parameters (mean, variance, etc) that are assumed to be known.

In this example, there are three unknown parameters: the amplitude a, the frequency f_0, and the phase ϕ. The observation model is linear in the amplitude parameter a but is non-linear in the frequency and phase parameters.

A third example is estimating the gravitational acceleration and the initial speed of a ball thrown into the air from height measurements. The height measurements can be modelled as

$$Z_n = v_i t_n + a t_n^2 + V_n, \tag{18.4}$$

where $t_n = n\Delta t$. Here the measurements are non-linear in t_n but linear in the parameters v_i and a.

A fourth example is the estimation of parameters of an unknown noise process,

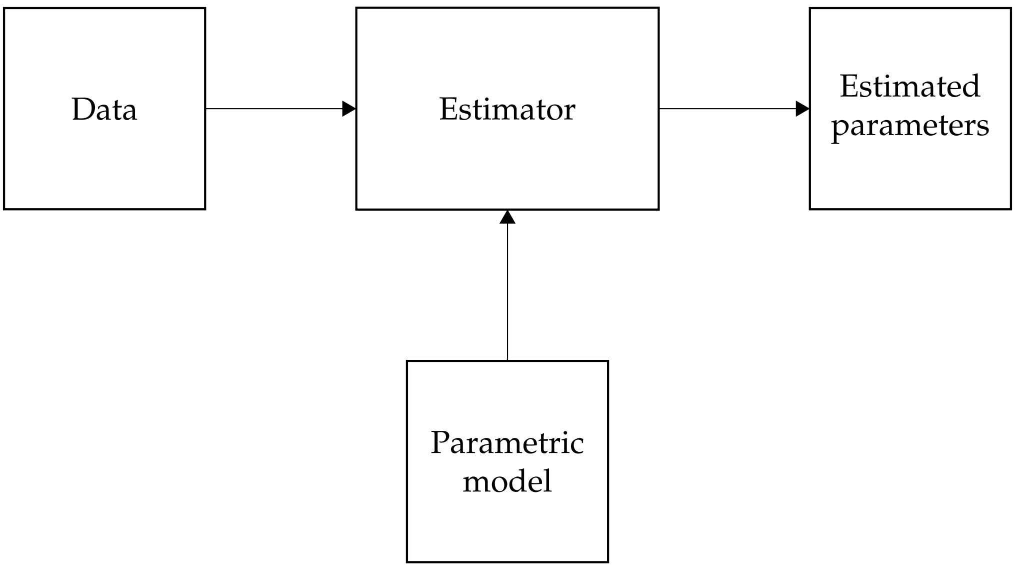
$$Z_n = V_n(\mu, \sigma). \tag{18.5}$$

Here μ is the unknown mean of the noise process and σ is the unknown variance.

2 *Estimators*

An estimator is a function of the observed random process and is itself a random variable. It is usually denoted with a hat, $\hat{X}$. A particular realization of this random variable is called an *estimate*, $\hat{x}$. This is the value determined from some specific measurements[3].

[3] A realisation of the observed random process.



2.1 Minimum variance unbiased estimator (MVUE)

A desirable estimator has no bias and the smallest possible variance. This is called a minimum variance unbiased estimator (MVUE). The smallest possible achievable variance is given by the Cramér-Rao Lower Bound (CRLB) or (CRB)[4]. The CRLB depends on the signal to noise ratio (SNR); the higher the SNR, the lower the variance of the estimator.

The minimum variance unbiased estimator (MVUE) may not exist or may be hard to determine for certain estimation problems. In practice, a maximum likelihood estimator (MLE) is used since it asymptotically reaches the CRLB with a large number of measurements.

[4] Much effort in estimators has been expended by the military. A better estimator allows an invader to be more reliably detected at greater distances, say using sonar or radar.

2.2 Best linear unbiased estimator (BLUE)

The best linear unbiased estimator (BLUE) is a weighted average where the weights are chosen to avoid a bias and to minimise the variance of the result. It is a MVUE for linear problems.

2.3 Estimator notation

$\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ commonly denote a vector of parameters[5] of a statistical model. The estimator[6] of β is denoted $\hat{\mathbf{B}}$. This is short-hand for $\hat{\mathbf{B}}(\mathbf{Z})$ where $\mathbf{Z}$ is a sequence of random variables ($Z_{0:N-1}$) corresponding to the observed data. Thus $\hat{\mathbf{B}}(\mathbf{Z})$ is a random variable that is a function of the observed data. $\hat{\beta}(\mathbf{z})$ denotes the estimate for a particular sequence of observed data, $\mathbf{Z} = \mathbf{z}$.

[5] Mean and variance for a Gaussian, rate parameter for Poisson distribution, etc.

[6] An estimator is a statistic since it is a function of the data.

3 Maximum likelihood estimator (MLE)

When it is difficult to find a MVUE, the maximum likelihood estimator (MLE) is popular. Unfortunately, this has a bias[7], but the bias gets smaller for larger numbers of samples, N. In the limit as $N \longrightarrow \infty$, there is no bias and MLE is equivalent to MVUE.

MLE requires the joint PDF for the random process to be known. Using this with the observation model, a likelihood function is formed as a function of the measurements and unknown parameters. A search is then performed to find the parameters that maximises the likelihood function given the measurements. Mathematically the maximum likelihood estimate can be formu-

[7] There are ML estimators that estimate and subtract the bias to first-order.

lated as an optimisation:

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} l_{Z_{0:N-1}}(\boldsymbol{\beta}; z_{0:N-1}), \tag{18.6}$$

where $l_{Z_{0:N-1}}(\boldsymbol{\beta}; z_{0:N-1})$ is the likelihood function for the set of N measurements, $z_{0:N-1} = z_0, z_1, \cdots z_{N-1}$, and parameter vector $\boldsymbol{\beta}$.

The likelihood function is greatly simplified if each random variable in the random process is independent and has the same distribution[8]. In this case, the joint PDF is the product of the PDFs of the random variables in the random process. For example, this applies for additive white Gaussian noise (AWGN).

