Australian Government
Department of Defence
Defence Science and
Technology Organisation

How to Create and Manipulate Radar Range–Doppler Plots

Don Koks

Cyber & Electronic Warfare Division

Defence Science and Technology Organisation

DSTO–TN–1386

ABSTRACT

This report lays out the mathematical framework and reasoning involved in addressing the question of how to produce sophisticated false targets in both range and Doppler to jam a radar. The explanations here are given from first principles in order to clarify the approach used in writing software to address the task. They constitute standard radar theory seen through a physicist's eyes, and are recorded as background to the approach taken in addressing the jamming task in a later report.

We discuss in detail how a radar generates a range–Doppler plot, using a set of parameters that describe the outgoing radar signal and the relevant characteristics of the target whose impulse response is known. We also explain necessary concepts from a mathematical physicist's viewpoint: bounds on pulse parameters, correlation/convolution, the theory of Hilbert transforms, the relevant Fourier analysis, and general concepts of radar signal processing such as ambiguity functions and the maximum detectable range of a target. This entire discussion is aimed at indicating the approach and philosophy used to solve the various problems encountered while working on the task.

APPROVED FOR PUBLIC RELEASE

Published by

DSTO Defence Science and Technology Organisation
PO Box 1500
Edinburgh, SA 5111, Australia

Telephone: (08) 7389 5555
Facsimile: (08) 7389 6567

AR No. AR–016–185
December 2014

APPROVED FOR PUBLIC RELEASE

How to Create and Manipulate Radar Range–Doppler Plots

Executive Summary

This report lays out the mathematical framework and reasoning involved in addressing the question of how to produce sophisticated false targets in both range and Doppler to jam a radar. The explanations here are given from first principles in order to clarify the approach used in writing software to address the task. They constitute standard radar theory seen through a physicist's eyes, and are recorded as background to the approach taken in addressing the jamming task in a later report, currently in publication.

We first derive the relevant radar equations that allow the radar scenario to be modelled numerically. An in-depth discussion of correlation follows, which is central to the operation of a matched filter. Next we explain how to model the returned signal from knowledge of the target's impulse response. Following a discussion of Doppler windowing is an analysis of what constitutes valid and useful pulse and processing parameters, such as sampling interval and bandwidth, with an explanation of blind speed and velocity aliasing. We then explain the mathematics of how a jammer can add time-dependent phases to a signal with the aim of manipulating that signal's Doppler content, together with the simpler manipulation of its range content carried out by adding delay to the signal.

Appendices discuss the Hilbert transform and the relation of correlation to convolution, including first-principles examples of these. Calculation of signal-to-noise ratios and how to combine the cross sections of many scatterers is also included. The report closes with an explanation of the use of the ambiguity function. Explanations of textbook concepts of radar signal processing have been given in this report as a way of indicating the approach and philosophy used to solve the various problems encountered while working on the jamming task.

This page is intentionally blank.

Author

Don Koks
Cyber & Electronic Warfare Division

Don Koks completed a doctorate in mathematical physics at Adelaide University in 1996, with a thesis describing the use of quantum statistical methods to analyse decoherence, entropy and thermal radiance in both the early universe and black hole theory. He holds a Bachelor of Science from the University of Auckland in pure and applied mathematics and physics, and a Master of Science in physics from the same university with a thesis in applied accelerator physics (proton-induced X ray and γ ray emission for trace element analysis). He has worked on the accelerator mass spectrometry programme at the Australian National University in Canberra, and in commercial Internet development.

Currently he is a Research Scientist with the Radio Frequency Technologies group in the Cyber and Electronic Warfare Division at DSTO, specialising in radar signal processing, 6-degree-of-freedom orientation concepts, and related specialties such as satellite orbital prediction. He is the author of the book *Explorations in Mathematical Physics: the Concepts Behind an Elegant Language* (Springer, 2006).

This page is intentionally blank.

Contents

Appendices

1 Introduction

This report lays out the mathematics involved in simulating the jamming of a radar to induce it to generate false targets having structure in both range and Doppler. We begin with the basic concepts of radar signal processing, then build on them, step by step, to arrive at the relevant false-target mathematics. This mathematics can then be used to simulate numerically the interaction of a radar with a target composed of perhaps many thousands of scatterers of known cross section. By simulating how the radar interacts, we can then also simulate jamming it to produce false targets.

We start by setting up the relevant radar equations that allow the radar scenario to be modelled numerically. An in-depth discussion of correlation follows, which is central to the operation of a matched filter. After a discussion of Doppler windowing and why it is generally used, is an analysis of what constitutes valid and useful pulse and processing parameters such as sampling interval and bandwidth, with an explanation of blind speed and velocity aliasing. We then explain how time-dependent phases can be added to a "returned" signal sent by a jammer to a radar with the aim of manipulating that signal's target-velocity content, together with the simpler manipulation of its range content which can be done by adding delay to the signal. Finally, appendices discuss the Hilbert transform, the relation of correlation to convolution (with examples of how correlation and convolution are implemented from first principles), signal-to-noise theory, and the use of the ambiguity function.

2 The Mathematics of Emitted and Received Signals

The pages that follow show each step of assembling the mathematics that describes a radar interaction. We begin with a discussion of the use of complex numbers to represent a real signal.

The electromagnetic field that comprises a propagating radar signal is very complicated both in space and in time. But there is no requirement to analyse it exactly: we need only consider its far-field component, and of this, it's sufficient to work with the electric field only, since that is always related to the magnetic field in a well-defined way. At any given point, the electric field of a radar signal can be represented by a unit-length vector (the field's polarisation) which is multiplied by a number that varies sinusoidally in time with frequency f. We will assume the vector is given (i.e. the polarisation is known), and will represent the field by the sinusoidally varying number alone.

> Sinusoids are particularly useful for signal processing, being a special case of signals with an exponential form. An exponential signal is unique in that its form is preserved when it passes through a *linear time-shift-invariant* system. A linear system can be treated as processing a sum of signals component-wise and adding the results; a time-shift-invariant system is one that processes identically today as it did yesterday. Most of the important systems in the signal processing world are linear time-shift invariant. The usefulness of the exponential signal here embodies the fact that exponential signals are the eigenfunctions of linear time-shift-invariant systems.

The information content of a signal is almost always placed as a modulation onto a high-frequency *carrier wave*. The use of this carrier is only partly dictated by available wave-generating technology and transparency of Earth's atmosphere to different wavelengths.

One main reason for transmitting information on a high-frequency carrier follows from the fact that waves of length λ emitted coherently[1] from an aperture of diameter a will diffract to form a beam of angular width approximately λ/a. Only by using relatively high frequencies (meaning small λ) will a beam's width be small enough that it will deposit a good fraction of its energy on a distant target. Another reason for using high frequencies follows from the fact that the radio waves travel at a set speed. To transmit more information, we can't hope to send the same wave at a higher speed. Instead, we must place a higher density of information into the wave, by giving the modulation more structure: and this we do by shortening the duration of the wave's modulation changes that encode the information. Thus the wave's shape must be made to change more rapidly, and Fourier analysis tells us that such rapid changes require the signal to be a sum of monochromatic waves over a large spread of frequencies. (That is, high data rates are built from large *bandwidth*.) The bandwidth is spread around the carrier frequency, and if the frequencies making up a large bandwidth are all to be high enough to ensure that the beam has the required low angular width, then the carrier frequency must be high.

Yet another reason for using a high carrier frequency involves the range–velocity tradeoff examined on page 34: higher carrier frequencies enable the radar to measure both larger ranges *and* higher radial velocities in a single measurement. But these higher frequencies also produce more "blind speeds", discussed on page 35.

Suppose then that we have formed a signal by modulating both the amplitude and phase of a carrier wave of angular frequency $\omega_0 = 2\pi f_0$. We have given the signal a large bandwidth, but in practice this bandwidth is still *much* less than the carrier frequency. In that case, the amplitude and phase do not vary so drastically as to alter the signal radically from a recognisable sinusoid; in other words, the signal is confined to a *relatively* narrow frequency interval around (and compared to) its carrier. This type of signal is often referred to as *narrow band*; even though it might have a large bandwidth, it is "relatively" narrow band compared to the carrier. (In contrast, a signal that is not narrow band is not usefully described by the "analytic signal" analysis that follows.) We can represent a narrow-band signal by a sine or cosine; the choice is arbitrary since the two functions differ only by a constant phase. Suppose we choose a sine, writing the signal as $A(t)\sin[\omega_0 t + \phi(t)]$ where $A(t)$ and $\phi(t)$ are the signal's instantaneous amplitude and phase respectively. (We'll drop the time dependence of A and ϕ for brevity in the equations that follow.) The task of the radar receiver is to determine A and ϕ from one measurement to the next. Note that "phase" is sometimes taken to mean $\phi(t)$ and sometimes $\omega_0 t + \phi(t)$, but there should be no ambiguity in the discussion that follows.

Focus on determining ϕ, since we'll see that the amplitude A emerges along with ϕ. The phase is an angle, and an angle always needs two pieces of information to be fully determined. Certainly any angle is just one number, but it's the ratio of two quantities: the arc length that the angle scribes on a given circle, and the radius of that circle. Ordinarily the radius is set to have length one, in which case the angle is a single number, the length of the arc—but this is a convention. Another way of realising that an angle needs two

[1] By *coherent* waves is meant a set of wave fronts with constant wavelength, direction, and relative phase. Electromagnetic waves produced by radar transmitters and lasers are coherent; electromagnetic waves produced by a household torch are not. The waves flowing from a torch *are* coherent over tiny time scales and tiny distances, but this coherence is constantly changing and so is somewhat trivial. *Real* coherence is all about the large-scale effects of wave interference, so it requires "tidy" waves at large scales and long periods of time.

pieces of information to be fully determined is to represent the angle by a vector of any length r in the xy plane; the vector's two components, $r\cos\phi$ and $r\sin\phi$, are the required two pieces of information that fully determine ϕ. In fact the length r is redundant, since it can be extracted from $r\cos\phi$ and $r\sin\phi$ by summing the squares of these two numbers. So we can ignore it, and our two pieces of information that will fully determine the phase ϕ of the signal will be $\cos\phi$ and $\sin\phi$. In principle, these can be extracted in the following way when representing the signal by a sine or by a cosine. (More discussion of this in a radar hardware context can be found in [1].)

Representing the Signal by a Sine: Write the signal as $A\sin(\omega_0 t+\phi)$ and extract $\cos\phi$ and $\sin\phi$ by *mixing* (multiplying) the signal with *reference signals* $2\sin\omega_0 t$ and $2\cos\omega_0 t$. In equations (2.2)–(2.8) that follow, we will employ the following standard trigonometric identities:

$$\begin{aligned} -2\sin\alpha\sin\beta &= \cos(\alpha+\beta)-\cos(\alpha-\beta)\,, \quad & 2\sin\alpha\cos\beta &= \sin(\alpha+\beta)+\sin(\alpha-\beta)\,, \\ 2\cos\alpha\cos\beta &= \cos(\alpha+\beta)+\cos(\alpha-\beta)\,, \quad & \cos(\alpha+\beta) &= \cos\alpha\,\cos\beta-\sin\alpha\,\sin\beta\,. \end{aligned} \tag{2.1}$$

First, mix the signal with $2\sin\omega_0 t$:

$$A\sin(\omega_0 t+\phi)\,2\sin\omega_0 t = \underbrace{A\cos\phi}_{\text{“I data”}} - \underbrace{A\cos(2\omega_0 t+\phi)}_{\text{filtered out}}\,. \tag{2.2}$$

The second term on the right-hand side of (2.2) has a high frequency and so is removed by a low-pass filter. The first term is our first piece of information that determines ϕ. Because it was produced by mixing our representative signal (a sine) with an oscillation described in the same way (i.e. also a sine), it's called the *in-phase data* (“I data”) of the signal.

Now mix the signal with $2\cos\omega_0 t$:

$$A\sin(\omega_0 t+\phi)\,2\cos\omega_0 t = \underbrace{A\sin\phi}_{\text{“Q data”}} + \underbrace{A\sin(2\omega_0 t+\phi)}_{\text{filtered out}}\,. \tag{2.3}$$

Again the second term is filtered out, and the first term is called the signal's *quadraphase data*[2] (“Q data”). Traditionally, the I and Q data are combined into a phasor: a vector with xy coordinates $(I,Q)=A(\cos\phi,\sin\phi)$. The amplitude $A(t)$ of the signal is the time-varying length of this phasor.

The above mixing process is reversed to convert the I/Q data back to a modulated signal ready to be sent by a radar transmitter. The I data in (2.2) resulted from mixing the signal with a sine, so now mix the I with a sine; the Q data in (2.3) resulted from mixing the signal with a cosine, so now mix the Q with a cosine. Then add the results:

$$\underbrace{A\cos\phi}_{\text{I data}}\,\sin\omega_0 t + \underbrace{A\sin\phi}_{\text{Q data}}\,\cos\omega_0 t = A\sin(\omega_0 t+\phi)\,. \tag{2.4}$$

The resulting signal $A\sin(\omega_0 t+\phi)$ can then be sent out by the transmitter.

[2] The use of “quad” here, as in quadrant or square, relates to the idea of a square's right-angled corners: these mimic the right angle between two phasors that represent a sine and a cosine.

Representing the Signal by a Cosine: If we choose instead to write the signal as $A\cos(\omega_0 t+\phi)$ (where this ϕ differs from that used in the sine representation above), very little changes in the above analysis. Again we mix the signal with $2\sin\omega_0 t$ and $2\cos\omega_0 t$. This time begin with $2\cos\omega_0 t$:

$$A\cos(\omega_0 t+\phi)\,2\cos\omega_0 t=\underbrace{A\cos\phi}_{\text{I data}}+\underbrace{A\cos(2\omega_0 t+\phi)}_{\text{filtered out}}\ . \tag{2.5}$$

The second term is filtered out, and because the first term was produced by mixing our representative signal (a cosine) with an oscillation described in the same way (i.e., a cosine), it's called the I data of the signal, just as for the sine case above. Now mix the signal with $2\sin\omega_0 t$:

$$A\cos(\omega_0 t+\phi)\,2\sin\omega_0 t=\underbrace{-A\sin\phi}_{\text{Q data}}+\underbrace{A\sin(2\omega_0 t+\phi)}_{\text{filtered out}}\ . \tag{2.6}$$

Filter the second term out and, as before, call the first term the signal's Q data. Again the I and Q data can be combined into a phasor, but this time a minus sign is included: $(I,-Q)=A(\cos\phi,\sin\phi)$. We see that the resulting phasor is identical to that formed when representing the signal by a sine (that is, if we include the minus sign).

Again the I/Q data can be converted to a signal ready for transmitting. The I data in (2.5) resulted from mixing the signal with a cosine, so now mix the I with a cosine; the Q data in (2.6) resulted from mixing the signal with a sine, so now mix the Q with a sine. Then add the results:

$$\underbrace{A\cos\phi}_{\text{I data}}\,\cos\omega_0 t\quad-\underbrace{A\sin\phi}_{\text{Q data}}\,\sin\omega_0 t=A\cos(\omega_0 t+\phi)\,. \tag{2.7}$$

The resulting signal $A\cos(\omega_0 t+\phi)$ is now sent out by the transmitter.

The above mixing procedure is all about our choosing to represent a sinusoid as either a sine or a cosine. We must choose one or the other, and the choice corresponds to our deciding what to label as the I data and what to label as the Q data after the mixing process. "I data" refers to the output of mixing the signal with the sinusoid that will represent the signal; i.e., to the output of mixing the signal with $2\sin\omega_0 t$ if we choose to represent the signal by a sine, and $2\cos\omega_0 t$ if we choose to represent the signal by a cosine. The I and Q numbers then describe the signal completely.

Refinements from Superhet Receivers In practice, most radar receivers are of the *superheterodyne* type; instead of removing the carrier in one step, superheterodynes remove it in a series of smaller steps as a way of improving the final signal-to-noise ratio. But the procedure at each step is identical to that described above. For example, begin with a signal $A\sin(\omega_0 t+\phi)$ and mix it with a reference wave that incorporates an *intermediate frequency* ω_{IF}:

$$A\sin(\omega_0 t+\phi)\,2\cos[(\omega_0-\omega_{\text{IF}})t]=A\sin(\omega_{\text{IF}}t+\phi)+A\sin[(2\omega_0-\omega_{\text{IF}})t+\phi]\,. \tag{2.8}$$

The second term on the right-hand side of (2.8) is now filtered out, leaving the first term which is just like the original signal but with a new, lower-frequency carrier ω_{IF}. The mixing/filtering process is now applied to this term (using a still lower intermediate frequency) in one or more steps until we have removed the carrier entirely. Along the way several stages of amplification will also have been applied.

2.1 Using Complex Numbers to Represent I/Q Data

The numbers A and ϕ—really $A(t), \phi(t)$—give a phasor $A(\cos\phi, \sin\phi)$ whose relatively small changes over time embody the modulation of the carrier wave. A phasor of the actual radio-frequency signal can be constructed by making the A-ϕ phasor spin at constant angular speed ω_0. To see why, simply rotate the A-ϕ phasor by angle $\omega_0 t$ using a matrix multiplication:

$$\text{spinning phasor} = \underbrace{\begin{bmatrix} \cos\omega_0 t & -\sin\omega_0 t \\ \sin\omega_0 t & \cos\omega_0 t \end{bmatrix}}_{\text{rotate vector by } \omega_0 t} \underbrace{\begin{bmatrix} A\cos\phi \\ A\sin\phi \end{bmatrix}}_{\substack{\text{vector to be} \\ \text{rotated}}} = \underbrace{\begin{bmatrix} A\cos(\omega_0 t + \phi) \\ A\sin(\omega_0 t + \phi) \end{bmatrix}}_{\text{rotated vector}}, \tag{2.9}$$

so that the radio-frequency signal is either the x- or the y-component of this rapidly spinning phasor.

Any phasor is traditionally a vector $\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]$, but converting it to the complex number $a + \mathrm{i}b$ makes it arguably easier to manipulate. For example, in the vector picture, to increase the phase by angle θ we rotate the phasor vector using a matrix multiplication:

$$\begin{bmatrix} a \\ b \end{bmatrix}_{\text{rotated by } \theta} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}; \tag{2.10}$$

in contrast, the complex-number version of (2.10) employs "everyday" multiplication:

$$(a + \mathrm{i}b)_{\text{rotated by } \theta} = (\cos\theta + \mathrm{i}\sin\theta)\,(a + \mathrm{i}b)\,. \tag{2.11}$$

This complex notation is compact but not mystical. Equation (2.11) is a "rendering down" of (2.10) in which the seemingly redundant repetition of the sine and cosine in the rotation matrix has been removed, at the small cost of introducing an "i".[3] The term $\cos\theta + \mathrm{i}\sin\theta$ is written as $\mathrm{e}^{\mathrm{i}\theta}$ by *defining* the exponential of a complex number to have the same Taylor-series form as the exponential of a real number.[4] Complex notation converts (2.9) to

$$\text{“circling” complex number} = \underbrace{\mathrm{e}^{\mathrm{i}\omega_0 t}}_{\substack{\text{rotate complex} \\ \text{number by } \omega_0 t}} \quad \underbrace{A\mathrm{e}^{\mathrm{i}\phi}}_{\substack{\text{complex number} \\ \text{to be rotated}}} = \underbrace{A\mathrm{e}^{\mathrm{i}(\omega_0 t + \phi)}}_{\substack{\text{rotated complex} \\ \text{number}}}, \tag{2.12}$$

which can (arguably) be thought of as simplifying the mathematics of rotating phasors: compare the elaborate layout of (2.9) with the simplicity of (2.12).