[8] Independent and identically distributed (IID).

When the noise process is Gaussian distributed, MLE is equivalent to generalised least squares (GLS). If the noise process is also IID, MLE is equivalent to ordinary least squares (OLS). Finally, if the observation model is linear in the parameters, a search over the parameter space is not required and a direct solution is possible. In this case, the observation model can be written in vector form[9] as

$$\mathbf{Z} = \mathbf{A}\boldsymbol{\beta} + \mathbf{N}. \tag{18.7}$$

[9] The columns of $\mathbf{A}$ are the basis vectors.

The least squares estimator[10] is

$$\hat{\mathbf{B}} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1} \mathbf{A}\mathbf{Z}, \tag{18.8}$$

[10] The matrix $\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}$ is called the Moore-Penrose pseudoinverse of $\mathbf{A}$. For complex data, the matrix transpose is replaced with the Hermitian transpose, $\mathbf{A}^{\mathrm{H}}$.

and for a specific vector of measurements, $\mathbf{z}$, the least squares estimate is

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1} \mathbf{A}\mathbf{z}. \tag{18.9}$$

Note, that the estimated parameters are given by a linear weighted sum of the measurements.

4 *Bayesian estimation*

ML estimators are completely data driven. However, there are times when we know something about the possible parameters. This additional prior information can be utilised by Bayesian estimators.

Bayesian estimators treat the unknown parameters in an observation model as random variables. Thus each parameter has a PDF. The PDF before a measurement is called the *prior PDF*; this represents our prior knowledge of the possible solutions. For example, a robot may be physically constrained and so its position can only have a limited range of values. If the constrained robot positions are equally likely, the prior PDF has a uniform distribution.

After measurements have been made, a Bayesian estimator combines the likelihood function with the prior PDF to achieve the *posterior PDF*. Finally, the 'best' value of the parameters is chosen as the estimate. This is found either by looking for the parameters that maximise the posterior PDF, the maximum a posteriori (MAP) estimate, or using the mean value of the posterior PDF, the minimum mean square estimate (MMSE)[11].

[11] If the posterior PDF is Gaussian, the MAP and MMSE estimates are equivalent.

5 *Mean estimation*

Consider a sequence of N independent and identically distributed (IID) random variables from a random process,

$$\mathbf{Z} = \{Z_0, Z_1, Z_2, \cdots, Z_{N-1}\}. \tag{18.10}$$

Since Z_n are IID, they have the same mean μ and variance σ^2. Let's say we wish to estimate the mean. This can be achieved with a well-known estimator[12], the time average[13]

[12] This is a point estimator.

[13] Sample average.

$$\hat{\mu}(\mathbf{Z}) = \frac{1}{N} \sum_{n=0}^{N-1} Z_n. \tag{18.11}$$

The estimator for the mean produces a random variable $\hat{\mu}(\mathbf{Z})$. This is a random variable since if someone else took N samples, they would get a different value, $\hat{\mu}(\mathbf{z})$, where

$$\hat{\mu}(\mathbf{z}) = \frac{1}{N} \sum_{n=0}^{N-1} z_n. \tag{18.12}$$

Here $\mathbf{z}$ denotes the sequence of observed values for a realisation of $\mathbf{Z}$,

$$\mathbf{z} = \{z_0, z_1, z_2, \cdots, z_{N-1}\}. \tag{18.13}$$

Equation (18.11) is an unbiased estimator since

$$\mathrm{E}\left\{\overline{\mathbf{Z}}\right\} = \mu, \tag{18.14}$$

The variance of the estimator is

$$\mathrm{Var}\left\{\overline{\mathbf{Z}}\right\} = \frac{\sigma^2}{N}. \tag{18.15}$$

6 Maximum likelihood amplitude estimation

Consider an observation model:

$$z[n] = As[n] + v[n], \tag{18.16}$$

where $s[n]$ is a known signal, $v[n]$ is an additive zero-mean white Gaussian random process, and A is a constant but unknown amplitude.

The likelihood function for N observations is

$$l\left(A; z_{0:N-1}\right) = \prod_{n=0}^{N-1} f_V(z[n] - As[n]), \tag{18.17}$$

and the maximum likelihood estimate can be found from a search,

$$\hat{A} = \arg\max_A l\left(A; z_{0:N-1}\right). \tag{18.18}$$

Since the noise is independent zero-mean Gaussian,

$$l\left(A; z_{0:N-1}\right) = \prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left(\frac{z[n] - As[n]}{\sigma}\right)^2\right), \tag{18.19}$$

and using the log-likelihood, the product is converted to a summation

$$\log l\left(A; z_{0:N-1}\right) = -\log\sqrt{2\pi\sigma^2} - \sum_{n=0}^{N-1}\left(\frac{1}{2}\left(\frac{z[n] - As[n]}{\sigma}\right)^2\right). \tag{18.20}$$

Since the log function is monotonic, the maximum likelihood estimate is equivalent to

$$\hat{A} = \arg\min_A \sum_{n=0}^{N-1}\left(\frac{1}{2}\left(\frac{z[n] - As[n]}{\sigma}\right)^2\right). \tag{18.21}$$

Since the noise is identically distributed, the ML estimate is equivalent to the least-squares estimate,

$$\hat{A} = \arg\min_A \sum_{n=0}^{N-1} \left(z[n] - As[n]\right)^2. \tag{18.22}$$

Expanding the square and dropping terms independent of A,

$$\hat{A} = \arg\min_A \sum_{n=0}^{N-1} -2As[n]z[n] + A^2 s^2[n]. \tag{18.23}$$

Differentiating the objective function with respect to A and equating to zero, the estimate of A can be directly obtained as

$$\hat{A} = \frac{\sum_{n=0}^{N-1} z[n]s[n]}{\sum_{n=0}^{N-1} s^2[n]}. \tag{18.24}$$

The ML estimator is known to have a bias since it can be shown that

$$\mu_{\hat{A}} = \mathrm{E}\left\{\hat{A}\right\} \approx A + \frac{\sigma^2}{NA}. \tag{18.25}$$

19
Sensor fusion II

1 Heading estimation

A common robotics problem is the estimation of the direction a robot is facing (the heading angle). This can be determined using a compass (magnetometer) but this is not accurate in the presence of ferrous materials. In principle, it can also be determined from the angular velocity measured by a gyroscope. However, since a gyroscope has an unknown bias, the angular speeds cannot be integrated to find the absolute angle. Thus it it necessary to estimate the bias using the magnetometer and then subtract this from the gyroscope measurements before they can be integrated.