The process of extracting the I/Q content at time t from a rotating phasor can now be viewed as multiplying it by $\mathrm{e}^{-\mathrm{i}\omega_0 t}$, which removes its rapidly rotating carrier content (or equivalently, switches to a rotating frame in phasor-space!). Likewise, converting I/Q data to a signal ready to send out at time t can be viewed as multiplying its phasor by $\mathrm{e}^{\mathrm{i}\omega_0 t}$.

[3] Here we have rendered down a two-dimensional rotation matrix. When the same idea is applied to three-dimensional rotations, what results is *quaternions*, whose $\mathrm{i}, \mathrm{j}, \mathrm{k}$ seem to attract mystical descriptions by some in the computer graphics community. But again, quaternions can be treated as simply the result of rendering down redundant matrix information while retaining "everyday" multiplication. In practice, it's tedious to multiply quaternions using $\mathrm{i}, \mathrm{j}, \mathrm{k}$; instead we multiply them using a non-standard multiplication. But aside from that, quaternions still replace a 9-element matrix by a 4-element array, so have advantages of economy in numerical work.

[4] That is, $\mathrm{e}^{\mathrm{i}\theta} \equiv 1 + \mathrm{i}\theta + (\mathrm{i}\theta)^2/2! + \cdots = \cos\theta + \mathrm{i}\sin\theta$, where the last equality comes about by noting that the real and imaginary parts of the complex series are identical to the Taylor series of cosine and sine respectively. Note that this sum *defines* $\mathrm{e}^{\mathrm{i}\theta}$, because there is otherwise no a priori meaning to a complex exponential; the series expansion defines it in the same way as if $\mathrm{i}\theta$ were real. But a small amount of analysis (found in books on complex variables) shows that this definition gives $\mathrm{e}^{\mathrm{i}\theta}$ all the behaviour that we know and expect of a *real* exponential—behaviour such as $\mathrm{e}^{\mathrm{i}\theta}\mathrm{e}^{\mathrm{i}\phi} = \mathrm{e}^{\mathrm{i}(\theta+\phi)}$, and so on.

2.2 Analytic Signals and Fourier Analysis

Regardless of whether a real signal is written as $A(t)\sin[\omega_0 t + \phi(t)]$ or $A(t)\cos[\omega_0 t + \phi(t)]$, the corresponding complex-number phasor $A(t)\mathrm{e}^{\mathrm{i}[\omega_0 t+\phi(t)]}$ produced by (2.12) is called the (corresponding) *analytic signal*. Its phasor rotates counter-clockwise with an instantaneous angular velocity of $\mathrm{d}/\mathrm{d}t\,(\text{phasor angle}) = \omega_0 + \phi'(t)$, which we have assumed is not too different from ω_0 because the signal is narrow band.

This conversion of a real signal to an analytic signal is related to, but crucially not the same as, the idea of complex-Fourier analysing the real signal. It's instructive to compare these processes of (1) converting a real signal to its analytic signal and (2) Fourier decomposing the real signal into complex components. Consider the real signal $\cos\omega_0 t$ with $\omega_0 > 0$—hardly a useful signal, but a sufficient example for this discussion. Note that a real signal can always be written using only positive frequencies; the very idea of a negative frequency emerges only in the following Fourier analysis.

Converting the real signal to its analytic signal: The analytic signal formed from the real signal $\cos\omega_0 t$ is a single phasor $\mathrm{e}^{\mathrm{i}\omega_0 t}$. Both the real signal and its analytic signal are considered to have the same *single* frequency ω_0, which is always positive.

Complex-Fourier decomposing the real signal: A real signal can always be Fourier decomposed into sines and cosines. This decomposition requires two sets of summations and integrations, one for the sines $\sin\omega t$ and one for the cosines $\cos\omega t$ (the zero subscript is omitted as ω is an index of summation or integration), so that all terms in the Fourier analysis tend to appear twice. But $\sin\omega t$ and $\cos\omega t$ can both be written as linear combinations of the complex exponentials $\mathrm{e}^{\mathrm{i}\omega t}$ and $\mathrm{e}^{-\mathrm{i}\omega t}$. Of course, we would gain nothing by converting real sinusoids to complex exponentials if the result was merely to replace a "sine, cosine" pair with an "$\mathrm{e}^{\mathrm{i}\omega t}$, $\mathrm{e}^{-\mathrm{i}\omega t}$" pair. But if we allow the frequencies $\omega = 2\pi f$ in the complex Fourier decomposition to assume negative values as well as positive, this pair of complex exponentials collapses to just one: $\mathrm{e}^{\mathrm{i}\omega t}$, where ω now ranges over *all* real numbers, negative as well as positive, in the summations and integrations. So the Fourier decomposition writes $\cos\omega_0 t$ (with $\omega_0 > 0$) as a sum of two phasors $1/2\,\mathrm{e}^{\mathrm{i}\omega_0 t}$ and $1/2\,\mathrm{e}^{-\mathrm{i}\omega_0 t}$ rotating in opposite directions.

This use of negative frequencies in Fourier analysis might initially seem strange, but they are simply a way of halving the amount of Fourier notation needed—with the added huge benefit of rendering the complex Fourier transform and its inverse almost identical, which aids the mathematics. However, this Fourier language now allocates the idea of a single frequency ω not to $\sin\omega t$ and $\cos\omega t$, but rather to $\mathrm{e}^{\mathrm{i}\omega t}$. So when we Fourier-decompose the real signal $\cos\omega_0 t$ into two complex exponentials, $\cos\omega_0 t = 1/2\,\mathrm{e}^{\mathrm{i}\omega_0 t} + 1/2\,\mathrm{e}^{-\mathrm{i}\omega_0 t}$, we say that $\cos\omega_0 t$ contains *two* frequencies: ω_0 and $-\omega_0$. The negative frequency $-\omega_0$ contains no more information than the positive frequency ω_0, and is present only to simplify the Fourier language. Perhaps these two frequencies ω_0 and $-\omega_0$ should be called "Fourier frequencies" to reinforce the idea that they result from complex-Fourier decomposing a signal $\cos\omega_0 t$ ($\omega_0 > 0$) that is clearly simple harmonic motion at a *single* frequency ω_0. More usual is to call ω_0 and $-\omega_0$ simply the frequencies, while the positive ω_0 is called the *repetition rate*.

Table 1: *A comparison of how the basic functions are viewed in the analytic-signal picture versus the complex-Fourier analysis picture*

	Analytic-signal picture	Fourier picture
$e^{i\omega_0 t}$	One frequency $\omega_0 > 0$	One frequency $\omega_0 > 0$ or < 0
$\sin\omega_0 t$ or $\cos\omega_0 t$	One frequency $\omega_0 > 0$	Two frequencies, ω_0 and $-\omega_0$
General real signal	One instantaneous frequency $\omega(t) > 0$	Frequency spectrum with $\omega = -\infty \to +\infty$

Table 1 summarises the difference between the analytic signal and Fourier decomposing the real signal. To reiterate:

– When converting the real signal $\cos\omega_0 t$ to its analytic signal $e^{i\omega_0 t}$, both the real signal and the analytic signal are considered to have the same single frequency ω_0, which is always positive.

– Fourier analysis of the real signal $\cos\omega_0 t$ works with components $e^{i\omega t}$ with $\omega = \omega_0$ and $-\omega_0$, and these components are each considered to have a single frequency ω. The real signal $\cos\omega_0 t$ is then considered to have *two* frequencies, one positive and one negative.

The analytic signal results when a real signal is converted to a single phasor, and is precisely the single-phasor model of an oscillating wave that is described in any number of introductory physics texts. In contrast, complex-Fourier decomposing a real signal produces phasors in pairs with frequencies of opposite sign, whose utility lies in their simplification of signal processing mathematics.

Indeed, converting $\cos\omega_0 t$ to its analytic signal $e^{i\omega_0 t}$ is indistinguishable from singling out the positive-frequency term $1/2\, e^{i\omega_0 t}$ from the complex Fourier pair and doubling its amplitude. Because of this, signal processing textbooks sometimes describe creating the analytic signal as "discarding the negative frequency". Unfortunately, this mixes complex-Fourier language—which considers $\cos\omega_0 t$ to have two frequencies of opposite sign—with analytic-signal language, which recognises only positive frequencies and so considers $\cos\omega_0 t$ to have a single frequency. The analytic signal in principle has nothing to do with discarding negative frequencies, and yet it *can be constructed* in practice by discarding negative frequencies and doubling the weights of the positive frequencies. This is explained further in Appendix A.3.

The signals emitted by most radars are not pure sinusoids, but they tend to be sufficiently narrow band that a phasor can still meaningfully be used to portray them. In this report we assume that the conversion of the received real signal to an analytic signal (the production of I/Q data) has already occurred in the radar receiver, so that we deal purely with the analytic signal at all times. That is, the radar receiver can be viewed as converting the incoming real signal to a stream of complex numbers. We also write the signal just prior to being *emitted* as a complex number, remembering that in practice the mixing operation in (2.7) will then be used by the radar hardware to convert that complex number to a real signal, which is then physically emitted.

In general of course, a real signal $s(t)$ could be converted to any one of an infinite number of different complex signals: just add an imaginary part which can be *anything*. But only one of these complex signals, the analytic signal $s_c(t)$, is considered to be special, due to its useful properties discussed in Appendix A. That appendix describes the *Hilbert transform*, a procedure that constructs the analytic signal from a given real signal *after* the entire real signal has been received. This is in contrast to the mixing of the real signal with $\sin\omega_0 t$ and $\cos\omega_0 t$ that is done by a radar receiver *while* it's receiving that signal.

If the real signal is narrow band around a carrier frequency ω_0, the analytic signal will contain a factor of $\mathrm{e}^{\mathrm{i}\omega_0 t}$. This mathematical factoring out of the zero-information carrier from the information-carrying modulation renders some common signal processing manipulations easier and tidier. For example, consider an emitted carrier wave with frequency ω_0 that is bounced from a receding target and returns "red-shifted" as $\omega_0 - 2\,\mathrm{Hz}$. If we were to use real sinusoids in our analysis, we would compare the emitted wave's time-varying part $\sin\omega_0 t$ with the received time-varying part $\sin(\omega_0 - 2\,\mathrm{Hz})t$. To extract the Doppler frequency and its sign, we'd need to refer always to the carrier ω_0, and would note that the received frequency was less than the carrier by $2\,\mathrm{Hz}$, so that the Doppler frequency was $-2\,\mathrm{Hz}$, corresponding to a receding target. On the other hand, complex notation renders the extraction of the Doppler shift much more straightforward: we have an "emitted signal" $\mathrm{e}^{\mathrm{i}\omega_0 t}$ and a "received signal" $\mathrm{e}^{\mathrm{i}(\omega_0 - 2\,\mathrm{Hz})t}$, and one stage of our mathematical analysis simply removes the carrier by multiplying the returned signal by $\mathrm{e}^{-\mathrm{i}\omega_0 t}$, which we can do in principle by simply ignoring $\mathrm{e}^{\mathrm{i}\omega_0 t}$ in the mathematics. The result is $\mathrm{e}^{\mathrm{i}(-2\,\mathrm{Hz})\,t}$, which trivially yields an angular Doppler frequency of $-2\,\mathrm{Hz}$. So the Doppler frequency and its correct sign emerge immediately in the complex approach.

2.3 Visualising Transmission and Reception of a Signal

The above discussion makes the point that when a radio signal enters a radar receiver, the receiver produces a stream of *complex* numbers. It's not the case that we are merely using complex numbers as a redundant way to represent real numbers for mathematical convenience. Instead, after processing, the received signal has been converted to a stream of complex numbers, with no redundancy in its information content.

With this in mind, we begin by constructing the signal received by a radar whose transmitter and receiver are co-located (as they always are in this report). Write the narrow-band analytic signal emitted at time t at the transmitter/receiver site as

$$\text{transmitted signal } s_c(t) = u(t)\,\mathrm{e}^{\mathrm{i}\omega_0 t}\,, \tag{2.13}$$

where $u(t)$ is the complex form of the I/Q data: the signal's modulation. The signal only attains this far-field form at a distance from the radar of several wavelengths, by which time the near-field contribution has become insignificant. But that distance is small enough that we can consider the signal to be $s_c(t)$ at the radar itself.

Suppose that this signal is bounced off a target at a distance r, which may be moving with a constant radial velocity[5] $v = \mathrm{d}r/\mathrm{d}t$. We ask: what analytic signal $g_c(t)$ is received by the radar at time t? (Note that the word "analytic" is understood as always present and will be omitted from now on.)

[5] Note that v is positive for a receding target. Some radar books use this convention, while others define $v = -\mathrm{d}r/\mathrm{d}t$, making v negative for a receding target.

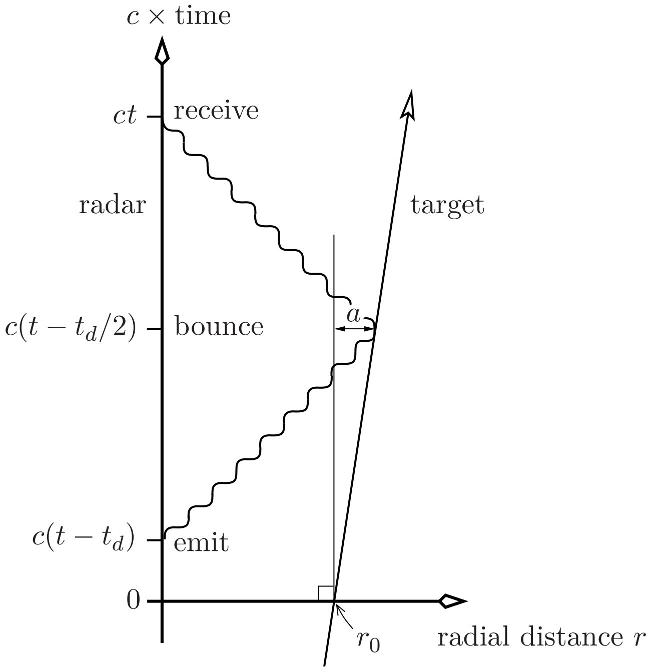


Figure 1: *A plot of $c \times$ time versus radial distance for the motion of both radar and target, where c is the speed of light. The scenario is analysed in the radar's frame, so the radar is at $r = 0$ for all time. The target is at $r = r_0$ at time 0. For a receding target, $v > 0$.*

The scenario is shown in Figure 1. With c the speed of light, this figure plots $c \times$ time versus radial distance for the motions of radar and target, and incorporates the radar signal. Because the vacuum speed of light is always measured by a constant-velocity emitter or receiver to have the same value (with corrections for air and a possibly accelerated frame being negligible here), we can, with complete generality, analyse the scenario in the radar's frame, meaning we can set the radar emitter/receiver to be at rest throughout the scenario. That is, only the relative motion of radar and target matters, so place the radar at $r = 0$ for the duration of the scenario, with the target at $r = r_0$ at $t = 0$. To facilitate later discussion, suppose that $t = 0$ starts a *coherent processing interval,* to be studied later in Section 5.

One very useful feature of this diagram is that because $\Delta r = c\Delta t$ for light, the radar signals always have a 45° slope, which allows us to make quick use of simple geometry to analyse the interactions presented on it.

The signal *emitted* at some general time t is $s_c(t) = u(t)\mathrm{e}^{\mathrm{i}\omega_0 t}$. What is *received* at some general time t? This received signal is essentially just a lower-amplitude version of what was emitted at the earlier time $t - t_d$, where t_d is the time delay from emission to reception. The signal emitted at $t - t_d$ was $u(t - t_d)\,\mathrm{e}^{\mathrm{i}\omega_0(t-t_d)}$. After bouncing from the target, this signal is received almost unchanged at time t except for the presence of (a) some amplitude $A(r_{\text{bounce}})$ that depends on the target's distance r_{bounce} at the time of bounce, and (b) a possible phase shift ϕ_0 generated when the signal interacted with the target, which we'll absorb into $A(r_{\text{bounce}})$ by making that amplitude complex. So the signal received at time t is

$$g_c(t) = A(r_{\text{bounce}})\, u(t - t_d)\, \mathrm{e}^{\mathrm{i}\omega_0 (t-t_d)}. \tag{2.14}$$

The geometry of 45° light rays in Figure 1 indicates immediately that

$$ct_d = 2r_{\text{bounce}}. \tag{2.15}$$

Thus (2.14) becomes

$$g_c(t) = A(r_{\text{bounce}})\, u(t - t_d)\, \mathrm{e}^{\mathrm{i}\omega_0 t}\, \mathrm{e}^{-\mathrm{i}\omega_0 2r_{\text{bounce}}/c}. \tag{2.16}$$

The target range at the time of bounce is $r_{\text{bounce}} = r_0 + v(t - t_d/2)$. How does this value compare with the range at the time of emission, $r_0 + v(t - t_d)$, and the range at the time of reception, $r_0 + vt$? The difference in both cases is $vt_d/2$. What is t_d? Since the target travels distance a at velocity v in time $t - t_d/2$, we know that

$$ct_d/2 = r_0 + a\,, \quad \text{and } a = v(t - t_d/2)\,. \tag{2.17}$$

It follows that

$$ct_d = \frac{2(r_0 + vt)}{1 + v/c} \simeq 2(r_0 + vt)\,, \tag{2.18}$$

where the last approximation holds because generally the target speed $|v|$ will be much less than c. Then

$$vt_d/2 = v/c \times ct_d/2 \simeq v/c\,(r_0 + vt)\,. \tag{2.19}$$

Suppose $r_0 = 10^4$ m, $v = 10$ m/s, and the current pulse is 32 pulses into a coherent processing interval. The time from the start of one pulse to the next is $100\,\mu$s, so that $t \simeq 32 \times 100\,\mu$s. The range difference $vt_d/2$ is then, making use of SI units,

$$vt_d/2 \;\simeq\; \frac{10}{3 \times 10^8} \times (10^4 + 10 \times 0.0032) \text{ metres } \;\approx\; 0.5 \text{ mm}. \tag{2.20}$$

Note also that $vt \simeq 3$ cm here, so $|vt| \ll r_0$, simplifying (2.19) to

$$t_d \simeq 2r_0/c\,. \tag{2.21}$$

The difference $vt_d/2$ between the ranges at emission, bounce, and reception is so small that we are free to replace the range at bounce in (2.16) with either the range at emission or reception. But it's important to note that comparing (2.15) with (2.21) doesn't allow us to replace r_{bounce} with the range at the start of the coherent processing interval r_0 when Doppler measurements are involved, because that replacement will discard *all* information about the target velocity, which won't allow later Doppler processing to be used. Even so, the amplitude of the pulse is not sensitive to the changing ranges involved, so *can* be calculated at the start of the coherent processing interval, meaning for range r_0.

Equation (2.16) becomes

$$\begin{aligned} g_c(t) &\simeq A(r_0)\,u(t-t_d)\,\mathrm{e}^{\mathrm{i}\omega_0 t}\begin{Bmatrix} \mathrm{e}^{-\mathrm{i}\omega_0 2/c\,\times\,\text{range at emission}} \\ \mathrm{e}^{-\mathrm{i}\omega_0 2/c\,\times\,\text{range at bounce}} \\ \mathrm{e}^{-\mathrm{i}\omega_0 2/c\,\times\,\text{range at reception}} \end{Bmatrix} \quad \text{(any choice is valid)} \\ &= A(r_0)\,u(t-t_d)\,\mathrm{e}^{\mathrm{i}\omega_0 t}\begin{Bmatrix} \mathrm{e}^{-\mathrm{i}\omega_0 2/c[r_0+v(t-t_d)]} \\ \mathrm{e}^{-\mathrm{i}\omega_0 2/c[r_0+v(t-t_d/2)]} \\ \mathrm{e}^{-\mathrm{i}\omega_0 2/c[r_0+vt]} \end{Bmatrix} \\ &= A(r_0)\,u(t-t_d)\,\mathrm{e}^{\mathrm{i}\omega_0 t}\,\mathrm{e}^{-\mathrm{i}\omega_0 2r_0/c}\begin{Bmatrix} \mathrm{e}^{-\mathrm{i}\omega_0 2v(t-t_d)/c} \\ \mathrm{e}^{-\mathrm{i}\omega_0 2v(t-t_d/2)/c} \\ \mathrm{e}^{-\mathrm{i}\omega_0 2vt/c} \end{Bmatrix}. \end{aligned} \tag{2.22}$$

The first option in (2.22) is the simplest when writing modelling code, and last option is the simplest to analyse mathematically:

Received signal when used for modelling:

$$g_c(t) \simeq A(r_0)\,u(\underbrace{t-2r_0/c}_{\text{time of emission}})\,\mathrm{e}^{\mathrm{i}\omega_0 t}\,\mathrm{e}^{-\mathrm{i}\omega_0 2/c\,\times\,\text{range at emission}} \tag{2.23}$$

Received signal when used for analysis:

$$g_c(t) \simeq A(r_0)\,u(t-2r_0/c)\,\mathrm{e}^{\mathrm{i}\omega_0 t}\,\mathrm{e}^{-\mathrm{i}\omega_0 2r_0/c}\,\mathrm{e}^{-\mathrm{i}\omega_0 2vt/c} \tag{2.24}$$

The Doppler Frequency

Consider that a wave's angular frequency is the rate of increase of its phase at a fixed position: $\omega = \partial\,\text{phase}/\partial t$. (Refer to (7.6) ahead for more analysis of this last expression.) So, from (2.24), the angular frequency of the returned wave measured at the receiver is $\omega_r = \mathrm{d}/\mathrm{d}t\,(\omega_0 t - \omega_0 2r_0/c - \omega_0 2vt/c)$. When the radial target velocity v is constant (which it is to a high approximation over the course of the radar interaction[6]), the received frequency is

$$\omega_r \simeq \omega_0 - \omega_0 2v/c\,. \tag{2.25}$$

The (angular) Doppler frequency ω_D is defined to be the increase in carrier frequency from transmission to reception:

$$\omega_D \equiv \omega_r - \omega_0 \simeq -\omega_0\,2v/c\,. \tag{2.26}$$

The usual (i.e. non-angular) Doppler frequency follows from (2.26):

$$f_D = \frac{\omega_D}{2\pi} = -f_0\,2v/c\,. \tag{2.27}$$

This is often written as $f_D = -2v/\lambda_0$, with λ_0 the wavelength corresponding to the carrier frequency f_0. All of the expressions above assume a narrow-band modulation $u(t)$ that, to a close approximation, doesn't appreciably alter the carrier frequency. This is certainly the case with radar signals.

[6] When v is not constant, (2.26) acquires a term $-\omega_0\,2\dot{v}t/c$, which will generally be very small.

Calculating the Amplitude $A(t)$

We find the amplitude A by asking how the amplitude of a radar wave relates to its transmitted power. The standard quantity in electromagnetic theory is the *Poynting vector* $\boldsymbol{S}$, which has the direction of energy flow of the wave, and whose magnitude is the wave's areal power density, being its power per unit area normal to that direction:

$$\boldsymbol{S} = \varepsilon_0 c^2 \boldsymbol{E} \times \boldsymbol{B}\,, \tag{2.28}$$

where ε_0 is the permittivity of the vacuum (we treat air as approximately a vacuum as far as radar propagation is concerned), and c is the speed of light in a vacuum. The electric and magnetic fields of a vacuum electromagnetic wave are always perpendicular with strengths related by $E = cB$, so it follows that the magnitude of the Poynting vector is $S = \varepsilon_0 c\, E^2$. The value of $1/(\varepsilon_0 c)$ is approximately 377 Ω, the famous "impedance of the vacuum".

Far from the source, the radiated electromagnetic field can be visualised as a sinusoid of amplitude A, so its electric-field strength is $E = A \sin(\omega_0 t + \phi)$. We see that the areal power density $S = \varepsilon_0 c\, E^2$ fluctuates sinusoidally over one period of the carrier wave; this is too rapid to be measured, but it's sufficient to consider the *average* areal power density $\langle S \rangle$ over a period of the wave:

$$\langle S \rangle = \varepsilon_0 c\, \langle E^2 \rangle = \varepsilon_0 c\, \langle A^2 \sin^2(\omega_0 t + \phi) \rangle = \varepsilon_0 c\, A^2/2\,. \tag{2.29}$$

Clearly, a similar expression holds if the wave is represented by a complex exponential with complex amplitude A: just replace A^2 by $|A|^2$. If $u(t)$ has value 0 when the radar is not emitting a pulse and modulus 1 when the radar *is* emitting a pulse, we can calculate A from (2.29). First consider one-way propagation. Here subscript T denotes target, t denotes transmitted, r denotes received by the radar, and we require A_T, the value of A immediately before the wave strikes the target.

$$\begin{aligned}
|A_T|^2 &= \frac{2}{\varepsilon_0 c} \times \text{average areal power density received } \textbf{by} \text{ target at } r \\
&= \frac{2}{\varepsilon_0 c} \times \frac{P_T}{\text{target area}} = \frac{2}{\varepsilon_0 c}\, P_T\, \frac{4\pi}{G_T \lambda^2} = \frac{2}{\varepsilon_0 c}\, \underbrace{\frac{P_t G_t \lambda^2}{(4\pi r)^2}\, G_T}_{\substack{P_T \text{ from one-way} \\ \text{radar equation}}}\, \frac{4\pi}{G_T \lambda^2} \\
&= \frac{P_t G_t}{2\pi \varepsilon_0 c\, r^2}\,,
\end{aligned} \tag{2.30}$$

so that

$$A_T = \frac{1}{r} \sqrt{\frac{P_t G_t}{2\pi \varepsilon_0 c}}\; \mathrm{e}^{\mathrm{i}\phi_0} \tag{2.31}$$

for some phase ϕ_0.

For two-way propagation, the value A_r of A for the signal received after being bounced from the target is not simply given by (2.31) with r replaced by $2r$, because the interaction with the target complicates the picture:

$$\begin{aligned}
|A_r|^2 &= \frac{2}{\varepsilon_0 c} \times \text{average areal power density received } \textbf{from} \text{ target at } r \\
&= \frac{2}{\varepsilon_0 c} \times \frac{P_r}{\text{receiver area}} = \frac{2}{\varepsilon_0 c}\, P_r\, \frac{4\pi}{G_t \lambda^2} = \frac{2}{\varepsilon_0 c}\, \underbrace{\frac{P_t G_t^2 \sigma \lambda^2}{(4\pi)^3\, r^4}}_{\substack{P_r \text{ from two-way} \\ \text{radar equation}}}\, \frac{4\pi}{G_t \lambda^2} \\
&= \frac{2 P_t G_t \sigma}{\varepsilon_0 c\, (4\pi)^2\, r^4}\,,
\end{aligned} \tag{2.32}$$

so that

$$A_r = \frac{1}{4\pi r^2}\sqrt{\frac{2P_tG_t\sigma}{\varepsilon_0 c}}\,\mathrm{e}^{\mathrm{i}\phi_0}\,, \tag{2.33}$$

where this ϕ_0 is in general different to the ϕ_0 in (2.31). We are interested in two-way propagation, so focus on (2.33). Equations (2.23) and (2.24) become

Received signal when used for modelling:

$$g_c(t) \simeq \frac{1}{4\pi r_0^2}\sqrt{\frac{2P_tG_t\sigma}{\varepsilon_0 c}}\,\mathrm{e}^{\mathrm{i}\phi_0}\,u(\underbrace{t-2r_0/c}_{\text{time of emission}})\,\mathrm{e}^{\mathrm{i}\omega_0 t}\,\mathrm{e}^{-\mathrm{i}\omega_0 2/c\,\times\,\text{range at emission}} \tag{2.34}$$

Received signal when used for analysis:

$$g_c(t) \simeq \frac{1}{4\pi r_0^2}\sqrt{\frac{2P_tG_t\sigma}{\varepsilon_0 c}}\,\mathrm{e}^{\mathrm{i}\phi_0}\,u(t-2r_0/c)\,\mathrm{e}^{\mathrm{i}\omega_0 t}\,\mathrm{e}^{-\mathrm{i}\omega_0 2r_0/c}\,\mathrm{e}^{\mathrm{i}\omega_D t} \tag{2.35}$$

As expected, the power in the received signal (which is proportional to its amplitude squared) is proportional to the fourth power of target distance, the effective radiated power P_tG_t, and the target cross section σ.

The determiner of range in (2.35) is the time delay $t_d = 2r_0/c$ from transmission to reception. Later we'll discuss the correlation procedure that extracts this time delay, and hence the range r_0. Equation (2.35) also includes a constant-phase term of $\mathrm{e}^{-\mathrm{i}\omega_0 2r_0/c}$, and of course the unknown phase $\mathrm{e}^{\mathrm{i}\phi_0}$ introduced when the signal interacted with the target. It doesn't matter that the radar receiver cannot be aware of these additional phases, because any constant-phase term in (2.35) won't affect the range–Doppler calculations in Section 4.2 and beyond.

3 Modelling the Returned Signal

For the sake of visualisation, we'll assume the radar illuminates a ship that is represented by a large number of known scatterers. So we require to model the signal returned from these scatterers, given some known emitted signal. If the emitted and received signals are represented by sampled I/Q data, two ways to model the received signal are as follows.

The high-fidelity way: We can construct the received signal at all sampled times, by back-tracing the rays that enter the receiver to their sources, one at each scatterer. This method works perfectly well but is very slow, because it requires the emitted signal to be calculated many times. For example, if there are 1000 scatterers and 500 sampling times at which we require the received signal, then at each sampling time, we must trace the signal via 1000 scatterers back to the emitter, thus calculating 1000 values of the emitted signal for each of those 500 times, making 500,000 calculations of the emitted signal. This approach is prohibitively CPU-expensive.

The slightly lower-fidelity way: This approach makes use of linear time-shift-invariant signal processing theory. It only requires the returned signal to be calculated corresponding to a ping signal emitted at time zero—meaning that a total of only 1000

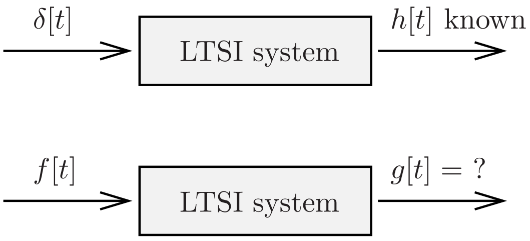


Figure 2: *A "black box" that illustrates the approach of calculating the response of an LTSI system.* ***Top:*** *given an impulse $\delta[t]$, we know the response $h[t]$.* ***Bottom:*** *given a signal $f[t]$, what is the response $g[t]$?*

calculations are needed for the above 1000 scatterers. A further processing gain can also be achieved, as some of the calculations depend only on the emitted signal and so can be made off line. We'll use this approach in this report and will describe it in detail now.

Linear time-shift-invariant theory calculates the output of a "linear time-shift-invariant (LTSI) system" given some input, subject to two assumptions:

– the system is *linear*: if we e.g. triple the strength of the input signal, the strength of the output signal will be tripled; and if two signals are input simultaneously, the output will be the sum of the individual outputs from those signals;

– the system is *time-shift invariant*: whether a signal is input today or tomorrow doesn't affect the output signal. Clearly this only applies to a moving ship over short time intervals.

The radar interaction is certainly linear. As for time-shift invariance, how small must the relevant time interval be for the invariance to apply? We saw in Figure 1 and (2.19) that an emitted signal will return after a time interval $t_d \simeq 2(r_0 + vt)/c$. Only when $vt \ll r_0$ will there be no "stretching" of flight times of successive signals. This inequality holds for the typical example considered in Section 2.3 so, for our scenarios, the time-shift invariance holds well.

The central result of LTSI theory is that the signal output from an LTSI system equals the *convolution* of the signal input with the system's *impulse response*. The system's impulse response is its output when the input is an impulse: a "spike" in the continuous case, or a "1" in the discrete case. We prove this central result while referring to Figure 2. Consider a signal that has been sampled at times t to give numbers $f[t]$ (the brackets denote the discreteness of f). It enters the LTSI "black box" and is transformed into an output g, sampled as $g[t]$, where we denote the transformation as $g = L(f)$. Given $f[t]$, what is $g[t]$? We know the system's response $h[t]$ when a single "1" is input to it at time zero. This signal that is "1" at time zero and "0" at all other times is denoted $\delta[t]$;

that is, $\delta[0] \equiv 1$ and $\delta[t \neq 0] \equiv 0$. So we know that $h[t] = L(\delta[t])$. Now:

$$\begin{aligned} g[t] &= L(f[t]) = L\Big(\sum_{t'} f[t']\,\delta[t-t']\Big) \\ &\overset{\text{linearity}}{=\!=\!=} \sum_{t'} f[t']\,L(\delta[t-t']) \underset{\text{invariance}}{\overset{\text{time-shift}}{=\!=\!=}} \sum_{t'} f[t']\,h[t-t'] \\ &\equiv (f*h)\,[t]\,, \end{aligned} \tag{3.1}$$

where $f * h$ is the *convolution* of f and h:

$$(f*h)\,[t] \equiv \sum_{t'} f[t']\,h[t-t']\,. \tag{3.2}$$

The key equation of LTSI theory is $g = f * h$, or

$$\boxed{\text{output signal} = \text{input signal} * \text{impulse response}\,.} \tag{3.3}$$

Appendix B has a tutorial on convolution; I think that practice with convolving two arrays from first principles using pen and paper is invaluable to understanding the procedure.

3.1 Calculating the Ship's Impulse Response

To calculate the elements of the ship's impulse response $h[t]$, realise that they are the returns of an emitted pulse that is a single "1". That is, emit a single "1" at time zero and place the returns from all the scatterers (in temporal order of course) into an array $\boldsymbol{h}$, which becomes the impulse response for use in later calculations. Of course, we don't assume that we know where the ship is, and so we should calculate the impulse response of a region of interest: everything that lies between some range R_{near} to some range R_{far}. This could include sea clutter, but in the following discussion I'll assume without loss of generality that only the ship gives a return.

The impulse response $\boldsymbol{h}$ of the "system" (ship plus environment) is supposed to be everything returned from the moment that we emit the "1" impulse. That is, we emit an impulse and then immediately listen to hear what the world throws back at us. We can begin to create this impulse response by initialising an array of zeroes, some of which will eventually be replaced by returns. How many zeroes are in this array? As shown in Figure 3 on the following page, the array is initialised with zeroes that are place holders for samples taken at time intervals of t_s. In principle we then calculate the return of each scatterer in the world, and add that return to the nearest element in time of the impulse response array $\boldsymbol{h}$. But in practice we need only calculate the returns of the scatterers in the region of interest. This means that the red zeroes in the figure don't interest us, and for efficient memory management we can ignore them, and account for their absence simply by adding R_{near} to all ranges that are calculated from the impulse response that was initialised with the *black* zeroes in Figure 3.

So given a sampling interval of t_s, the impulse response is initialised to an array of M zeroes, such that $(M-1)t_s$ is the time interval between the return from R_{near} to the return from R_{far}:

$$(M-1)t_s \simeq 2R_{\text{far}}/c - 2R_{\text{near}}/c\,. \tag{3.4}$$

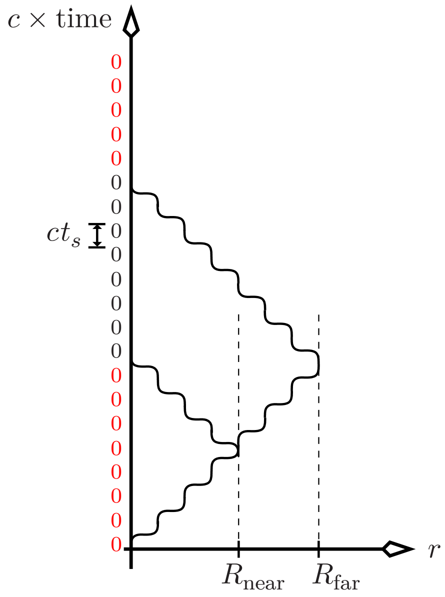


Figure 3: *The idea behind calculating the ship's impulse response: create an array of zeroes, then increment its relevant elements by the returns from a "1" sent out that bounces from each scatterer.*

In practice we must ensure that M is a whole number:

$$M - 1 = \text{round}\left[\frac{2(R_{\text{far}} - R_{\text{near}})}{ct_s}\right]. \tag{3.5}$$

Now bounce a signal "1" off each scatterer and find the time when it returns, adding the returned signal to the element of $\boldsymbol{h}$ whose index is the nearest one that corresponds to that time of return. For scatterer n at range r_n the nearest index will be k_n, where

$$(k_n - 1)t_s \simeq 2r_n/c - 2R_{\text{near}}/c\,, \tag{3.6}$$

again with some rounding:

$$k_n - 1 = \text{round}\left[\frac{2(r_n - R_{\text{near}})}{ct_s}\right]. \tag{3.7}$$

The return of the impulse "1" is the return of a signal with modulation 1 emitted at time zero. Now use (2.34) to write

$$\text{modulation of return } = \frac{1}{4\pi r_0^2}\sqrt{\frac{2P_tG_t\sigma}{\varepsilon_0 c}}\; e^{i\phi_0}\, e^{-i\omega_0 2/c \,\times\, \text{range at emission}}. \tag{3.8}$$

Calculate this quantity for each scatterer; e.g., calculate it for scatterer n and add this to element k_n of the array $\boldsymbol{h}$, found from (3.7). This modulation must be *added* to element k_n because it might well be that two closely spaced scatterers (in range) both contribute to this element.