Of the many ways to estimate the heading, the complementary filter is the easiest to implement. The Kalman filter can give a better solution but is considerably more difficult to implement.

1.1 Complementary filter

A complementary filter estimates the heading by fusing magnetometer and gyroscope measurements using

$$\hat{\theta}_{n+1} = \alpha \left(\hat{\theta}_n + \omega_n \Delta t \right) + (1 - \alpha)\theta_n, \tag{19.1}$$

where ω_n is the angular speed measured by the gyroscope, θ_n is the heading measured by the magnetometer, Δt is the sampling interval, and α is a weighting parameter.

The principle behind the complementary filter is that the gyroscope has less noise than the magnetometer but since it has an unknown bias it cannot be directly integrated. So instead it is integrated and high-pass filtered. The missing part of the spectrum is filled in by adding low-pass filtered magnetometer measurements. This is shown in Figure 19.1.

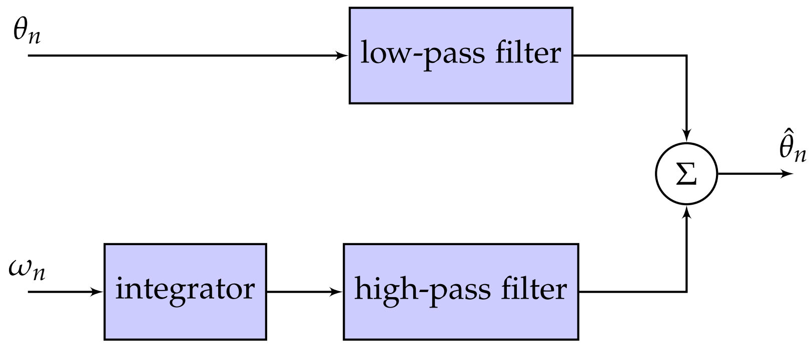


Figure 19.1: Complementary filter for fusing magnetometer and gyroscope measurements.

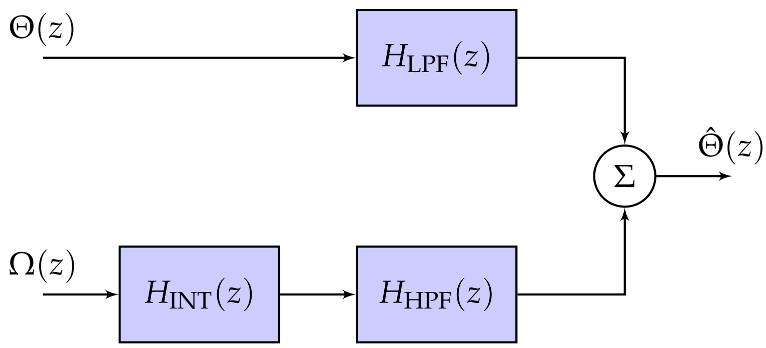


Figure 19.2: Complementary filter in the z-domain. $H_{\text{HPF}}(z)$ is the transfer function for a high-pass filter, $H_{\text{LPF}} = 1 - H_{\text{HPF}}(z)$ is the transfer function of a complementary low-pass filter.

In the z-domain (see Figure 19.2), these operations can be represented using

$$\hat{\Theta}(z) = H_{\text{LPF}}(z)\Theta(z) + H_{\text{INT}}(z)H_{\text{HPF}}(z)\Omega(z), \tag{19.2}$$

where $H_{\text{LPF}}(z)$ is the transfer function of the low-pass filter, and $H_{\text{HPF}}(z)$ is the transfer function of the high-pass filter, and $H_{\text{INT}}(z)$ is the transfer function of the integrator.

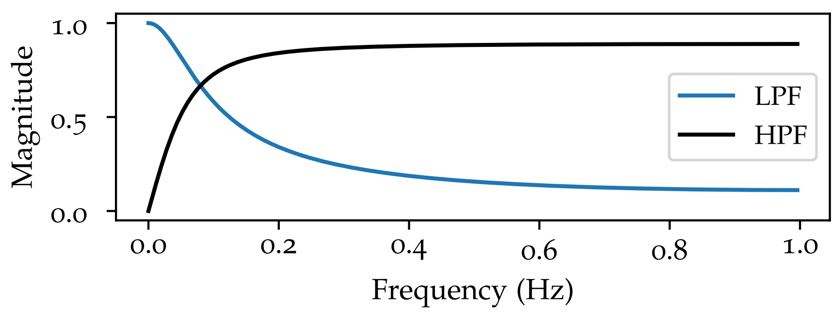


Figure 19.3: First order complementary high-pass and low-pass filter responses.