4 The Correlation Procedure

When our radar emits a signal that reflects from a stationary target and returns to us in a degraded form, we can correlate the emitted and received signals to calculate the round-trip time. Each signal is sampled and digitised. Write the sampled emitted signal (carrier removed) as an array $\boldsymbol{E}$ of complex numbers, of length e. Likewise, the received signal (sampled at the same rate and with carrier removed) is an array $\boldsymbol{R}$ of length r. Further, denote the complex conjugate of a number z by z^*, and introduce the idea of the complex conjugate of a sequence and its reverse. For example, a 3-sample array $\boldsymbol{E} = [E_1 \;\; E_2 \;\; E_3]$ has

$$\boldsymbol{E}^* \equiv [E_1^* \;\; E_2^* \;\; E_3^*]\,, \quad \text{and} \quad \boldsymbol{E}^\dagger \equiv [E_3^* \;\; E_2^* \;\; E_1^*]\,. \tag{4.1}$$

Write the correlation of $\boldsymbol{E}$ with $\boldsymbol{R}$ as a new sequence $\boldsymbol{E} \star \boldsymbol{R}$.

Calculating $\boldsymbol{E} \star \boldsymbol{R}$ is visualised in (4.2) below. Write the elements of $\boldsymbol{E}^*$ as a length-e movable row above a stationary row of the r elements of $\boldsymbol{R}$, such that the last-received (rightmost) element of $\boldsymbol{E}^*$ lies above the first element of $\boldsymbol{R}$, and then multiply these two elements; this product $E_e^* R_1$ is the first element of $\boldsymbol{E} \star \boldsymbol{R}$. To calculate the next element of $\boldsymbol{E} \star \boldsymbol{R}$, shift $\boldsymbol{E}^*$ one element to the right and again multiply each element of $\boldsymbol{E}^*$ with the element of $\boldsymbol{R}$ (if any) immediately below it. There will now be two such products, and they are added to give the second element of $\boldsymbol{E} \star \boldsymbol{R}$. Continue this procedure until there are no more products to be calculated. In other words, after each shift of $\boldsymbol{E}^*$ one element to the right, we form the dot product of the overlapping subsequences:

$$\begin{array}{ll}
\boldsymbol{E}^* = [E_1^* \quad E_2^* \quad E_3^* \;\ldots\; E_e^*] \longrightarrow \text{moving to right} & \\
\boldsymbol{R} \;\; = \qquad\qquad\qquad\quad [\,\underbrace{R_1 \quad R_2 \quad R_3 \;\ldots\; R_r}_{\text{received signal}}\,] & \\
\hline
\end{array}$$

$$\boldsymbol{E} \star \boldsymbol{R} \;= [\,E_e^* R_1, \quad E_{e-1}^* R_1 + E_e^* R_2, \quad E_{e-2}^* R_1 + E_{e-1}^* R_2 + E_e^* R_3, \;\ldots,\; E_1^* R_r] \tag{4.2}$$

[The correlation process is sometimes defined using $\boldsymbol{E}$ and $\boldsymbol{R}^*$, which differs only by a complex conjugate from (4.2) and so is just as valid to use.] This procedure will *always* produce a peak in the absolute value of the correlation when a sequence, on being correlated with itself, matches up with itself—which is the whole point of correlating two sequences: to search for one within the other. (The proof is omitted but straightforward, and uses the Cauchy-Schwarz theorem for complex sequences.) This peak would not be guaranteed to occur—and in practice will seldom occur—if the complex conjugate were not used in the correlation procedure.

Correlation "$\star$" turns out to be almost the same process as *convolution* "$*$" (not to be confused with a complex conjugate here, which uses this symbol as a superscript):

$$\boldsymbol{A} \star \boldsymbol{B} = \boldsymbol{A}^\dagger * \boldsymbol{B}\,, \tag{4.3}$$

where "†" was defined in (4.1). This is essential to note, because *convolution* can be carried out efficiently using a fast Fourier transform, implying that *correlation* can be implemented in the same efficient way. Of great use is the fact that convolution is commutative: $\boldsymbol{A} * \boldsymbol{B} = \boldsymbol{B} * \boldsymbol{A}$, and associative: $(\boldsymbol{A} * \boldsymbol{B}) * \boldsymbol{C} = \boldsymbol{A} * (\boldsymbol{B} * \boldsymbol{C})$. Correlation is neither of these; specifically,

$$\boldsymbol{A} \star \boldsymbol{B} = (\boldsymbol{B} \star \boldsymbol{A})^\dagger, \quad (\boldsymbol{A} \star \boldsymbol{B}) \star \boldsymbol{C} = \boldsymbol{A}^\dagger \star (\boldsymbol{B} \star \boldsymbol{C}). \tag{4.4}$$

Appendix B gives some further comments on how correlation is related to convolution.

4.1 Relating Correlation to Distance and Velocity

Each of the above "correlation steps" (shifting $\boldsymbol{E}^*$ to the right by one sample) corresponds to testing for a new posited time interval between emission and reception; this time interval corresponds to a posited range to the target. If we plot the absolute value of the correlation versus this range, then the range value at which the graph peaks is where we presume the target to be.

The target may well be moving, in which case we cannot expect a good correlation of emitted and returned signals. Instead, we must allow that $\boldsymbol{R}$ might be Doppler shifted from $\boldsymbol{E}$. This might be arranged by performing the above correlation process many times; each time, we subtract some amount of posited Doppler frequency from $\boldsymbol{R}$. What results is many plots of correlation output versus range: one plot for each of the posited Doppler frequencies. These plots can be stacked to form a surface plot of correlation output (or rather its absolute value, since it's generally complex) versus range and Doppler frequency. The properties of this graph, the *ambiguity function* for the radar signal used, have been studied extensively, and it forms the standard approach to determining how well any newly designed signal might perform in practice. We might hope to find a sharp spike in the function at the correct values of range and Doppler frequency. In fact, one of the various theorems of the function's properties discussed in [2] states that it's impossible to craft a signal that will produce such a spike. Appendix G shows how the ambiguity function is created in practice.

The ambiguity function is built by stacking many correlation–range plots, each for a particular value of Doppler frequency. Historically, radar receivers have not calculated target range and velocity in this way, perhaps because of the large amount of computer processing required. Instead they build a *range–Doppler* plot by stacking many *frequency spectra* of correlation data that have been obtained pulse by pulse from a *coherent pulse train*, with each spectrum calculated for a particular value of *range*. The details of this procedure are given in Section 5.

Summarising, the correlation of emitted and received signals is traditionally done in either of two ways:

The ambiguity function: For each value of Doppler frequency in an appropriate domain, plot the correlation of part or all of the signal sent out versus range, and stack the graphs across frequency. This approach is used off-line for analysing radar waveforms.

The range–Doppler plot: For each value of range in an appropriate domain, plot the frequency spectrum of the correlations produced from a series of pulses in a coherent pulse train, and stack the graphs across range. Radar hardware does this.

The process of correlating the returned signal with the emitted signal is known as applying a *matched filter*, whose name perhaps derives from something like the following argument. The primary process here is that of *correlating* the emitted and returned signals to locate the emitted signal within the returned signal. In principle, this correlation could be done in an infinite number of different ways; the question is, which of these ways can best detect a signal in the presence of noise? Determining this correlation procedure has traditionally followed a linear time-shift-invariant approach (mentioned at the start of Section 2), perhaps because that approach was historically the easiest method to build into analogue hardware. If the noise in the returned signal is assumed to be "white" (meaning temporally uncorrelated), then it's not difficult to show [3] that the best way to correlate emitted and returned signals using a linear time-shift-invariant approach is via a *convolution*. Convolution turns out to be a type of *moving mean*, and a moving mean removes ("filters") signal components of certain frequencies. So in this sense the correlation process can be viewed as a type of filtering process, whose filter stencil "matches" the signal. Although the standard correlation procedure is often simply introduced as a convolution, it's useful to remember that convolution is just one choice of how correlation might be implemented. But convolution is the standard choice, because it turns out to maximise the ratio of what results from filtering the signal, compared to what results from filtering the ever-present noise.

4.2 Calculating the Target Range

As on page 17, sample the modulation of the emitted signal into a length-e array $\boldsymbol{E}$ of complex numbers. Likewise, sample the modulation of the received signal at the same rate into the complex array $\boldsymbol{R}$ of length r. The sampling time interval is t_s. We will correlate $\boldsymbol{E}$ with $\boldsymbol{R}$, using (4.3) to produce an array of correlation numbers via a convolution. Calculating the target range requires knowledge only of the absolute value of each of these numbers, so call the array of these absolute values $\boldsymbol{C}$:

$$\boldsymbol{C} = |\boldsymbol{E} \star \boldsymbol{R}| = |\boldsymbol{E}^\dagger * \boldsymbol{R}| \,, \tag{4.5}$$

which is accomplished by the Matlab command

```
C = abs( conv(conj(fliplr(E)), R) );
```

The array $\boldsymbol{C}$ has length $e + r - 1$, as can be shown by examining what happens when $\boldsymbol{E}$ (length e) is correlated with $\boldsymbol{R}$ (length r), with reference to (4.6) below. First, there are r correlation operations (producing the first r elements of $\boldsymbol{C}$) as the right-most element of $\boldsymbol{E}^*$ starts at the left-most element of $\boldsymbol{R}$ and shifts element-by-element to eventually "hit" the right-most element of $\boldsymbol{R}$. As $\boldsymbol{E}^*$ continues moving to the right, $e - 1$ more correlation operations occur until $\boldsymbol{E}^*$ eventually "falls off the right-hand end" of $\boldsymbol{R}$.

One of the elements of $\boldsymbol{C}$ will be a maximum, defining the best correlation of $\boldsymbol{E}$ and $\boldsymbol{R}$. What time delay between emitted and received signals does this or any other element of $\boldsymbol{C}$ correspond to? For the sake of the numerical argument below, suppose that the sampled received signal contains the emitted signal somewhere within it. That is, suppose that $\boldsymbol{R}$ is composed of n noise numbers (small numbers that won't correlate well with $\boldsymbol{E}$, which we replace by zeroes for the purpose of this discussion) followed by $\boldsymbol{E}$ (which might have some added noise, but that's immaterial to this discussion so is omitted), followed by more noise numbers:

$$\begin{aligned} \boldsymbol{E}^* &= [E_1^* \;\; E_2^* \;\; E_3^* \;\; \dots \;\; E_e^*] \longrightarrow \text{ moving to right} \\ \boldsymbol{R} &= \qquad\qquad [\underbrace{0\;\,0\;\,0\,\dots\,0}_{n \text{ noise numbers}} \;\; \underbrace{E_1\;\,E_2\;\,E_3\,\dots\,E_e}_{\text{emitted signal}} \;\; \underbrace{0\;\,0\;\,0\,\dots\,0}_{\text{more noise numbers}}]. \end{aligned} \tag{4.6}$$

The peak correlation occurs when $\boldsymbol{E}^*$ has slid in from the left and now is exactly alongside (above) the copy of $\boldsymbol{E}$ that is inside $\boldsymbol{R}$, element for element. The number of correlation events that this corresponds to is the sum of two numbers: n correlations with the noise numbers, and then e correlations to line up alongside $\boldsymbol{E}$. So the peak occurs at element number $n+e$ of the correlation array $\boldsymbol{C}$ (whose elements are numbered 1 to $e+r-1$). But we know that the time interval t_d between emission and reception of the signal is composed of[7] n lots of t_s, because there are $n-1$ inter-element occurrences of t_s in the noise signal, plus one extra occurrence of t_s to reach element E_1:

$$\underbrace{\underbrace{0 \overset{t_s}{\longleftrightarrow} 0 \overset{t_s}{\longleftrightarrow} 0 \ldots 0}_{\substack{n \text{ noise numbers, so}\\ n-1 \text{ lots of } t_s}} \underbrace{\overset{t_s}{\longleftrightarrow} E_1}_{1 \text{ lot of } t_s}}_{t_d = nt_s} \ldots \tag{4.7}$$

Because the index of the element of $\boldsymbol{C}$ that corresponds to maximum correlation is $n+e$, we conclude that

$$t_d = nt_s = \big([\text{index of peak correlation of } \boldsymbol{C}\,] - e\big)\, t_s\,. \tag{4.8}$$

(This equation refers to the *index* of the element of $\boldsymbol{C}$ that gives the peak correlation, not the element itself.) The time interval t_d corresponds to a target range of $ct_d/2$, so the range inferred from the above correlation is

$$\boxed{\text{target range} = \big([\text{index of peak correlation of } \boldsymbol{C}\,] - e\big)\, \frac{ct_s}{2}\,.} \tag{4.9}$$

As a check, (4.9) says that if the correlation elements of $\boldsymbol{C}$ peak at index e, the target range must be zero. This makes sense because it corresponds to the absence of the first n noise numbers in (4.6): the signal is being returned as soon as it's emitted, corresponding to a target at zero distance from the radar.

Equation (4.9) computes the target range from the peak correlation index of $\boldsymbol{C}$; but we can just as well apply it to *every* index of $\boldsymbol{C}$ to give the computed range had that index denoted the maximum element of $\boldsymbol{C}$. (In particular, increasing the correlation index by one results in a computed range increase of $ct_s/2$.) This idea allows us to rescale the time-delay axis to range, which is a more useful parameter than time delay for analysing a real radar return.

Incorporating the Target's Impulse Response $\boldsymbol{h}$

Equation (3.3) says that $\boldsymbol{R} = \boldsymbol{E} * \boldsymbol{h}$. The correlation of emitted and received signals is then

$$\boldsymbol{E} \star \boldsymbol{R} \overset{(4.3)}{=\!=\!=} \boldsymbol{E}^\dagger * \boldsymbol{R} \;=\; \boldsymbol{E}^\dagger * (\boldsymbol{E} * \boldsymbol{h}) \;=\; (\boldsymbol{E}^\dagger * \boldsymbol{E}) * \boldsymbol{h}\,. \tag{4.10}$$

This is the central equation that we'll use to correlate emitted and returned signals:

$$\boxed{\text{Correlation } = \boldsymbol{E} \star \boldsymbol{R} = (\boldsymbol{E}^\dagger * \boldsymbol{E}) * \boldsymbol{h}\,.} \tag{4.11}$$

[7] This calculation is "coarse" by an amount set by t_s since we are treating $n = t_d/t_s$ as a whole number, even though it will not be in practice. But when t_s is small, the range error in the above analysis is negligible.

The array of emitted I/Q data $\boldsymbol{E}$ doesn't change from pulse to pulse, so $\boldsymbol{E}^\dagger * \boldsymbol{E}$ can be pre-computed, and then convolved at run time with the ship's impulse response $\boldsymbol{h}$, which is updated at the start of each new pulse. The absolute values of the result form the array $\boldsymbol{C}$ in (4.5) that will return the ship's distance via (4.9). Also, the correlation arrays $\boldsymbol{E}_1 \star \boldsymbol{R}_1,\ \boldsymbol{E}_2 \star \boldsymbol{R}_2, \dots$ from pulses $1, 2, \dots$ form successive rows of the matrix M discussed ahead on page 22, whose columns turn out to hold Doppler information.

Note that in general, prepending or appending zeroes to one or both of any arrays $\boldsymbol{A}$ and $\boldsymbol{B}$ has the effect of prepending or appending those zeroes to $\boldsymbol{A} * \boldsymbol{B}$. This fact is worth remembering whenever an array begins with a set of zeroes that we wish to exclude for reasons of efficient calculation. This idea was actually implicit in the discussion on page 15 when we spoke of adding the offset R_{near} to all ranges calculated using an $\boldsymbol{h}$ that had no unnecessary leading zeroes.

So much for extracting range information from the returned signal. We'll now show how to extract Doppler information from the correlations.

5 Creating a Range–Doppler Plot

A radar receiver generates a range–Doppler plot by emitting a set of N coherent pulses, then correlating each pulse with that pulse's return. (It doesn't correlate the entire pulse train it emits with the entire train returned; cf. the ambiguity function in Appendix G.) We'll see shortly that the correlations over successive pulses contain Doppler information, which can be extracted for each range cell from a discrete Fourier transform (DFT) of the correlations in that range cell.

For simplicity, suppose we have a single point target, and consider the first pulse $\boldsymbol{E}_1$ emitted in the coherent pulse train. Its modulation $u(t)$ is sampled at intervals of t_s and digitised into a sequence of, say, 4 complex numbers:

$$\boldsymbol{E}_1 = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}. \tag{5.1}$$

The return $\boldsymbol{R}_1$ of the first pulse is received beginning at some time t_0. Referring to (2.35) for one scatterer, write $\boldsymbol{R}_1$ as some modulation times the sampled Doppler shift:

$$\begin{aligned} \boldsymbol{R}_1 &= \left[\mathrm{e}^{\mathrm{i}\omega_D t_0}\, u_a\,, \quad \mathrm{e}^{\mathrm{i}\omega_D (t_0+t_s)}\, u_b\,, \quad \mathrm{e}^{\mathrm{i}\omega_D (t_0+2t_s)}\, u_c\,, \quad \mathrm{e}^{\mathrm{i}\omega_D (t_0+3t_s)}\, u_d \right] \\ &= \mathrm{e}^{\mathrm{i}\omega_D t_0} \left[u_a\,, \quad \mathrm{e}^{\mathrm{i}\omega_D t_s}\, u_b\,, \quad \mathrm{e}^{\mathrm{i}\omega_D 2t_s}\, u_c\,, \quad \mathrm{e}^{\mathrm{i}\omega_D 3t_s}\, u_d \right], \end{aligned} \tag{5.2}$$

where u_a, u_b, u_c, u_d are values of the signal emitted at earlier times that incorporate the other phase shifts in (2.35). For simplicity in what follows, write the constants in these expressions as

$$\alpha \equiv \mathrm{e}^{\mathrm{i}\omega_D t_0}\,, \quad \beta \equiv \mathrm{e}^{\mathrm{i}\omega_D t_s}, \tag{5.3}$$

so that

$$\boldsymbol{E}_1 = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}, \quad \boldsymbol{R}_1 = \alpha \begin{bmatrix} u_a & \beta u_b & \beta^2 u_c & \beta^3 u_d \end{bmatrix}. \tag{5.4}$$

We might expect the next pulse $\boldsymbol{E}_2$ emitted in the coherent pulse train to be identical to $\boldsymbol{E}_1$, with the target still returning the modulation u_a, u_b, u_c, u_d since it moves negligibly in the pulse repetition interval T from one pulse to the next. Or there might be some known phase shift from one emitted pulse to the next, as well as the fact that the reference oscillator used to remove the carrier signal is not completely stable and introduces some arbitrary phase to each emitted pulse. So call the total phase shift ϕ_n for the n^{th} pulse, with $\phi_1 \equiv 0$. This phase is preserved in the returned pulse:

$$\begin{aligned}
\boldsymbol{E}_2 &= \big[u_1 \ \ u_2 \ \ u_3 \ \ u_4\big]\,\mathrm{e}^{\mathrm{i}\phi_2}, \\
\boldsymbol{R}_2 &= \big[\mathrm{e}^{\mathrm{i}\omega_D(t_0+T)}\,u_a\,, \quad \mathrm{e}^{\mathrm{i}\omega_D(t_0+T+t_s)}\,u_b\,, \quad \mathrm{e}^{\mathrm{i}\omega_D(t_0+T+2t_s)}\,u_c\,, \quad \mathrm{e}^{\mathrm{i}\omega_D(t_0+T+3t_s)}u_d\big]\,\mathrm{e}^{\mathrm{i}\phi_2} \\
&= \alpha\mathrm{e}^{\mathrm{i}\omega_D T}\big[u_a \quad \beta u_b \quad \beta^2 u_c \quad \beta^3 u_d\big]\,\mathrm{e}^{\mathrm{i}\phi_2}. \qquad (5.5)
\end{aligned}$$

Similarly,

$$\begin{aligned}
\boldsymbol{E}_3 &= \big[u_1 \ \ u_2 \ \ u_3 \ \ u_4\big]\,\mathrm{e}^{\mathrm{i}\phi_3}, \\
\boldsymbol{R}_3 &= \big[\mathrm{e}^{\mathrm{i}\omega_D(t_0+2T)}\,u_a\,, \quad \mathrm{e}^{\mathrm{i}\omega_D(t_0+2T+t_s)}\,u_b\,, \quad \mathrm{e}^{\mathrm{i}\omega_D(t_0+2T+2t_s)}\,u_c\,, \quad \mathrm{e}^{\mathrm{i}\omega_D(t_0+2T+3t_s)}u_d\big]\,\mathrm{e}^{\mathrm{i}\phi_3} \\
&= \alpha\mathrm{e}^{\mathrm{i}\omega_D 2T}\big[u_a \quad \beta u_b \quad \beta^2 u_c \quad \beta^3 u_d\big]\,\mathrm{e}^{\mathrm{i}\phi_3}, \qquad (5.6)
\end{aligned}$$

and in general, for the n^{th} pulse in the coherent train,

$$\boldsymbol{E}_n = \big[u_1 \ \ u_2 \ \ u_3 \ \ u_4\big]\,\mathrm{e}^{\mathrm{i}\phi_n}\,, \quad \boldsymbol{R}_n = \alpha\mathrm{e}^{\mathrm{i}\omega_D(n-1)T}\big[u_a \quad \beta u_b \quad \beta^2 u_c \quad \beta^3 u_d\big]\,\mathrm{e}^{\mathrm{i}\phi_n}. \qquad (5.7)$$

Now correlate $\boldsymbol{E}_n$ with $\boldsymbol{R}_n$, noting that the conjugation used in the correlation process cancels the unknown phase ϕ_n:

$$\begin{aligned}
\boldsymbol{E}_n \star \boldsymbol{R}_n &= \boldsymbol{E}_n^\dagger * \boldsymbol{R}_n \\
&= \big[u_4^* \quad u_3^* \quad u_2^* \quad u_1^*\big]\,\cancel{\mathrm{e}^{-\mathrm{i}\phi_n}} * \alpha\mathrm{e}^{\mathrm{i}\omega_D(n-1)T}\big[u_a \quad \beta u_b \quad \beta^2 u_c \quad \beta^3 u_d\big]\,\cancel{\mathrm{e}^{\mathrm{i}\phi_n}} \\
&= \alpha\mathrm{e}^{\mathrm{i}\omega_D(n-1)T}\big[u_4^*\,u_a\,, \ \ u_4^*\,\beta u_b + u_3^*\,u_a\,, \ \ \ldots, \ \ u_1^*\,\beta^3 u_d\big]\,. \qquad (5.8)
\end{aligned}$$

Now make this sequence the n^{th} row of a matrix M: this matrix will have N rows because each row contains one of the N correlations performed during the *coherent processing interval* (CPI) comprised of N pulses. The process of correlating each of the N pulses with itself is illustrated in Figure 4. We correlate the first pulse sent out with its return, and set the resulting sequence of numbers to be the first row of the matrix M. The numbers in this row are the correlations represented by the blue dots in the top peaked "curve" of the figure. The peak corresponds to the best correlation, which gives the range to the target using the procedure of (4.9). But we won't actually use these correlation numbers directly to calculate this range, because after having done the correlations, it turns out that we'll Fourier-transform the matrix M. However, the resulting matrix will preserve the placement of the correlation peaks.

After having correlated the first emitted pulse with its return, we do the same with the second pulse emitted and its return, setting the resulting sequence of numbers to be the second row of the matrix M. This row's numbers are the blue dots in the second peaked curve of Figure 4. (The peak will lie at approximately the same range as the peak in the top row, because the target has barely moved in time T.) We do this for all N pulses—all of the blue dots in the figure—to build the entire correlation matrix M, row by row.

With M constructed, focus on any of its columns. Inspection of (5.8) shows that this

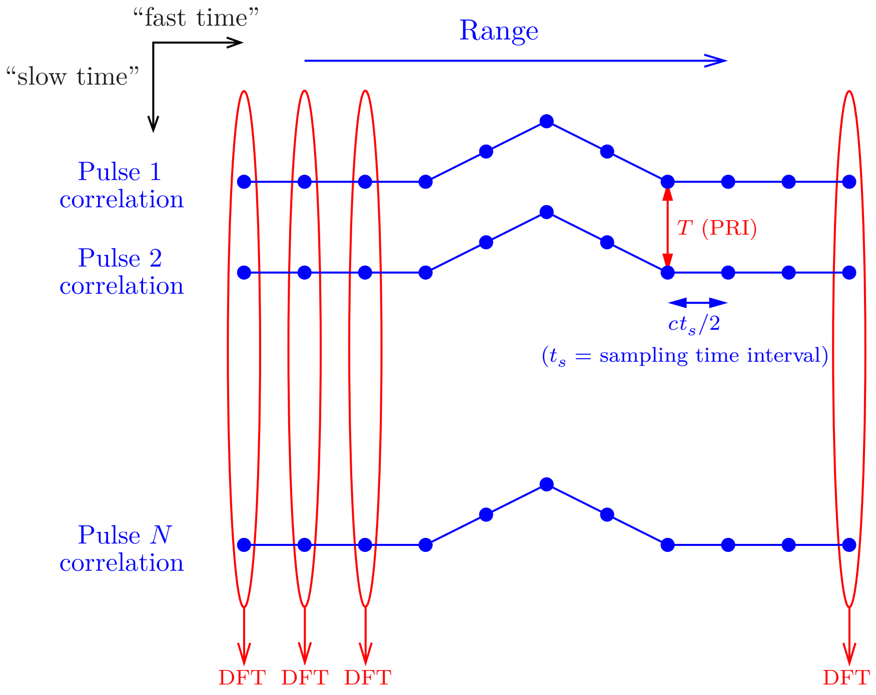


Figure 4: *The general scheme followed to build a range–Doppler plot via range slices*

column will be

$$\begin{bmatrix} 1 \\ e^{i\omega_D T} \\ e^{i\omega_D 2T} \\ \vdots \\ e^{i\omega_D (N-1)T} \end{bmatrix} \times \begin{pmatrix} \text{some constant that changes} \\ \text{from column to column} \end{pmatrix}. \tag{5.9}$$

The sequence of N numbers in this column equals an irrelevant constant times a sequence of N numbers that can be treated as samples of $e^{i\omega_D t}$ at times $t = 0, T, 2T, \ldots$ So if we calculate the frequency spectrum of the entire column, then—given that we're using a point target in this discussion—we will (ideally) find a single angular frequency ω_D to be present. This corresponds to the target recession velocity v via (2.26) or (2.27):

$$\boxed{\text{target recession velocity } v = \frac{-c\omega_D}{2\omega_0} = \frac{-cf_D}{2f_0}.} \tag{5.10}$$

The frequency spectrum of the column in (5.9) is found using a DFT, which produces an array of the amounts of each frequency present. These frequencies are equally spaced, running from approximately minus the Nyquist frequency to the Nyquist frequency for the sampled set; see the discussion in Section 6 for details of the exact bounds of the spectrum, specifically (6.4) and (6.5). The Nyquist frequency is $1/(2T)$, or half the pulse repetition frequency. This array of frequencies can be converted to radial velocities using (5.10). That is, each of these frequencies, if it were the actual Doppler frequency of the target, would correspond to a velocity given by (5.10), in which case we need only multiply the entire array of frequencies by $-c/(2f_0)$ to convert it from frequency to radial velocity. Ideally, a moving point target might be expected to produce a single sharp peak at its Doppler frequency, but in practice the sampling process spreads this peak out over many frequencies.

We now repeat the above DFT analysis for each column of M. Only the common factor outside the brackets in (5.9) changes from column to column but, being constant in time, it's not detected by the DFT and so doesn't affect the spectrum produced. Hence each column yields a frequency spectrum, and this column (with its frequency spectrum) corresponds to a particular range by way of the discussion following (4.9) on page 20. That discussion is worth repeating here: if we apply (4.9) to *every* correlation index—meaning each column number of M—then we'll obtain the range corresponding to that column number, and this converts the time axis to range. So the frequency spectrum for each column of M becomes a slice of a surface drawn over range and velocity axes, and this surface is the range–Doppler plot.

The same procedure applies to a target composed of multiple point scatterers. The signals returned from multiple point scatterers simply add, and the linearity of the convolution procedure ensures that each velocity present gives rise to its own Doppler peak.

6 Comments on Applying a Discrete Fourier Transform

Different schemes for ordering the elements of a DFT exist. For this report we define a length-N sequence $\{x_0, \dots, x_{N-1}\}$ to have a length-N DFT of $\{X_{-[N/2]}, X_{-[N/2]+1}, \dots\}$ (where, for positive a, the notation $[a]$ denotes the greatest integer $\leqslant a$). We use the convention of reference [4] with the choice $\alpha = 1$ there:

$$X_n = \frac{1}{N} \sum_{k=0}^{N-1} x_k \, \mathrm{e}^{-\mathrm{i}2\pi kn/N}, \quad n = -[N/2], -[N/2]+1, \ \dots, \ \left[\frac{N-1}{2}\right] \ (N \text{ values}). \tag{6.1}$$

The inverse transform too is a sum over N numbers:

$$x_k = \sum_{n=-[N/2]}^{\left[\frac{N-1}{2}\right]} X_n \, \mathrm{e}^{\mathrm{i}2\pi kn/N}, \quad 0 \leqslant k \leqslant N-1. \tag{6.2}$$

In practice for any data set of reasonable size, the DFT is implemented using an algorithm called a fast Fourier transform (FFT). Older implementations of the FFT required N to be a power of 2; if N wasn't a power of 2, zeroes were typically appended to the data set to increase its length to a power of 2. Of course, this zero padding changed the data, and so had the obvious side effect of introducing spurious frequencies to the spectrum. These unwanted frequencies were tolerated some years ago because only by zero padding to a power-of-2 length could the FFT be used at all. But current FFT algorithms are very fast even when N is not a power of 2, and so the absolute need for zero padding is now a thing of the past. (Appendix D has more discussion of this zero padding.) Note that this padding is distinct from the appending with zeroes that's required when using DFTs to convolve sequences of different lengths, discussed in Appendix B, and which has nothing to do with powers of 2. That procedure works *exactly* with the extra zeroes: it's simply not about building a spectrum or a choice of FFT algorithm.

Nonetheless, emitting a power-of-2 number of pulses in a coherent processing interval still seems to be common practice, even in digital radars.

The Matlab function `fft` orders its DFT amplitudes in a way that renders negative frequencies positive and greater than the Nyquist frequency. I don't see any point in adding twice the Nyquist frequency to a negative frequency purely to make it positive (as if negative frequencies are something to avoid), when that resulting positive frequency, being now greater than Nyquist, is invalid from a DFT point of view. Instead, I give an example of implementing the DFT in Matlab that leaves negative frequencies negative, using the code of Table C1 in Appendix C. Note that the arrays `x` and `X` in that table are arrays of numbers; that is, the first element of `X` is denoted `X(1)` in Matlab, even though it's called $X_{-[N/2]}$ in (6.1). Table C2 gives equivalent Mathematica code.

Here is an example of plotting a frequency spectrum using Table C1's `Dft` function in Matlab. Suppose the data has been sampled in time intervals of t_s into an array `data`. The DFT of `data` is the array

```
DFT_of_data = Dft(data);
```

Plot `DFT_of_data` versus frequency by first constructing an appropriate frequency domain, which will be a sequence of N numbers $\{-\hat{f}, -\hat{f}+\Delta f, -\hat{f}+2\Delta f, \dots\}$ for some $\hat{f} > 0$ and frequency increment Δf. The increment Δf is the reciprocal of the duration of the sampling interval. Note that this duration is *not* the time t_f of the last sample minus the time t_i of the first sample. Rather, the periodic nature of sinusoids means that the internals of the DFT really work on an infinite-length array made by repeating the data endlessly before and after the actual period of sampling. So the DFT effectively assumes the data begins anew at time $t_f + t_s$, and thus the sampling interval has duration $t_f + t_s - t_i$. If the data consists of N samples spaced t_s apart, this duration must equal Nt_s, so

$$\Delta f = \frac{1}{t_f + t_s - t_i} = \frac{1}{Nt_s}\,. \tag{6.3}$$

The value of $\hat{f}$ must be determined separately for each of the cases of N even or odd.

N **even:** For example, suppose $N = 4$ so that (6.1) gives the transform as $\{X_{-2}, X_{-1}, X_0, X_1\}$. It follows that in general, the frequency domain is $\{-\hat{f}, \dots, \hat{f} - \Delta f\}$, and this has extent $\hat{f} - \Delta f - (-\hat{f})$, which we know must equal $(N-1)\Delta f$. Hence

$$\hat{f} = \frac{N\Delta f}{2} \overset{(6.3)}{=\!=\!=} \frac{1}{2t_s} = \text{Nyquist frequency.} \tag{6.4}$$

N **odd:** Suppose $N = 5$, so that the transform is $\{X_{-2}, X_{-1}, X_0, X_1, X_2\}$. In general the frequency domain is $\{-\hat{f}, \dots, \hat{f}\}$ with extent $2\hat{f}$, which must also equal $(N-1)\Delta f$. Hence

$$\hat{f} = \frac{(N-1)\Delta f}{2} = \frac{N\Delta f}{2} - \frac{\Delta f}{2} = \text{Nyquist frequency} - \frac{\Delta f}{2}\,. \tag{6.5}$$

This can all be implemented in Matlab with

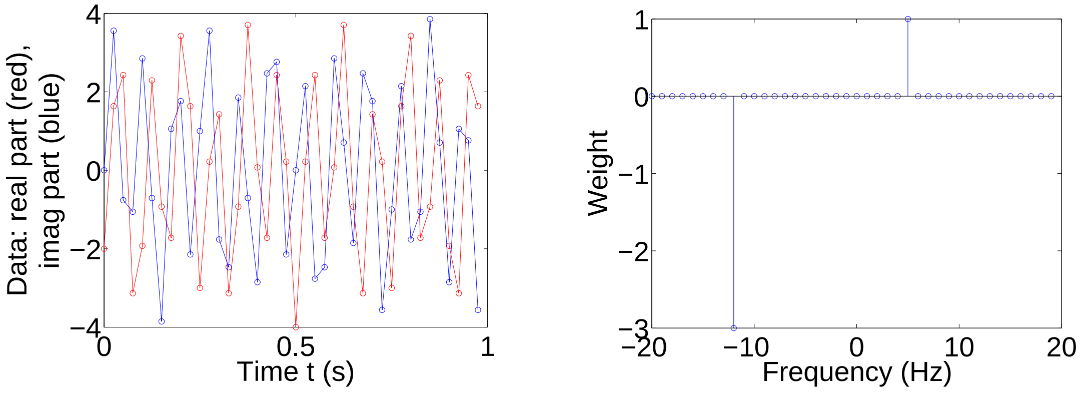


Figure 5: ***Left:*** *Data sampled at 40 Hz of two complex exponentials at frequencies 5 Hz and −12 Hz.* ***Right:*** *The DFT has peaks at the correct frequencies with the correct weights.*

```
Delta_f = samplingFrequency/N;
nyquistFrequency = samplingFrequency/2;

if abs(mod(N,2)) < 1e-10
  % DFT has even length N.
  f_hat = nyquistFrequency;
  frequencyDomain = linspace(-f_hat, f_hat-Delta_f, N);
else
  % DFT has odd length N.
  f_hat = nyquistFrequency - Delta_f/2;
  frequencyDomain = linspace(-f_hat, f_hat, N);
end
```

The plot of, say, the absolute value of the DFT is accomplished with

```
stem(frequencyDomain, abs(DFT_of_data))
```

As an example, suppose we sample the function $e^{i2\pi(5\,\mathrm{Hz})t} - 3\,e^{i2\pi(-12\,\mathrm{Hz})t}$ at intervals of 1/40 second. The Nyquist frequency is half the sampling rate, or 20 Hz, so the DFT will certainly detect the frequencies of 5 Hz and −12 Hz. Suppose now, that by chance we happened to choose the end sampling time to be one sampling interval (1/40 s) less than a period of time that contains a whole number of cycles of both frequencies present. One second is such a period of time (it contains exactly 5 cycles of one wave and 12 cycles of the other), so suppose we sample from time 0 to time $1 - {}^{1}\!/_{40}$ s, shown at left in Figure 5. This sampling choice guarantees that the DFT "sees" two infinitely long sinusoids, so that it will faithfully produce only peaks at the two frequencies 5 Hz and −12 Hz. This can indeed be seen in the frequency spectrum at the right in Figure 5, where we have plotted the actual DFT and not its absolute value, because in this case it's composed of real numbers anyway. The peaks are in the expected places with the correct weights.

Now suppose we sample the same function at intervals of 1/20 second, from 0 to $1 - {}^{1}\!/_{20}$ s, shown at left in Figure 6. The Nyquist frequency is now 10 Hz. This is high enough to capture the 5 Hz sinusoid (shown at the right in Figure 6), but not high enough to capture the one at −12 Hz. As a result, the DFT algorithm effectively shifts the −12 Hz peak up or down by multiples of the frequency domain width (which is twice the

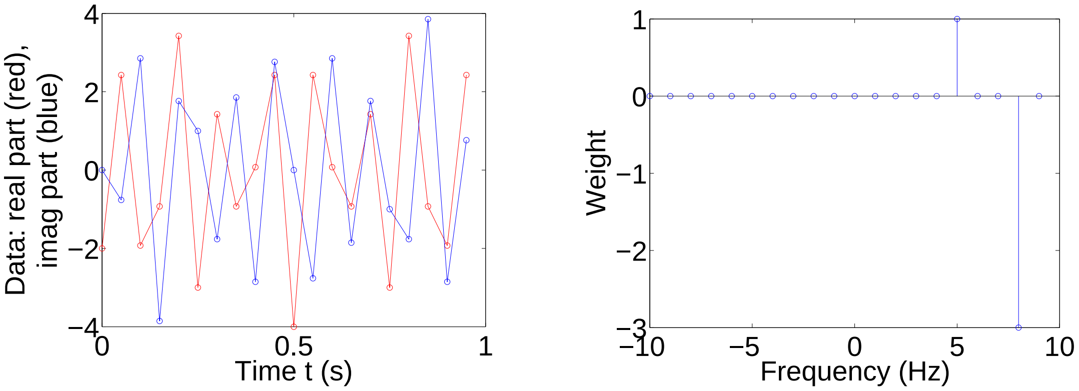


Figure 6: ***Left:*** *Data sampled at 20 Hz of two complex exponentials at frequencies 5 Hz and −12 Hz.* ***Right:*** *The lower sampling rate preserves the 5 Hz peak, but the −12 Hz peak has been aliased to −12 Hz + sampling rate = 8 Hz.*

Nyquist frequency—i.e., the sampling frequency) until the result is within the frequency domain. So we find the correct frequency "aliased" to −12 Hz + 20 Hz = 8 Hz, as seen at the right in Figure 6.