Let's choose a first-order integrator so

$$H_{\text{INT}}(z) = \frac{\Delta t}{1 - z^{-1}}. \tag{19.3}$$

Let's also choose a first-order recursive low-pass filter for $H_{\text{LPF}}(z)$,

$$H_{\text{LPF}}(z) = \frac{1-\alpha}{1-\alpha z^{-1}}. \tag{19.4}$$

The complementary high-pass filter is

$$H_{\text{HPF}}(z) = 1 - H_{\text{LPF}}(z) = \frac{\alpha(1-z^{-1})}{1-\alpha z^{-1}}. \tag{19.5}$$

Using these filters, (19.2) becomes

$$\hat{\Theta}(z) = \frac{1-\alpha}{1-\alpha z^{-1}}\Theta(z) + \frac{\Delta t}{1-z^{-1}}\frac{\alpha(1-z^{-1})}{1-\alpha z^{-1}}\Omega(z), \tag{19.6}$$

$$= \frac{1-\alpha}{1-\alpha z^{-1}}\Theta(z) + \frac{\alpha\Delta t}{1-\alpha z^{-1}}\Omega(z). \tag{19.7}$$

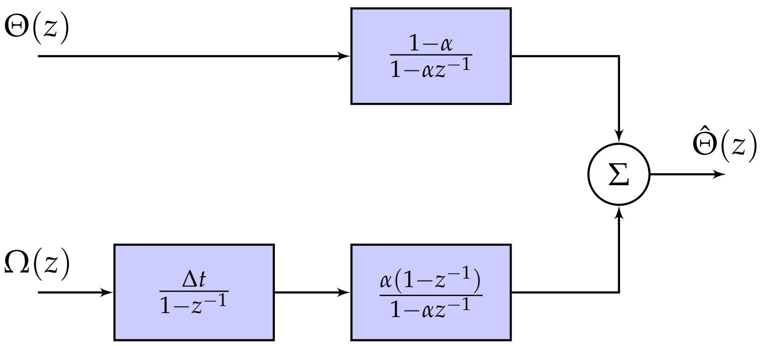


Figure 19.4: Complementary first-order filter in the z-domain.

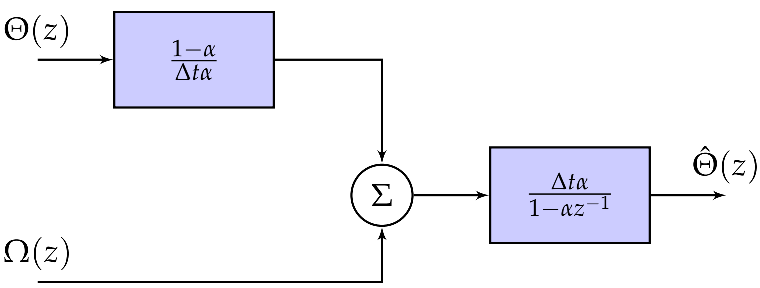


Figure 19.5: Simplified complementary first-order filter in the z-domain.

Equation (19.7) can be expressed as (see Figure 19.5)

$$\hat{\Theta}(z)\left(1-\alpha z^{-1}\right) = (1-\alpha)\Theta(z) + \alpha\Delta t\,\Omega(z), \tag{19.8}$$

and taking inverse z-transforms yields

$$\hat{\theta}_n = \alpha\left(\hat{\theta}_{n-1} + \omega_n\Delta t\right) + (1-\alpha)\theta_n. \tag{19.9}$$

This equation describes one implementation of a complementary filter. It is easy to implement on a microcontroller since it only requires three multiplications per time-step. The parameter α is chosen to achieve the desired filter time-constant, τ, where

$$\tau = \frac{\alpha}{1-\alpha}\Delta t. \tag{19.10}$$

Table 19.1: Relationship between complementary filter parameter α and its time constant τ.

α	$\tau/\Delta t$
0.9	9
0.95	19
0.98	49
0.99	99
0.999	999

Some examples of the performance of a complementary filter with different time-constants are shown in Figure 19.6, Figure 19.7, and Figure 19.8.

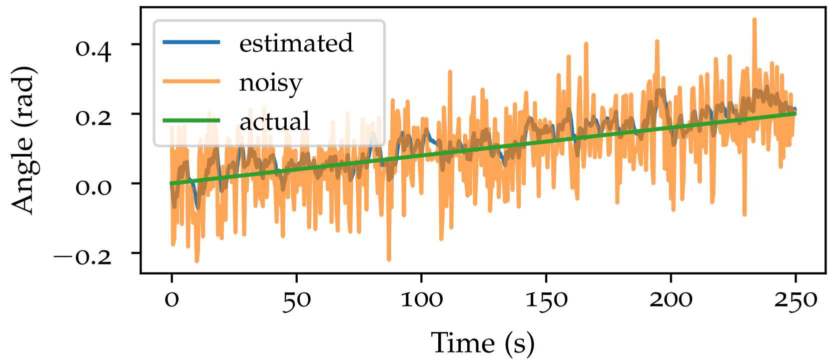


Figure 19.6: Heading estimation with a complementary filter; $\alpha = 0.8$.

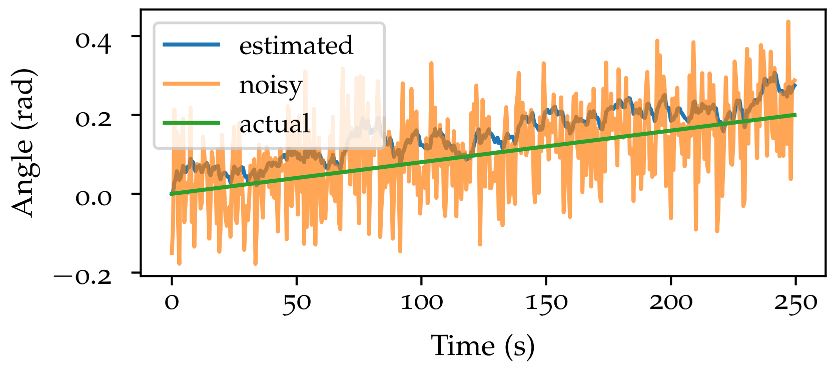


Figure 19.7: Heading estimation with a complementary filter; $\alpha = 0.9$.

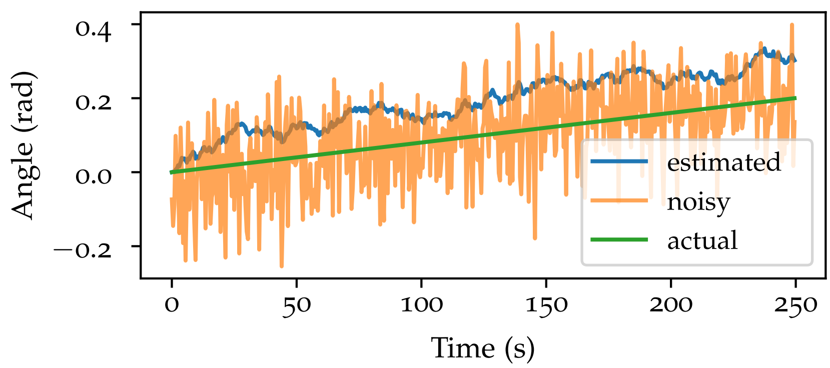


Figure 19.8: Heading estimation with a complementary filter; $\alpha = 0.95$. The time-constant is too long to remove the low-frequency gyroscope bias.

The complementary filter is equivalent to a steady-state Kalman filter for a certain class of problems.

1.2 Heading estimation with a Kalman filter

A Kalman filter attempts to estimate the state given the system model, measurement model, measure noise statistics, and process noise statistics. It assumes a linear system model, a linear measurement model, and that the process noise and measurement noise are both Gaussian distributed. With this assumption, the estimated state also has a Gaussian distribution.

The measurements at time-step n can be represented by a vector

$$\mathbf{z}_n = \begin{bmatrix} z_{\theta_n} \\ z_{\omega_n} \end{bmatrix}. \tag{19.11}$$

Here z_{θ_n} is the angle measured by the magnetometer[1] and z_{ω_n} is the angular speed measured by the gyroscope. Both these measurements are corrupted by noise and can be considered samples of a random measurement vector

[1] For example, using $z_{\theta_n} = \text{atan2}(m_{x_n}, m_{y_n})$ where m_{x_n} and m_{y_n} are the x and y magnetometer components.

$$\mathbf{Z}_n = \begin{bmatrix} Z_{\theta_n} \\ Z_{\omega_n} \end{bmatrix}. \tag{19.12}$$

For each time-step n, Z_{θ_n} is a random variable modelling the angle estimated by the magnetometer and Z_{ω_n} is a random variable modelling the angular speed measured by the gyroscope. Assuming that the measurements are corrupted by additive noise, the measurement random variables can be expressed as

$$Z_{\theta_n} = \theta_n + V_{\theta_n}, \tag{19.13}$$
$$Z_{\omega_n} = \omega_n + b_n + V_{\omega_n}. \tag{19.14}$$

Here θ_n is the true heading, ω_n is the true angular speed, b_n is the true gyroscope bias, V_{θ_n} is a random variable that models the measurement noise for the magnetometer, and V_{ω_n} is a random variable that models the measurement noise for the gyroscope.

A good choice of state vector for estimating the heading is

$$\mathbf{x}_n = \begin{bmatrix} \theta_n \\ \omega_n \\ b_n \end{bmatrix}. \tag{19.15}$$

Here θ_n is the heading angle, $\omega_n = \dot{\theta}_n$ is the angular speed, and b_n is the gyroscope bias.

The state transition equations (system model) are:

$$\theta_n = \theta_{n-1} + \omega_{n-1}\Delta t, \tag{19.16}$$
$$\omega_n = \omega_{n-1}, \tag{19.17}$$
$$b_n = b_{n-1}. \tag{19.18}$$

This model assumes a constant angular speed (or at least a slowly changing angular speed) and that the gyroscope bias is constant (or slowly changing). In practice, the model is not perfect and it is better to employ a stochastic model,

$$\Theta_n = \Theta_{n-1} + \Omega_{n-1}\Delta t + W_{\theta_n}, \tag{19.19}$$

$$\Omega_n = \Omega_{n-1} + W_{\omega_n}, \tag{19.20}$$

$$B_n = B_{n-1} + W_{b_n}. \tag{19.21}$$

Here the random variable W_{θ_n} represent the uncertainty in the model (process noise) for the heading, W_{ω_n} represent the uncertainty in the model for the angular speed, and W_{b_n} represent the uncertainty in the model for the gyroscope bias.

Equation (19.21) can be written in matrix form as

$$\mathbf{X}_n = \mathbf{A}\mathbf{X}_{n-1} + \mathbf{W}_n, \tag{19.22}$$

where the state random vector is

$$\mathbf{x}_n = \begin{bmatrix} \Theta_n \\ \Omega_n \\ B_n \end{bmatrix}, \tag{19.23}$$

the process noise random vector is

$$\mathbf{W}_n = \begin{bmatrix} W_{\theta_n} \\ W_{\omega_n} \\ W_{b_n} \end{bmatrix}, \tag{19.24}$$

and where the state transition matrix is

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{19.25}$$

Here Δt is the sampling period.

The measurement random vector can be related to the state random vector by

$$Z_{\theta_n} = \Theta_n + V_{\theta_n}, \tag{19.26}$$

$$Z_{\omega_n} = \Omega_n + B_n + V_{\omega_n}, \tag{19.27}$$

or in matrix form,

$$\mathbf{Z}_n = \mathbf{C}\mathbf{X}_n + \mathbf{V}_n, \tag{19.28}$$

where

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \tag{19.29}$$

and

$$\mathbf{V}_n = \begin{bmatrix} V_{\theta_n}, \\ V_{\omega_n}. \end{bmatrix} \tag{19.30}$$

Denoting the estimated state vector as a random vector, $\hat{\mathbf{X}}_n$, the Gaussian assumption means that $\hat{\mathbf{X}}_n$ can be represented by a mean vector and a covariance matrix. The covariance matrix handles the correlations between the different state variables. The probability density function of the estimated state vector is

$$f_{\hat{\mathbf{X}}_n}(\hat{\mathbf{x}}_n) = \frac{1}{\sqrt{(2\pi)^N \det\left(\boldsymbol{\Sigma}_{\hat{\mathbf{X}}_n}\right)}} \exp\left(-\frac{1}{2}\left(\hat{\mathbf{x}}_n - \mu_{\hat{\mathbf{X}}_n}\right)^{\mathrm{T}} \boldsymbol{\Sigma}_{\hat{\mathbf{X}}_n}{}^{-1}\left(\hat{\mathbf{x}}_n - \mu_{\hat{\mathbf{X}}_n}\right)\right). \tag{19.31}$$