More realistically, suppose now that we sample the same function from 0 to fully 1 second, since we'll generally have no knowledge of what frequencies make up the spectrum—that is, after all, why we are Fourier transforming the data in the first place! This will add one extra sampling point to the sinusoids, with the result that the DFT no longer sees two pure sinusoids; rather, it sees a jump where each copy of the signal begins and ends, in the infinite chain of copies placed side by side in time that the DFT works with. As a result, even if we do sample quickly enough to capture both frequencies—again at 1/40 second at top left in Figure 7—the discontinuity in the joining of the sinusoids in the data results in many more pure sinusoids (i.e. frequencies) being present in the Fourier spectrum. That is, more sinusoids are required to synthesise a curve with a discontinuous jump.

This sampling behaviour of the discrete Fourier transform is the reason why more frequencies are present (now with complex weights) in Figure 7 around the dominant frequencies of 5 Hz and −12 Hz. This broadening of the peaks in the Fourier spectrum is an unavoidable consequence of our practical inability to sample the data set for just the right amount of time to ensure each sinusoid joins smoothly to a copy of itself when the data is repeated endlessly by the Fourier transform.

6.1 Windowing Sampled Data

The above jump that occurs in the wrap-around of the sampling sequence will almost always be present in any real sample, because we almost never deal with simple sinusoids along with time intervals that have fortuitously been chosen as accurately as was done to create Figures 5 and 6. The extraneous frequencies introduced by this jump can be partly removed by eliminating the jump; that is, by forcing the data to match up in the sampling period wrap-around that the Fourier transform "assumes" takes place. The usual way of doing this is simply to reduce the data values to zero near the beginning

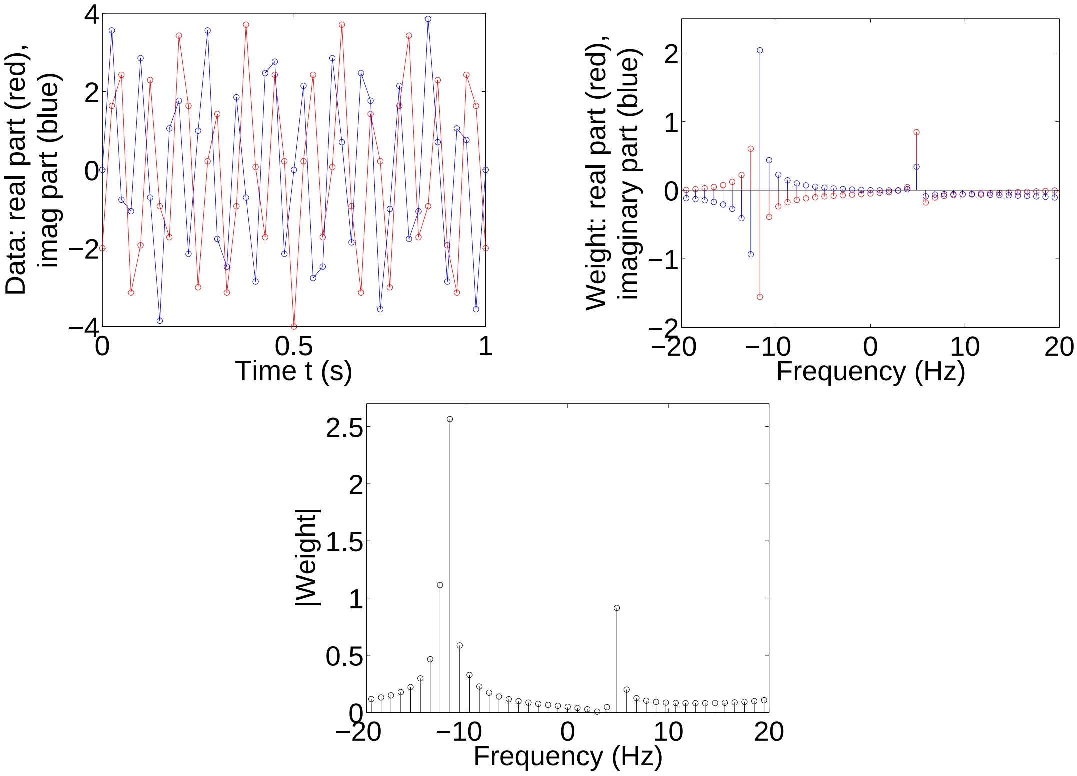


Figure 7: ***Top left:*** *Data sampled at 40 Hz of two complex exponentials at frequencies 5 Hz and −12 Hz.* ***Top right:*** *The data has been sampled in such a way as to cause a jump where the DFT "joins" copies of it together in an infinite chain. The result is that more frequencies are required to reconstruct the data.* ***Bottom:*** *Absolute values of the complex weights of the spectrum at top right.*

and end of the sampling period. This is done by multiplying the data by a smooth bell-shaped function that has value zero at the end points of the data sequence and value one in its middle. Several choices of this "windowing" function exist, each with different trade-offs in the resulting spectrum. Perhaps the two most popular are the Hamming and Blackman windows. The three main indicators of filter performance in the frequency domain are:

Flat passband: Ideally, frequencies required to be passed with equal weightings should not acquire any ripples in those weightings. The Blackman window has less passband ripple than the Hamming.

High roll-off rate: There should be a well-defined drop in the Fourier spectrum between frequencies required to be passed and frequencies required to be stopped. The Hamming window has a slightly faster (better) roll-off than the Blackman.

Good stopband attenuation: Frequencies required to be stopped by the filter should be attenuated very strongly. The Blackman window has a better stopband attenuation than the Hamming.

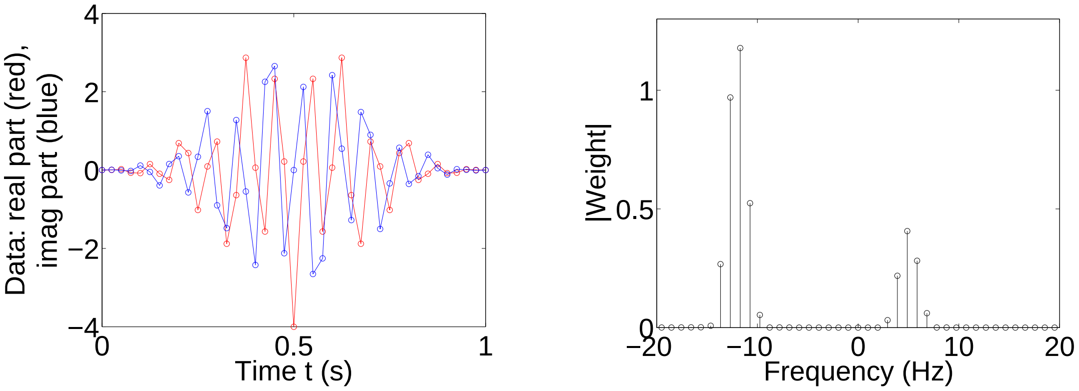


Figure 8: ***Left:*** *The sampled data of Figure 7, now Blackman windowed.* ***Right:*** *The neighbouring frequencies near 5 Hz and −12 Hz have been suppressed, at the cost of the peaks at 5 and −12 being widened.*

The stopband attenuation is arguably more important than roll-off, so the Blackman window might be considered as generally the most desirable to use, as discussed in [5].

If we take the "badly sampled" data plotted in Figure 7 and create a new data set by multiplying this data by a Blackman window, and then plot the absolute value of the DFT of this new data, we obtain the plots of Figure 8. The windowing does suppress the spurious "side lobes" around 5 Hz and −12 Hz, but at the cost of broadening and lowering the two peaks at those frequencies.

Windowing is used in range–Doppler processing to reduce side lobes in both Doppler and range. For Doppler windowing, recall that for each range value, the Doppler information in the received pulse is simply the Fourier spectrum of the column of correlations in (5.9). As discussed in the previous few pages and in Figure 8, we can eliminate spurious Doppler frequencies by windowing each of these columns before Fourier-transforming them.

Windowing in range can be performed by windowing a copy of the emitted pulse, then correlating that copy with the returned pulse. For modelling the radar interaction, this procedure modifies (4.10) in the following way. Write the windowed emitted pulse as $\boldsymbol{E}_w$, in which case (4.10) becomes

$$\text{correlation} = \boldsymbol{E}_w \star \boldsymbol{R} = \boldsymbol{E}_w^\dagger * \boldsymbol{R} = \boldsymbol{E}_w^\dagger * (\boldsymbol{E} * \boldsymbol{h}) = (\boldsymbol{E}_w^\dagger * \boldsymbol{E}) * \boldsymbol{h}\,. \tag{6.6}$$

As with the unwindowed case, $\boldsymbol{E}_w^\dagger * \boldsymbol{E}$ can be calculated offline.

Range–Doppler plots showing "before and after" results of Doppler windowing are given in Section 10. The use of range windowing does not give as dramatic an improvement in the final plot as the use of Doppler windowing, so no range windowing has been used in those plots.

7 Representative Pulse Types

A rectangular pulse can be viewed as a continuous wave being switched on for a time τ, its pulse width. Aside from this, we investigate two other pulse types in this report.

7.1 Barker-coded Pulse

A Barker-coded pulse has more structure than a rectangular pulse of the same width. Signals with rich structure autocorrelate very well (i.e. they correlate with themselves to give a well-defined and narrow single peak, resulting in a high-quality matched filter), but the larger bandwidth required to build this structure destroys some Doppler information when compared to the smaller bandwidth of a rectangular pulse. The reason is that a Doppler frequency shift will generally be less apparent (i.e. less measurable) when a larger spread of frequencies is present.

The modulation that defines a Barker code is a series of rectangular pulses, called *chips*, placed side by side with no gap, with weightings either +1 or −1; that is, they either have no effect on the phase of the carrier (+1) or they add 180° to it (−1). Such a pulse autocorrelates to give a high central peak when the signal matches with itself, flanked by low correlation "side lobes" with a saw-tooth shape. Demanding an even saw-tooth pattern of side lobes results in just seven known Barker codes: it has been conjectured that no longer Barker codes exist. For an example of Barker autocorrelation, consider the length-7 Barker code: $[+1\ +1\ +1\ -1\ -1\ +1\ -1]$ (i.e. this has 7 chips). This autocorrelates to give the following, remembering from Appendix B that correlation is accomplished by complex-conjugating one reversed signal (which is real in this case) and convolving this with the other signal—which here is of course identical to the first signal:

$$\begin{aligned}\begin{bmatrix}-1 & 1 & -1 & -1 & 1 & 1 & 1\end{bmatrix} &* \begin{bmatrix}1 & 1 & 1 & -1 & -1 & 1 & -1\end{bmatrix}\\ &= \begin{bmatrix}-1 & 0 & -1 & 0 & -1 & 0 & 7 & 0 & -1 & 0 & -1 & 0 & -1\end{bmatrix}. \qquad (7.1)\end{aligned}$$

We see here the high peak in the middle, with the characteristic Barker saw-tooth side lobes.

7.2 Chirped Pulse

A chirped signal varies its frequency continuously throughout each of its pulses. This endows it with much structure, resulting in a higher-quality autocorrelation. The single chirped pulse thus acquires the range resolution that only a much shorter constant-frequency pulse would have. But the benefit of using the longer (chirped) pulse is that this long pulse requires a lower energy density than a short (non-chirped) pulse to achieve the same result, and so is both easier to generate and less visible to the target than the non-chirped pulse.

As in (2.13), write the emitted signal as $s_c(t) = u(t)\, e^{i\omega_0 t}$. We will show shortly that for a signal beginning at time t_0 composed of N chirped pulses of period T and width τ, the emitted chirped signal is

$$s_c(t) = e^{i\omega_0 t}\Big[F(t-t_0) + F(t-t_0-T) + \dots + F(t-t_0-\overline{N-1}\,T)\Big], \qquad (7.2)$$

where

$$F(t) = \begin{cases} e^{i\pi\mu t^2} & 0 \leqslant t \leqslant \tau \\ 0 & \text{otherwise} \end{cases} \qquad (7.3)$$

for some constant μ. (In fact, it will turn out that such an $s_c(t)$ is *not* quite an analytic signal because its spectrum can contain negative frequencies; however, these are negligibly present when the carrier frequency is large enough, which it usually is. See further discussion and examples in Section 9.1.) The rationale behind the form of the modulation $F(t)$ in (7.3) is as follows. A general signal $s_c(t) = A(t)\,\mathrm{e}^{\mathrm{i}\phi(t)}$ can be Fourier decomposed as

$$s_c(t) = \int\limits_{-\infty}^{\infty} S_c(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega\,, \tag{7.4}$$

where

$$S_c(\omega) = \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} s_c(t)\,\mathrm{e}^{-\mathrm{i}\omega t}\,\mathrm{d}t = \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} A(t)\,\mathrm{e}^{\mathrm{i}[\phi(t)-\omega t]}\,\mathrm{d}t\,. \tag{7.5}$$

Now choose any value of ω: almost no contribution to the last integral in (7.5) occurs when its phase $\phi(t) - \omega t$ is changing rapidly with time, because in such a case many complex numbers of essentially random phases are being added, giving a result of about zero. This cancellation won't happen when $\phi(t) - \omega t$ is not changing rapidly with time. That is, we expect the dominant contribution to the integral to occur when $\phi(t) - \omega t$ is roughly constant over time, or $\mathrm{d}/\mathrm{d}t\,[\phi(t) - \omega t] \approx 0$, or $\phi'(t) \approx \omega$. (This reasoning is known as the *method of stationary phase*.) For example, at some time, say $t = 41$, $S_c(\omega)$ has the most support at $\omega = \phi'(41) =$ (say) 2. That is, we presume that at $t = 41$, the frequency spectrum $S_c(\omega)$ is peaked around a dominant frequency of $\omega = 2$. In general, at time t the dominant frequency of the signal $s_c(t) = \mathrm{e}^{\mathrm{i}\phi(t)}$ is

$$\boxed{\omega \approx \phi'(t)\,.} \tag{7.6}$$

Refer to (7.2) and (7.3) and focus on the basic signal component $\mathrm{e}^{\mathrm{i}\omega_0 t + \mathrm{i}\pi\mu t^2}$. At time t its dominant angular frequency is $\omega_0 + 2\pi\mu t$, which is the carrier plus an extra part that grows linearly with time; this constitutes the chirp and is the rationale for the form of (7.3). The angular frequency $\omega_0 + 2\pi\mu t$ corresponds to an actual frequency of $f_0 + \mu t$, which explains why π is conventionally included in (7.3). It's clear that μ is the slope of the straight-line plot of chirped frequency versus time, as shown in Figure 10 on page 36.

8 Selecting Valid Pulse Parameters

In this section we describe how to choose pulse parameters that guarantee valid range–Doppler plots. We illuminate a single possibly moving target at distance r from our radar by the train of pulses shown in Figure 9. The parameters describing this pulse train are used in the subsections that follow.

8.1 Attainable Range Measurement of a Pulse Train

A radar's receiver is necessarily switched off during the time interval τ that it emits a pulse. Any target closer than $c\tau/2$ will then cause some of the reflected pulse to be received while the receiver is switched off. This minimum range $c\tau/2$ is called the radar's *blind range*.

Likewise, the return from any target at range greater than $cT/2$ (where T is the PRI) will arrive at the radar receiver either during or after it emits the *next* pulse of its train.

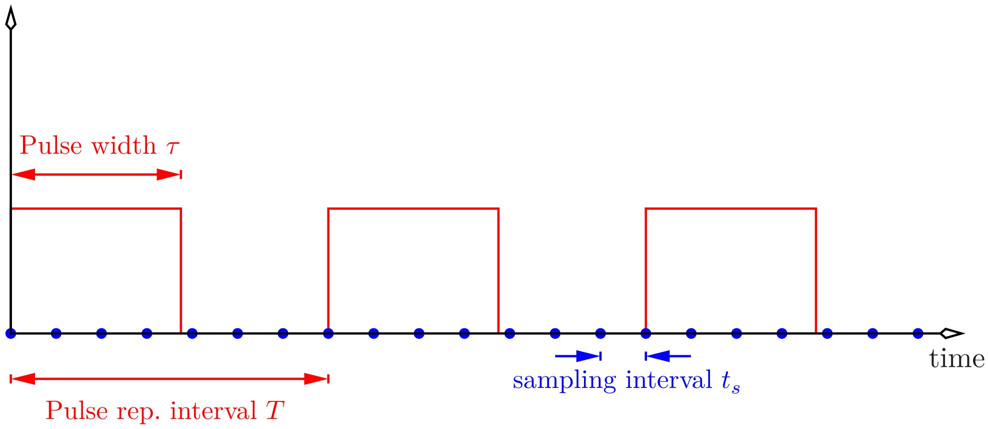


Figure 9: *A series of pulses with parameters indicated. Only the envelopes of the pulses are shown.*

We cannot know that such a return belonged to the first pulse and not the second, so ranges greater than $cT/2$ create ambiguities in the measured range; thus $cT/2$ is called the *maximum unambiguous range*. The radar's useful range extent then runs from the blind range $r_{\text{min}} = c\tau/2$ to the maximum unambiguous range $r_{\text{max}} = cT/2$:

$$\boxed{c\tau/2 \leqslant r \leqslant cT/2\,.} \tag{8.1}$$

Note that $r_{\text{min}} = c\tau/2$ is also the *range resolution* or *range bin width* Δr_{res} of a simple rectangular pulse [i.e. $\mu = 0$ in (7.3)], being the difference of the two closest-together ranges r_1, r_2 such that the returned pulse from a target at r_1 will be received without overlapping the returned pulse from a target at r_2. Section 9—starting with (9.1)—calculates the range resolution of pulses with more structure.