Here $\mu_{\hat{\mathbf{X}}_n}$ is the mean of the estimated state vector and $\boldsymbol{\Sigma}_{\hat{\mathbf{X}}_n}$ is the covariance matrix of the estimated state vector. All these subscripts make the notation confusing, so let's define

$$\hat{\mathbf{P}}_n = \boldsymbol{\Sigma}_{\hat{\mathbf{X}}_n}. \tag{19.32}$$

At each time-step the Kalman filter performs two steps, a prediction step followed by an update step. The predicted (prior) state estimate is

$$\hat{\mathbf{x}}_n^- = \mathbf{A}\hat{\mathbf{x}}_{n-1}^+ + \mathbf{B}\mathbf{u}_n, \tag{19.33}$$

and the predicted (prior) state covariance[2] estimate is

$$\hat{\mathbf{P}}_n^- = \mathbf{A}\hat{\mathbf{P}}_{n-1}^-\mathbf{A}^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathbf{W}}, \tag{19.34}$$

[2] This is a generalisation of variance for multiple random variables.

where, $\boldsymbol{\Sigma}_{\mathbf{W}} = \mathrm{E}\left\{\mathbf{W}\mathbf{W}^{\mathrm{T}}\right\}$ is the process noise covariance matrix[3]. Note, for this problem there is no control vector, $\mathbf{u}_n$, and so $\mathbf{B} = 0$. However, if there was information available saying how the much the robot was expected to turn, then this can be included to improve the estimate of the robot's state.

[3] In general, this can vary with time.

The updated (posterior) mean state estimate is[4]

$$\hat{\mathbf{x}}_n^+ = \hat{\mathbf{x}}_n^- + \mathbf{K}_n\left(\mathbf{z}_n - \mathbf{C}\hat{\mathbf{x}}_n^-\right), \tag{19.35}$$

[4] The first term is the predicted state estimate. The second term is called the correction since it corrects the predicted state using the measurement. It is the product of the Kalman gain and the innovation, the difference between the actual and predicted measurements.

and the updated (posterior) state covariance estimate is

$$\hat{\mathbf{P}}_n^+ = \left(\mathbf{I} - \mathbf{K}_n\mathbf{C}\right)\hat{\mathbf{P}}_n^-, \tag{19.36}$$

where $\mathbf{K}_n$ is the optimal Kalman gain calculated by

$$\mathbf{K}_n = \hat{\mathbf{P}}_n^-\mathbf{C}^{\mathrm{T}}\left(\mathbf{C}\hat{\mathbf{P}}_n^-\mathbf{C}^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathbf{V}}\right)^{-1}. \tag{19.37}$$

Here, $\boldsymbol{\Sigma}_{\mathbf{V}} = \mathrm{E}\left\{\mathbf{V}\mathbf{V}^{\mathrm{T}}\right\}$ is the measurement noise covariance matrix[5].

[5] In general, this can vary with time.

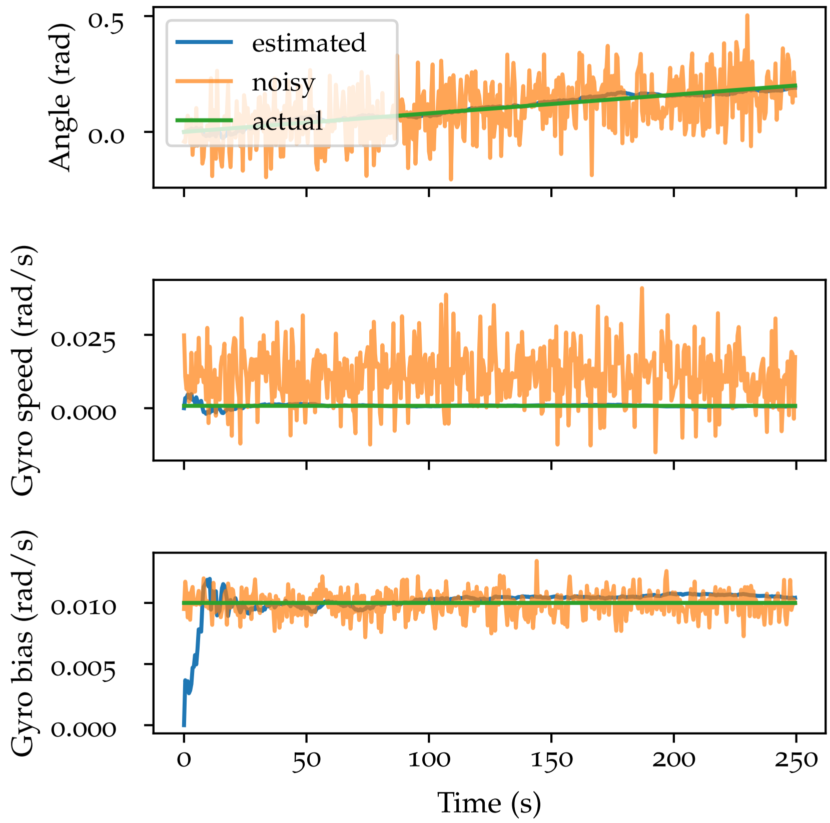


Figure 19.9: Heading estimation with a Kalman filter.

1.3 *Kalman filter considerations*

A Kalman filter requires estimates of the process noise covariance matrix and the measurement noise covariance matrix. It also requires an initial guess for the state vector and its covariance matrix. If the initial guess is poor, then the variances of the covariance matrix must be large otherwise the Kalman filter will be slow to respond. By specifying low variances you are telling the Kalman filter to put more emphasise on the poor initial guess compared to the sensor measurements.

Calculating the Kalman gain requires a matrix inverse and matrix multiplications. The latter is straightforward but the former is tedious to implement on a small microcontroller. Moreover, floating point calculations are required but these are slow on small microcontrollers since they need to be emulated with integer arithmetic.

One approach to avoid a matrix inverse is to fix the Kalman gains. This is called a steady state Kalman filter. It calculates the estimated state using:

$$\hat{\mathbf{x}}_n^+ = \mathbf{A}\hat{\mathbf{x}}_{n-1}^+ + \mathbf{K}\left(\mathbf{z}_n - \mathbf{C}\mathbf{A}\hat{\mathbf{x}}_{n-1}^+\right). \tag{19.38}$$

While this is simple to implement, it cannot adapt to the measurements. If good smoothing is required, then a good guess is required for the initial state otherwise the filter will take a long time to change, see for example Figure 19.10.

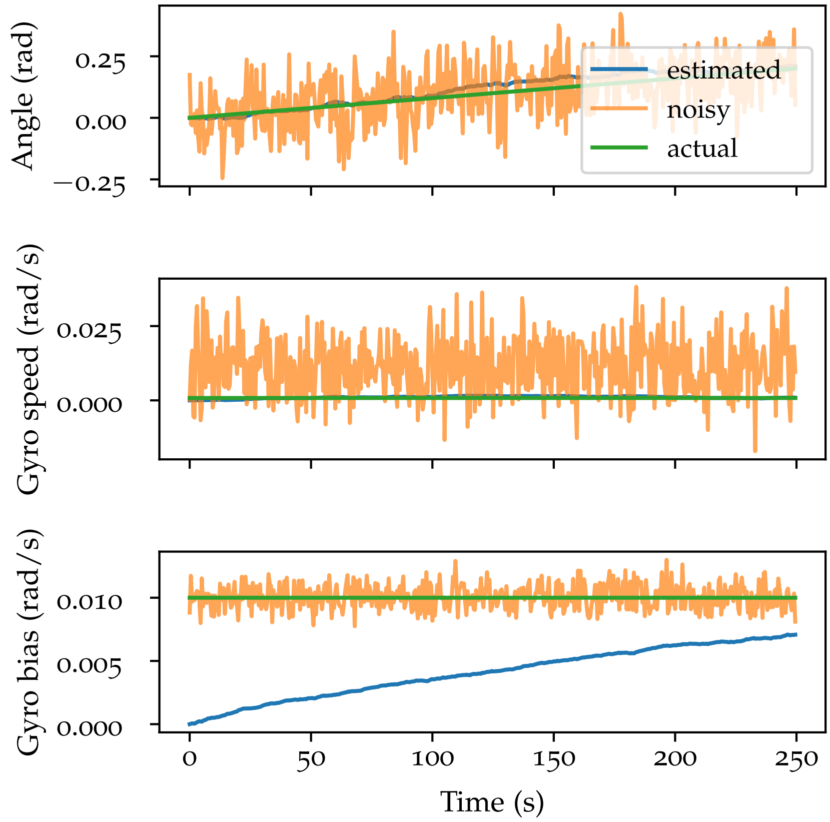


Figure 19.10: Heading estimation with a steady state Kalman filter.

20
PID control as a digital filter

1 Continuous time PID control

The output of a PID controller is given by

$$c(t) = K_p e(t) + K_i \int_0^t e(u)\mathrm{d}u + K_d \frac{\mathrm{d}e(t)}{\mathrm{d}t}, \tag{20.1}$$

where $e(t)$ is the error signal. Taking Laplace transforms,

$$C(s) = K_p E(s) + \frac{1}{s} K_i E(s) + s K_d E(s), \tag{20.2}$$

and thus the transfer function is

$$H(s) = \frac{C(s)}{E(s)} = K_p + \frac{1}{s} K_i + s K_d. \tag{20.3}$$

2 Discrete time PID control

There are several approaches to derive a discrete-time PID controller. If time derivatives of the error signal are approximated with backward differences then

$$\frac{\mathrm{d}e(t)}{\mathrm{d}t} \approx \frac{e_n - e_{n-1}}{\Delta t}, \tag{20.4}$$

and so

$$s \approx \frac{1}{\Delta t}\left(1 - z^{-1}\right). \tag{20.5}$$

With this result the discrete-time transfer function for a PID controller is

$$H(z) = \frac{C(z)}{E(z)} = K_p + \frac{\Delta t}{1 - z^{-1}} K_i + \frac{1 - z^{-1}}{\Delta t} K_d, \tag{20.6}$$

which corresponds to the difference equation

$$c_n = c_{n-1} + K_p(e_n - e_{n-1}) + K_i \Delta t\, e_n + \frac{K_d}{\Delta t}\left(e_n + 2e_{n-1} - e_{n-2}\right). \tag{20.7}$$

This can be expressed as a recursive filter

$$c_n = a_1 c_{n-1} + b_0 e_n + b_1 e_{n-1} + b_2 e_{n-2}, \tag{20.8}$$

where

$$a_1 = 1, \tag{20.9}$$

$$b_0 = K_p + K_i \Delta t + \frac{K_d}{\Delta t}, \tag{20.10}$$

$$b_1 = -K_p + \frac{2K_d}{\Delta t}, \tag{20.11}$$

$$b_2 = -\frac{K_d}{\Delta t}. \tag{20.12}$$

2.1 *Other implementations*

The implementation of (20.7) is called the *velocity PID algorithm*. Another approach, called the *positional PID algorithm*, integrates the position separately so

$$q_n = q_{n-1} + \Delta t \, e_n, \tag{20.13}$$

$$c_n = K_p e_n + K_i q_n + \frac{K_d}{\Delta t} \left(e_n - e_{n-1} \right). \tag{20.14}$$

This is a combination of a recursive filter for q_n and a moving-average filter for c_n.