8.2 Bounds on the Pulse Repetition Interval T

Given a target at range r, the above discussion of blind range and maximum unambiguous range shows that τ and T must be chosen to ensure the target is visible to the radar, or

$$c\tau/2 \leqslant r \leqslant cT/2\,. \tag{8.2}$$

This places bounds on the pulse width τ and the PRI T:

$$c\tau_{\text{max}}/2 = r = cT_{\text{min}}/2\,. \tag{8.3}$$

We focus in particular on T_{min}:

$$T_{\text{min}} = 2r/c\,. \tag{8.4}$$

An upper bound T_{max} to the PRI arises by avoiding Doppler ambiguities in the following way. Refer to (2.27), which gives the Doppler frequency as

$$f_D = -2f_0\, v/c \tag{8.5}$$

for a carrier frequency f_0. We will be extracting the Doppler frequency from numbers sampled at a rate of $1/T$ (*not* $1/t_s$: remember that Doppler information is sampled from pulse to pulse, at *one* range value). Nyquist's theorem says that approximately half this rate is the maximum Doppler frequency that a DFT can detect,[8] so

$$\frac{2f_0|v|}{c} = |f_D| < \frac{1}{2T}\,. \tag{8.6}$$

This rearranges to give

$$T < \frac{c}{4f_0|v|}\,, \quad \text{so that } \ T_{\max} = \frac{c}{4f_0|v|}\,. \tag{8.7}$$

We must choose a PRI between $T_{\min}$ and $T_{\max}$:

$$\boxed{\frac{2r}{c} < T < \frac{c}{4f_0|v|}\,.} \tag{8.8}$$

8.3 Attainable Speed Measurement of a Pulse Train

There is a *maximum unambiguous speed* $v_{\max}$ that a given pulse train is able to detect, irrespective of whether the target is approaching or receding. Also, while the train can in principle detect a minimum speed of zero, it does have a *speed resolution* Δv_{res}. We will calculate $v_{\max}$ and Δv_{res} here. First, (8.5) gives the target's radial velocity (positive for receding) as

$$v = \frac{-cf_D}{2f_0}\,. \tag{8.9}$$

Remember from (8.6) that the maximum-detectable Doppler frequency is approximately $f_D^{\max} \equiv 1/(2T)$. The maximum unambiguous speed comes from the absolute value of (8.9):

$$\boxed{v_{\max} = \frac{cf_D^{\max}}{2f_0} = \frac{c}{2f_0}\frac{1}{2T} = \frac{c}{4f_0T}\,.} \tag{8.10}$$

The speed resolution Δv_{res} arises because the *calculated* Doppler spectrum is a set of points and not a continuous curve, since this spectrum results from Fourier-transforming the N numbers in (5.9). The resolution corresponds to the speed spacing between these points. The Doppler frequency spacing is $\Delta f_D = 1/(NT)$ [use (6.3) with t_s replaced by T], and this converts to the speed resolution using the same factor as in (8.9):

$$\boxed{\Delta v_{\text{res}} = \left|\frac{-c\Delta f_D}{2f_0}\right| = \frac{c}{2f_0NT} = \frac{c}{2f_0\Delta t_{\text{CPI}}}\,,} \tag{8.11}$$

where Δt_{CPI} is the length NT of the CPI.

[8] I say "approximately" because the actual number is $\hat{f}$ in (6.4) and (6.5), where the t_s of those equations is replaced by T here.

Alternative Views of $v_{\mathbf{max}}$ and $\Delta v_{\mathbf{res}}$

It's instructive to derive the maximum unambiguous speed detectable by one particularly simple wave train in an alternative way to the above. Suppose the radar emits an infinite number of impulses spaced T apart. (That is, let $\tau \to 0$ and $N \to \infty$.) The frequency spectrum of this train is easily calculated: it turns out to be an infinite series of spikes spaced $1/T$ apart. We can define the maximum unambiguous speed to be that whose effect is to shift these spikes by no more than half their spacing, since otherwise we wouldn't be able to tell whether any particular spike had moved to the left or the right. That is, we require the absolute value of the Doppler shift, $f_0 2v/c$ from (2.27), to be less than half the frequency spike spacing of $1/T$:

$$\frac{f_0 2v}{c} \leqslant \frac{1}{2T}\,. \tag{8.12}$$

Equality in (8.12) defines v_{max}:

$$\frac{f_0 2v_{\text{max}}}{c} \equiv \frac{1}{2T}\,, \tag{8.13}$$

and this rearranges to give (8.10) again.

The speed resolution Δv_{res} is sometimes related heuristically to a simple wave picture in the following way. (This argument doesn't use the above infinite wave train.) Recall (8.11):

$$\Delta v_{\text{res}} \equiv \frac{c\Delta f_D}{2f_0}\,, \tag{8.14}$$

where Δf_D is the (positive) spacing between points in the Doppler spectrum—which is also the smallest positive Doppler frequency measurable. Let this smallest frequency correspond to a nominal longest "Doppler wavelength" $\lambda_D \equiv c/\Delta f_D$ (but note that this does not correspond to a physical wave). Because $\Delta f_D = 1/(NT) = 1/\Delta t_{\text{CPI}}$, we have

$$\lambda_D = \frac{c}{\Delta f_D} = c\,\Delta t_{\text{CPI}}\,. \tag{8.15}$$

This equation portrays the speed resolution as resulting from the fitting of precisely one of these longest "Doppler wavelengths" into the distance travelled by light in one coherent processing interval.

Distance–Velocity Maxima Tradeoff

Equations (8.1) and (8.10) combine to give

$$r_{\text{max}}\, v_{\text{max}} = \frac{c^2}{8f_0}. \tag{8.16}$$

We see here a tradeoff between maximum unambiguous range and maximum unambiguous speed: for a given carrier frequency f_0, increasing one must decrease the other. This is another reason why high carrier frequencies are desirable in radar: a high carrier frequency f_0 allows both r_{max} *and* v_{max} to be large simultaneously. But see the remark just after (8.18) ahead.

8.4 Doppler Aliasing

The target's radial velocity was given in terms of the Doppler frequency in (8.9). But all velocities that produce Doppler frequencies outside the DFT's domain will be *aliased* into that domain by the action of the DFT: the DFT will add the necessary number of multiples of the frequency domain width $1/T$ to the Doppler frequencies of those higher velocities to shift their frequencies into that domain. In other words, the Doppler frequency $f_D^{\rm DFT}$ produced by the DFT (which might not equal the actual Doppler frequency f_D) will be indistinguishable from all frequencies present in the data that have value $f_D^{\rm DFT} + n/T$ with n ranging over all integers. Those other frequencies correspond to velocities v_n via (8.9):

$$v_n \equiv \frac{-c\,(f_D^{\rm DFT} + n/T)}{2f_0} \quad \text{for all integers } n. \tag{8.17}$$

So all v_n will be aliased to the v given by (8.9), meaning that a target with recession velocity v_n (for any integer n) will appear to have velocity v on the range–Doppler plot. In particular, target velocities that are aliased down to zero speed are potentially hazardous, since they become indistinguishable from the background and hence cannot trigger any velocity-sensitive sensor to acquire the target as being possibly of interest. These *blind speeds* are $|v_n|$ such that $f_D^{\rm DFT} = 0$, or

$$|v_n| \equiv \frac{cn}{2f_0T}\,, \quad \text{for } n = 1, 2, 3, \dots \tag{8.18}$$

Unfortunately, given that the spacing between blind speeds is $c/(2f_0T)$, a high carrier frequency f_0 allows more blind speeds in some given speed range. This is a counter-argument to the remark just after (8.16) that high carrier frequencies are desirable.

8.5 Range–Doppler Coupling

Refer to Figure 10, whose black line segment is the frequency versus time of a single chirped pulse of duration τ, the pulse width. A stationary target returns a pulse whose frequency is plotted as the blue line segment, and a moving target at the same range returns a pulse whose frequency is plotted as the red line segment. The amount of Doppler shift f_D in the picture is exaggerated; a realistic figure is about one millionth of the carrier frequency f_0.

The slope of the red segment is actually slightly different to that of the blue: if the frequency of the emitted pulse is $f = f_0 + \mu t$ (refer Section 7.2), and the returned pulses start at time t_d (the time delay in Figure 1), then the blue segment is just the black segment shifted to the right by t_d, meaning the blue has equation

$$f = f_0 + \mu(t - t_d)\,. \tag{8.19}$$

This has slope μ. But (2.25) says that the received frequency is $\omega_0(1 - 2v/c)$, so that the red segment has equation

$$f = (1 - 2v/c)\,[f_0 + \mu(t - t_d)]\,, \tag{8.20}$$

which has slope $(1 - 2v/c)\mu$. The value of v/c is typically $\pm 10^{-6}$, so the slopes of red and blue are almost identical.

The key feature of Figure 10 relates to finding the parts of the emitted and received pulses that correlate best. The emitted pulse segment labelled "1" in the figure (the small black rectangle) will correlate well with the returned segment labelled 2 in the absence of Doppler shift. In the presence of Doppler shift, we might hope segment 1 would correlate

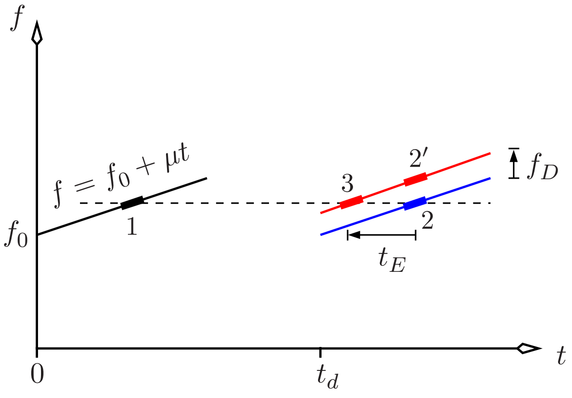


Figure 10: *Range–Doppler coupling for a chirped pulse. The black line segment is the emitted pulse. A returned pulse with no Doppler shift is the blue line segment, and a returned pulse with Doppler shift is the red line segment, in this case for an approaching target.*

well with segment 2′ in order to derive the same time shift necessary to give the range. But, in fact, segment 1 might better correlate with segment 3 since they share the same frequency. (This assumes *some* chirp is present, in which case we cannot expect to be able to take the limit $\mu \to 0$ at the end of our calculation.) That means the Doppler shift has the effect of making it appear as if the returned pulse arrived early by an error time t_E, for the case when f_D has the same sign as μ. (If the signs are different, the red segment will appear to have arrived later than the blue segment.) Because the red line in the figure has about the same slope as the blue and black lines, we can write $\mu \simeq f_D/t_E$ for any choice of signs, and infer a range error of

$$\begin{aligned}\Delta r &= \text{measured range} - \text{true range}\\ &= \frac{c(t_d - t_E)}{2} - \frac{ct_d}{2}\\ &= \frac{-ct_E}{2} = \frac{-cf_D}{2\mu} = \frac{-c}{2\mu} \times \frac{-2vf_0}{c} = \frac{vf_0}{\mu}\,, \quad \mu \not\to 0\,. \end{aligned} \tag{8.21}$$

This error is called *range–Doppler coupling*, and amounts to about 7 centimetres for the target of Figure 16 ahead. Given that it has such a small value for our scenarios of interest, we don't consider it further in this report.

9 Range Resolutions of Different Pulse Types

A standard expression in radar signal processing is that a pulse's range resolution, or range bin width, is given by

$$\Delta r_{\text{res}} \approx \frac{c}{2B}\,, \tag{9.1}$$

where B is the pulse's bandwidth. This isn't a hard and fast rule, in that it suggests that "bandwidth" is a well-defined term, which isn't quite the case. In principle the total bandwidth of a signal is defined as the width (say, the "full width at half maximum") of the support of the signal's frequency spectrum; but this width can be infinite, so in practice the total bandwidth is defined as the width of the dominant part of the spectrum's support.

One rationale for (9.1) is the following. Most pulses used in radar have a spectrum that is shaped something like a top hat of width B, meaning this spectrum's support is fairly well defined and the spectrum is reasonably flat throughout this width. Neglecting noise, the returned pulse's spectrum is similar to the emitted pulse's spectrum, save for a small Doppler shift. The correlation of the emitted signal $\boldsymbol{E}$ with the returned signal $\boldsymbol{R}$ is $\boldsymbol{E} \star \boldsymbol{R} = \boldsymbol{E}^{\dagger} * \boldsymbol{R}$; in other words, a convolution of a reversed and conjugated $\boldsymbol{E}$ with $\boldsymbol{R}$. This convolution is the inverse Fourier transform of a product: the product of the Fourier transform of $\boldsymbol{E}^{\dagger}$ and the Fourier transform of $\boldsymbol{R}$, probably with some zero appending as discussed around (B11), which will alter the bandwidths somewhat, but presumably not by much. The Fourier transform of $\boldsymbol{E}^{\dagger}$ has bandwidth B, and the Fourier transform of $\boldsymbol{R}$ has bandwidth approximately B, and both spectra are centred at almost the same frequency. So their product is approximately another top hat of width B. The inverse Fourier transform of this bandwidth-B function (which will be the correlation $\boldsymbol{E} \star \boldsymbol{R}$) must then have a typical width of $1/B$. But a well-crafted pulse's correlation with its return will be a spike, so that this spike has a width of about $1/B$, meaning the correlation peak is spread over a time interval of $1/B$. The associated range resolution of the pulse is $c/2$ times this width, which gives (9.1).

Equation (9.1) is reasonable qualitatively, in that a large bandwidth (large B) means high pulse complexity, which means high autocorrelation ability, which leads to very good range resolution (small Δr_{res}). Fourier analysis shows that bandwidth of a rectangular pulse formed by multiplying a single-frequency carrier by a top-hat window of duration τ is $B \approx 1/\tau$, for which case (9.1) produces the expected $\Delta r_{\text{res}} = c\tau/2$ quoted just after (8.1).

So range resolution is determined by bandwidth, not pulse width. Suppose we have a radar pulse of duration τ and bandwidth B, which might have some internal structure such as a chirp or a Barker code. The bandwidth of this highly structured pulse is greater than the bandwidth of a duration-τ rectangular pulse (i.e. one with no internal structure apart from its carrier), this bandwidth being $1/\tau$. Now imagine a rectangular pulse with no structure, duration $1/B$, and bandwidth B, which has therefore the same range resolution as the actual pulse. If the actual pulse has high structure (large B), this imagined rectangular pulse with the same large bandwidth B will have a very small duration $1/B$, smaller than τ. This smaller duration of this (structureless) rectangular pulse is called the *effective pulse width* of the original highly structured pulse: $\tau_{\text{eff}} \equiv 1/B < \tau$. Although this imagined narrow and simple-to-generate pulse has the same range resolution as the wide and difficult-to-generate structured pulse, emitting the imagined simple pulse over the comparatively short time τ_{eff} requires the radar to reach a high power level, which might be impossible; better is for the radar to emit the structured pulse over the longer time τ, for which a lower average power suffices. Also, a lower average emitted power means the radar is less visible to the target.

The actual pulse of duration τ is sometimes called the *uncompressed pulse*, while the imagined plain rectangular pulse of duration τ_{eff} is sometimes called the *compressed pulse*. The *compression ratio* is defined as the ratio of these widths:

$$\text{compression ratio} \equiv \frac{\tau}{\tau_{\text{eff}}} = B\tau \geqslant 1\,. \tag{9.2}$$

The compression ratio is a measure of the emitted pulse's structure: higher structure means higher compression ratio. Also, the compression ratio determines the emitted pulse's range resolution using (9.1):

$$\Delta r_{\text{res}} \approx \frac{c}{2B} = \frac{c\tau_{\text{eff}}}{2} = \frac{c\tau/2}{\text{compression ratio}} \,. \tag{9.3}$$

In words, the range resolution Δr_{res} of the actual long structured pulse (the "uncompressed pulse", duration τ) equals the range resolution of the imagined short structureless rectangular "compressed pulse" (duration $\tau_{\text{eff}} < \tau$), which equals the range resolution of an imagined structureless rectangular pulse (duration τ, *not* τ_{eff}) divided by the compression ratio.

In some analogue radars, chirped pulses are physically compressed on reception by filtering them through, for example, a "surface acoustic wave device", which passes different frequencies at different speeds to physically compress the returned pulse, achieving the range resolution of this "compressed pulse" without the high power requirement of a short pulse. But modern digital radars no longer compress pulses on reception, so for them the terms "uncompressed pulse", "compressed pulse", and "compression ratio" are not literal: the terms are not meant to imply that the received (or emitted) *radar pulse* is ever compressed. (I think this terminology of compression, once useful for analogue radars, is misleading for digital radars and shouldn't be used.) Note also that in some texts, the actual pulse's *autocorrelation function* is also called the compressed pulse, as might be the actual pulse's correlation with its return. So, what is called a compressed pulse might be (a) an imagined narrow pulse of duration τ_{eff}, or (b) a correlation sequence of the actual ("uncompressed") pulse with either itself or its return; and this sequence is not a radar pulse at all.

Either way, in a digital radar, a compressed pulse is *not* the compressed version of some kind of "uncompressed" pulse. Nothing ever gets compressed. In the digital radar age where surface acoustic wave devices are no longer used to physically compress radar pulses, the technique of frequency modulation is not meant to be visualised as having the magical effect of physically compressing a radar pulse. It simply gives the pulse more structure, which leads to better autocorrelation. The emitted or "uncompressed" pulse is the *only* radar pulse that actually exists.

9.1 Range Resolution of a Chirped Signal

Using a large chirp [a large μ in (7.3)] endows a pulse with more structure, which in turn gives it more bandwidth (a "high compression ratio"), thus giving a better (i.e. smaller) range resolution.

Let's calculate the amount of chirp μ in (7.3) required for a given pulse width and desired range resolution. The range resolution is given by (9.1):

$$\Delta r_{\text{res}} \approx \frac{c}{2B_{\text{tot}}} \,, \tag{9.4}$$

where we've now written B_{tot} for the total bandwidth of a single chirped pulse; this bandwidth is distinct from the amount by which the pulse's instantaneous frequency changes during the chirp, which is its *swept* bandwidth B_{swept}. Although the notion of bandwidth is not scrupulously well-defined, we can find an approximate value of this total bandwidth by Fourier transforming a single chirped pulse, described next.