Both the positional and velocity PID algorithms derive from the same continuous-time transfer function but have different numerical behaviour. Indeed, there are many other approaches depending on how the integration and differentiation is approximated numerically. Some try to better match the impulse response; others try to better match the frequency response. One popular approach is Tustin's algorithm where

$$s = \frac{2}{\Delta t} \frac{1 - z^{-1}}{1 + z^{-1}}. \tag{20.15}$$

With this choice, the difference equation is

$$c_n = c_{n-2} + K_p \left(e_n - e_{n-2} \right) + K_i \frac{\Delta t}{2} \left(e_n + 2e_{n-1} + e_{n-2} \right) + K_d \frac{2}{\Delta t} \left(e_n - 2e_{n-1} + e_{n-2} \right). \tag{20.16}$$

21
Reference

1 Periodic signal synthesis

A periodic signal can be represented as a superposition of harmonics,

$$x(t) = \sum_{k=1}^{\infty} A_k \sin\left(2\pi k f_0\right), \tag{21.1}$$

here f_0 is the fundamental frequency and A_k is the amplitude of the k-th harmonic of frequency $f_K = k f_0$.

1.1 Square wave synthesis

The amplitudes of a square wave are given by

$$A_k = \frac{4}{\pi} \begin{cases} 0, & k \text{ even}, \\ \frac{1}{k}, & k \text{ odd}. \end{cases} \tag{21.2}$$

1.2 Triangle wave synthesis

The amplitudes of a triangle wave are given by

$$A_k = \begin{cases} 0, & k \text{ even}, \\ \frac{-1^{\frac{k-1}{2}} 8}{\pi^2 k^2}, & k \text{ odd}. \end{cases} \tag{21.3}$$

2 Responses

The transfer function is defined by

$$H(s) = \frac{Y(s)}{X(s)} \tag{21.4}$$

on the assumption of no initial conditions. The frequency response is found from a transfer function with $s = \mathrm{j}2\pi f$,

$$H_f(f) = H(\mathrm{j}2\pi f). \tag{21.5}$$

The impulse response is found from the inverse Laplace transform of the transfer function,

$$h(t) = \mathcal{L}^{-1}\left\{H(s)\right\}. \tag{21.6}$$

It can also be found from the inverse Fourier transform of the frequency response,

$$h(t) = \mathcal{F}^{-1}\left\{H_f(f)\right\}. \tag{21.7}$$

The step response is found from the inverse Laplace transform of the transfer function divided by s,

$$g(t) = \mathcal{L}^{-1}\left\{\frac{H(s)}{s}\right\}. \tag{21.8}$$

It can also be found from the time-integral of the impulse response,

$$g(t) = \int_{-\infty}^{t} h(\tau)\mathrm{d}\tau. \tag{21.9}$$

The AC response to a frequency f can be found from

$$y(t) = A\left|H(\mathrm{j}2\pi f)\right|\cos\left(2\pi f t + \angle H(\mathrm{j}2\pi f t)\right), \tag{21.10}$$

where the input signal is

$$x(t) = A\cos\left(2\pi f t\right). \tag{21.11}$$

2.1 *First-order low-pass filter*

The transfer function is

$$H(s) = \frac{\alpha}{s+\alpha}, \tag{21.12}$$

with a pole at $s = -\alpha$.

$$h(t) = \alpha e^{-\alpha t}u(t) \tag{21.13}$$

$$g(t) = (1 - e^{-\alpha t})u(t) \tag{21.14}$$

2.2 *First-order high-pass filter*

The transfer function is

$$H(s) = \frac{s}{s+\alpha}, \tag{21.15}$$

with a pole at $s = -\alpha$ and a zero at $s = 0$.

$$h(t) = \delta(t) - \alpha e^{-\alpha t}u(t) \tag{21.16}$$

$$g(t) = e^{-\alpha t}u(t) \tag{21.17}$$

2.3 *Second-order low-pass filter*

The transfer function can be parameterised by the undamped angular resonant frequency, ω_0, and the damping factor, ζ,

$$H(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 + \omega_0^2} \tag{21.18}$$

There are three cases depending on the damping factor

$\zeta < 1$ *(under damped)* has complex conjugate poles at

$$s = -\omega_0\zeta \pm \mathrm{j}\omega_0\sqrt{1-\zeta^2} = -\alpha_1 \pm \mathrm{j}\omega_1. \tag{21.19}$$

The impulse response is

$$h(t) = \frac{\left(\alpha_1^2 + \omega_1^2\right) e^{-\alpha_1 t} \sin\left(\omega_1 t\right) u\left(t\right)}{\omega_1}, \tag{21.20}$$

and the step response is

$$g(t) = \left(-\frac{4\alpha_1\omega_1\left(\alpha_1^2 + \omega_1^2\right) e^{-\alpha_1 t} \sin\left(\omega_1 t\right)}{4\alpha_1^2\omega_1^2 + 4\omega_1^4} + \frac{4\omega_1^2\left(-\alpha_1^2 - \omega_1^2\right) e^{-\alpha_1 t} \cos\left(\omega_1 t\right)}{4\alpha_1^2\omega_1^2 + 4\omega_1^4} + 1\right) u\left(t\right). \tag{21.21}$$

$\zeta = 1$ *(critically damped)* has two repeated poles at

$$s = -\omega_0, -\omega_0. \tag{21.22}$$

The impulse response is

$$h(t) = \omega_0^2 t e^{-\omega_0 t} u\left(t\right), \tag{21.23}$$

and the step response is

$$g(t) = \left(-\omega_0 t e^{-\omega_0 t} + 1 - e^{-\omega_0 t}\right) u\left(t\right). \tag{21.24}$$

$\zeta > 1$ *(over damped)* has two real poles at

$$s = -\omega_0\zeta \pm \omega_0\sqrt{\zeta^2 - 1} = -\alpha_1, -\alpha_2. \tag{21.25}$$

The impulse response is

$$h(t) = \left(\frac{\alpha_1\alpha_2 e^{-\alpha_2 t}}{\alpha_1 - \alpha_2} - \frac{\alpha_1\alpha_2 e^{-\alpha_1 t}}{\alpha_1 - \alpha_2}\right) u\left(t\right), \tag{21.26}$$

and the step response is

$$g(t) = \left(-\frac{\alpha_1 e^{-\alpha_2 t}}{\alpha_1 - \alpha_2} + \frac{\alpha_2 e^{-\alpha_1 t}}{\alpha_1 - \alpha_2} + 1\right) u\left(t\right). \tag{21.27}$$