Calculating the Bandwidth of a Chirped Pulse

Over the duration τ of a single pulse, the chirped wave train of (7.2) and (7.3) is

$$s_c(t) = \mathrm{e}^{\mathrm{i}2\pi\left(f_0 t + \frac{\mu}{2}t^2\right)} \equiv \mathrm{e}^{\mathrm{i}\phi(t)}\,, \tag{9.5}$$

where f_0 is the carrier frequency and μ is the chirp parameter, quantifying how the signal frequency changes during the pulse. Equation (7.6) expresses the frequency used to quantify the bandwidth as

$$f(t) = \frac{\phi'(t)}{2\pi} = f_0 + \mu t\,, \tag{9.6}$$

showing that the instantaneous frequency $f(t)$ does indeed grow linearly. The pulse's frequency spectrum $S_c(\omega)$ is given by the Fourier transform of $s_c(t)$:

$$S_c(\omega) = \int_{-\infty}^{\infty} \mathrm{e}^{-\mathrm{i}\omega t} s_c(t)\,\mathrm{d}t = \int_0^{\tau} \mathrm{e}^{\mathrm{i}(-\omega+\omega_0)t + \mathrm{i}\pi\mu t^2}\mathrm{d}t\,, \tag{9.7}$$

where $\omega_0 \equiv 2\pi f_0$. Evaluate this integral with the identity

$$\int \mathrm{e}^{-ax^2+bx}\,\mathrm{d}x = \frac{1}{2}\sqrt{\frac{\pi}{a}}\,\mathrm{e}^{\frac{b^2}{4a}}\,\mathrm{erf}\left(\sqrt{a}\,x - \frac{b}{2\sqrt{a}}\right), \tag{9.8}$$

to give

$$S_c(\omega) = \frac{1}{2}\sqrt{\frac{\mathrm{i}}{\mu}}\,\mathrm{e}^{\frac{-\mathrm{i}(\omega-\omega_0)^2}{4\pi\mu}}\left[\mathrm{erf}\left(\sqrt{-\mathrm{i}\pi\mu}\,\tau + \frac{\mathrm{i}(\omega-\omega_0)}{2\sqrt{-\,\mathrm{i}\pi\mu}}\right) - \mathrm{erf}\,\frac{\mathrm{i}(\omega-\omega_0)}{2\sqrt{-\,\mathrm{i}\pi\mu}}\right]. \tag{9.9}$$

Note that μ is not trivially able to be set to zero here, because the identity (9.8) relies on the process of "completing the square", which has no meaning if $\mu = 0$. But we can certainly set μ to be arbitrarily small.

Alternatively, we could make the real and imaginary parts of $S_c(\omega)$ more explicit by using the Fresnel integrals $S(x)$ and $C(x)$, where

$$\begin{Bmatrix} S(x) \\ C(x) \end{Bmatrix} \equiv \int_0^x \begin{Bmatrix} \sin \\ \cos \end{Bmatrix} \frac{\pi u^2}{2}\,\mathrm{d}u\,. \tag{9.10}$$

These lead to the identity

$$\int \mathrm{e}^{\mathrm{i}ax^2+\mathrm{i}bx}\,\mathrm{d}x = \sqrt{\frac{\pi}{2a}}\,\mathrm{e}^{\frac{-\mathrm{i}b^2}{4a}}\left[C(u) + \mathrm{i}S(u)\right] \tag{9.11}$$

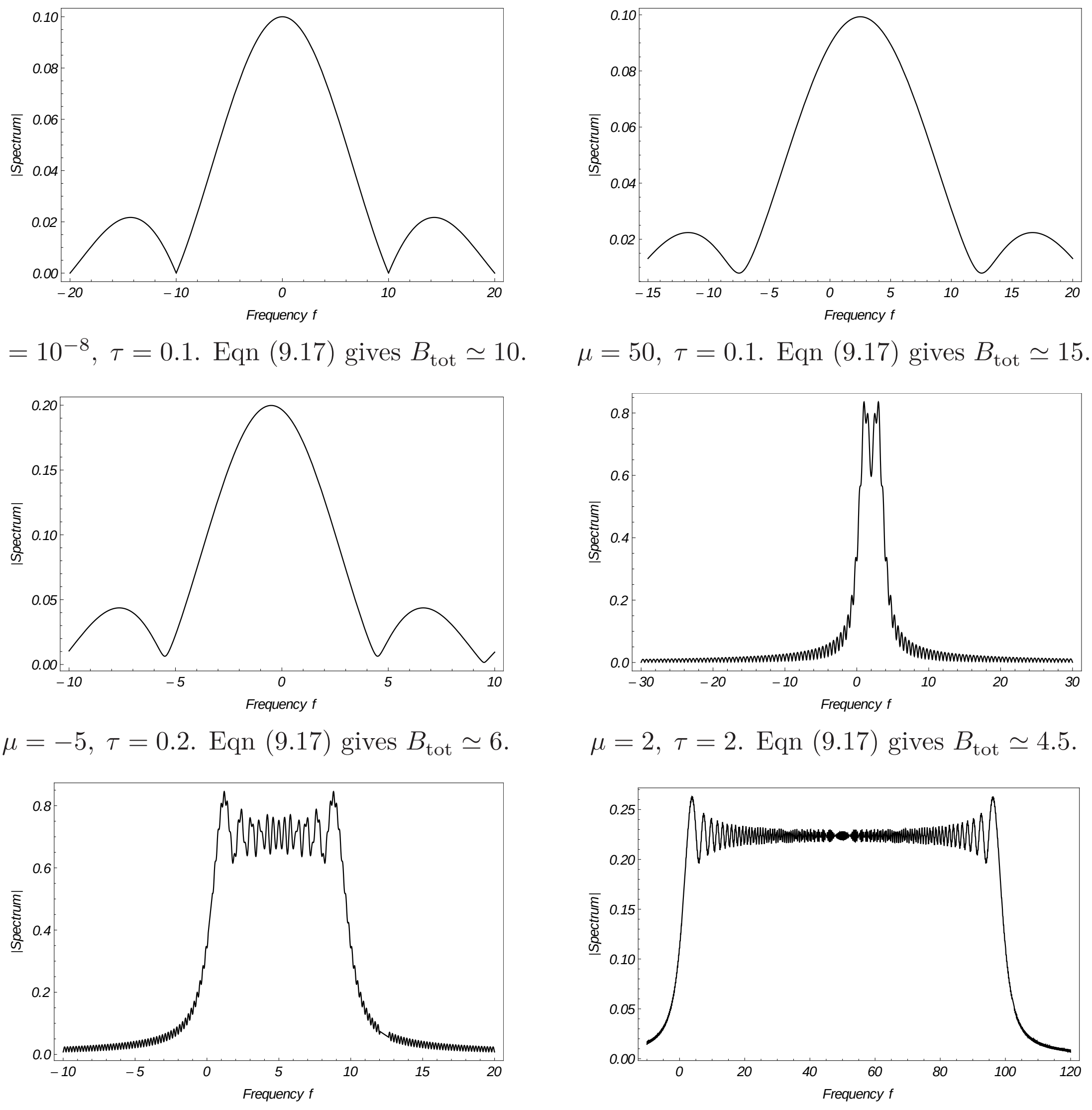
where

$$u \equiv \sqrt{\frac{2a}{\pi}}\,x + \frac{b}{\sqrt{2\pi a}}\,, \tag{9.12}$$

enabling the Fourier transform in (9.7) to be written alternatively to (9.9) as

$$S_c(\omega) = \frac{1}{\sqrt{2\mu}}\mathrm{e}^{\frac{-\mathrm{i}(\omega-\omega_0)^2}{4\pi\mu}}\left[\begin{array}{c} C\left(\sqrt{2\mu}\,\tau + \frac{-\omega+\omega_0}{\pi\sqrt{2\mu}}\right) - C\left(\frac{-\omega+\omega_0}{\pi\sqrt{2\mu}}\right) \\ +\,\mathrm{i}S\left(\sqrt{2\mu}\,\tau + \frac{-\omega+\omega_0}{\pi\sqrt{2\mu}}\right) - \mathrm{i}S\left(\frac{-\omega+\omega_0}{\pi\sqrt{2\mu}}\right) \end{array}\right]. \tag{9.13}$$

Plots of $|S_c(\omega)|$ versus frequency $f = \omega/(2\pi)$ using (9.9) are shown in Figure 11 on the next page. The carrier ω_0 just shifts the spectrum, so has been set to zero in each plot.



$\mu = 10^{-8},\ \tau = 0.1$. Eqn (9.17) gives $B_{\text{tot}} \simeq 10$.

$\mu = 50,\ \tau = 0.1$. Eqn (9.17) gives $B_{\text{tot}} \simeq 15$.

$\mu = -5,\ \tau = 0.2$. Eqn (9.17) gives $B_{\text{tot}} \simeq 6$.

$\mu = 2,\ \tau = 2$. Eqn (9.17) gives $B_{\text{tot}} \simeq 4.5$.

$\mu = 2,\ \tau = 5$. Eqn (9.17) gives $B_{\text{tot}} \simeq 10.2$.

$\mu = 20,\ \tau = 5$. Eqn (9.17) gives $B_{\text{tot}} \simeq 100.2$.

Figure 11: *Frequency spectra of single pulses with various values of μ and τ, whose units are immaterial here. The carrier has been removed; thus zero frequency corresponds to the carrier frequency. The values of the bandwidth calculated using* (9.17) *are listed, and compare well with a visual inspection of, say, the full width at half maximum in each plot.*

Note that usually the bandwidth is much smaller than the carrier frequency, so that in accordance with the comments in Section 2 and Appendix A, the signal $s_c(t)$ in (9.5) essentially has only a positive-frequency spectrum. Nevertheless, were ω_0 to be very small, $s_c(t)$ would certainly have some negative frequencies in its spectrum, which is at odds with the fact that an analytic signal only has positive-frequency components. The fact is that $s_c(t)$ in (9.5) is not *exactly* an analytic signal; applying the Hilbert transform to its real part is not guaranteed to give its imaginary part. But when the carrier frequency is high enough—which is perhaps always the case—the plots of Figure 11 show that the spectrum essentially does have only positive frequencies.

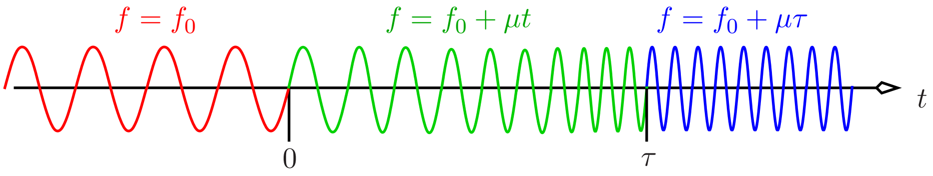


Figure 12: *An idealised fusion of two plane wave trains used to derive* (9.17)

The plots of Figure 11 show that the behaviour of the spectra as μ and τ vary is not straightforward. But there is some latitude in the definition of a spectrum's bandwidth, given that technically each plot in Figure 11 has infinite support. The bandwidth in each plot can be considered or defined as a kind of "full width at half maximum". Here we derive a simple expression for the bandwidth which agrees well with full-width-at-half-maximum values in Figure 11.

Begin by treating a chirped pulse as the product of a particular type of idealised wave train and a top-hat function. The idealised wave train is shown in Figure 12. It has a constant amplitude and the following frequency behaviour:

- For $t = -\infty \to 0$ it has a frequency f_0.
- For $t = 0 \to \tau$ it has a frequency $f_0 + \mu t$, so that its frequency increases linearly at a rate of μ over the width τ of the pulse, reaching a value of $f_0 + \mu\tau$.
- For $t = \tau \to \infty$ it remains at this frequency of $f_0 + \mu\tau$.

The top-hat function is nonzero only for the pulse duration $t = 0 \to \tau$. The real chirped pulse is then the product:

$$\text{chirped pulse} = \text{idealised wave train} \times \text{top-hat of width } \tau. \tag{9.14}$$

The bandwidth B_{tot} is found by Fourier analysing this pulse; here $\mathcal{F}\{\}$ stands for a Fourier transform and "$*$" for convolution:

$$\begin{aligned}\underbrace{\text{spectrum}}_{\text{bandwidth } B_{\text{tot}}} &= \mathcal{F}\{\text{chirped pulse}\}\\ &= \mathcal{F}\{\text{idealised wave train} \times \text{top hat}\}\\ &= \underbrace{\mathcal{F}\{\text{idealised wave train}\}}_{\text{bandwidth } = B_{\text{swept}}} * \underbrace{\mathcal{F}\{\text{top hat}\}}_{\substack{\text{a sinc function}\\ \text{of bandwidth } 1/\tau}}\,. \end{aligned} \tag{9.15}$$

The idealised wave train is a fusion of two plane waves: its early-time or "left part" has a single frequency f_0 (shown in red in Figure 12), its late-time or "right part" has a single frequency $f_0 + \mu\tau$ (shown in blue in Figure 12), and there is an intermediate transition region (green) which passes through all frequencies in between, $f_0 \to f_0 + \mu\tau$. Remember that an infinite plane wave with a single frequency has no bandwidth, so we can approximate (even define) the bandwidth of the entire wave train as being solely the sweep of frequencies in the intermediate green region:

$$B_{\text{swept}} = |\mu|\tau\,. \tag{9.16}$$

This is the first factor in the convolution on the right-hand side of (9.15). The second factor there is a sinc function of well-known bandwidth $1/\tau$. Now, if we convolve two functions of approximate widths $|\mu|\tau$ and $1/\tau$ in (9.15), we can expect the result—which amounts to "smudging" one using the other[9]—to have a width that is approximately the sum of these widths, or $|\mu|\tau + 1/\tau$. So the required single-pulse bandwidth B_{tot} is approximately this sum:

$$\boxed{B_{\text{tot}} \simeq |\mu|\tau + 1/\tau\,.} \tag{9.17}$$

As far as I'm aware, radar textbooks tend not to distinguish between B_{swept} and B_{tot}, and seem only ever to write (9.16) as the expression for "the" bandwidth B. But simply writing "$B = |\mu|\tau$" (or what is more likely, "$B = \mu\tau$") doesn't return the baseline value of $1/\tau$ in the no-chirp case ($\mu = 0$). In contrast, (9.17) *does* return $1/\tau$ in that case.

Applying (9.17) to the μ, τ values in Figure 11 gives the bandwidths listed in its subcaptions. There is a good visual agreement between (9.17)'s B_{tot} and the typical support of each plot, given by e.g. its full width at half maximum.

Now substitute (9.17) into (9.4) to give the range resolution of the chirped pulse:

$$\Delta r_{\text{res}} \simeq \frac{c}{2\left(|\mu|\tau + 1/\tau\right)}\,. \tag{9.18}$$

Given a pulse width τ and a desired range resolution Δr_{res}, we can find the necessary chirp parameter μ from (9.18):

$$\boxed{\mu \simeq \pm\left(\frac{c}{2\Delta r_{\text{res}}} - \frac{1}{\tau}\right)\frac{1}{\tau}\,.} \tag{9.19}$$

There are two values possible here, corresponding to an "up chirp" ($\mu > 0$) and a "down chirp" ($\mu < 0$).

The chirped pulse has an effective pulse width of

$$\tau_{\text{eff}} = \frac{1}{B_{\text{tot}}} = \frac{1}{|\mu|\tau + 1/\tau} = \frac{\tau}{|\mu|\tau^2 + 1}\,. \tag{9.20}$$

Alternatively, its compression ratio is $\tau/\tau_{\text{eff}} = |\mu|\tau^2 + 1$.

Choosing a Sampling Interval for a Chirped Pulse

We require our pulse sampling procedure to capture the emitted wave's modulation adequately, but there is no point in making the sampling frequency excessively high. Given that we will sample at a constant rate, then when the emitted signal is chirped, we need to sample at two or more times the largest modulation frequency present to satisfy Nyquist's criterion for detecting this modulation. However, if we do this and then reduce the chirp to zero (so that its largest modulation frequency goes to zero), the sampling frequency will then go to zero: we will stop sampling it! To prevent this, we also require to sample the pulse at least, say, 10 times. So we must set a lower bound on the sampling frequency.

[9] Convolving two sequences is exactly the same procedure as setting the reverse of one sequence to be a moving-mean template of weights that "smoothens" or "smudges" the other sequence. Which sequence plays which role here is immaterial, because $\boldsymbol{A} * \boldsymbol{B} = \boldsymbol{B} * \boldsymbol{A}$.

To quantify this further, note that (9.5) gives the chirp modulation as $\mathrm{e}^{\mathrm{i}\pi\mu t^2}$, so the instantaneous non-angular frequency of this modulation at time t is approximately, from (7.6),

$$\frac{1}{2\pi}\frac{\mathrm{d}}{\mathrm{d}t}\pi\mu t^2 = \mu t\,. \tag{9.21}$$

Irrespective of whether μ is positive or negative, the absolute value of the modulation frequency is $|\mu|t$, which attains a maximum value of $|\mu|\tau$ at the end of the chirp. We sample at n_1 times this rate at the very least (where n_1 is at least the 2 of the previous paragraph to satisfy Nyquist); i.e. the sampling frequency must be at least $n_1|\mu|\tau$. But also, we require at least n_2 samples in the whole pulse (where n_2 was 10 in the previous paragraph), giving a sampling frequency of at least n_2/τ. So choose the larger of $n_1|\mu|\tau$ and n_2/τ; any larger number would only lead to wasteful oversampling. Now, sampling *frequencies* are reciprocals of sampling *times*, so this choice of larger sampling frequency corresponds to a choice of smaller sampling time:

$$t_s = \min\left\{\frac{1}{n_1|\mu|\tau},\ \frac{\tau}{n_2}\right\}. \tag{9.22}$$

9.2 Range Resolution of a Barker-coded Pulse

The bandwidth of a binary phase-coded pulse such as a Barker approximately equals the bandwidth of one of its chips.[10] Since each chip is just the carrier modulated by a top hat, the chip's bandwidth equals the reciprocal of its duration. If the pulse width is τ, each chip of an n-chip Barker code has duration τ/n; hence the bandwidth of the pulse must be $B = n/\tau$, giving the pulse a range resolution of

$$\Delta r_{\text{res}} \approx \frac{c}{2B} = \frac{c\tau}{2n}\,. \tag{9.23}$$

Compare this to (9.3): the number of chips n plays the same role here as the compression ratio of a chirped pulse. Alternatively, the Barker's effective pulse width is

$$\tau_{\text{eff}} = 1/B = \tau/n\,, \tag{9.24}$$

giving the Barker-n pulse a "compression ratio" of $\tau/\tau_{\text{eff}} = n$.

10 Representative Range–Doppler Plots

In this section we generate range–Doppler plots for various coherent processing intervals of non-chirped and chirped pulses. The first plots use 32 pulses per interval, first without and then with Doppler windowing. The number of pulses per CPI in the windowed set is then increased to 256 as a simple example of showing how this increase leads to a better Doppler resolution—but not a better range resolution. All plots use a 10 GHz carrier.

[10] This statement can be verified numerically when the chip polarities are chosen randomly, but I doubt that any solid proof of it is possible, because bandwidth itself is not precisely defined. Clearly it will fail if the chips all have the same polarity. Its truth is probably due to the idea that generally bandwidth is inversely proportional to signal length, so that a short-duration feature of the signal (such as a chip) requires more bandwidth than a long-duration feature, and so the bandwidth of that short-duration feature will account for most of the total bandwidth.

10.1 Series of Non-Chirped Rectangular Pulses, Before and After Windowing

For a target at 500 m approaching at 100 m/s, the information on selecting valid pulse parameters in Section 8 allows us to set parameters for a series of rectangular non-chirped pulses. Equation (8.8) specifies that the PRI should lie between 3.3 and 75 microseconds, so we have used the average value of 39 microseconds, corresponding to a PRF of about 26 kHz. A range–Doppler plot with no windowing is shown at the left of Figure 14. The peak correlation occurs at range 510 metres and velocity -105 m/s.

Target distance:	500 m
Target recession velocity:	-100 m/s
Range-bin width:	$\Delta r_{\text{res}} = 150$ m
Pulse-train length:	$N = 32$ pulses per CPI
Pulse width:	$\tau = 1\ \mu$s
PRI (8.8):	$T = 39\ \mu$s (PRF = 26 kHz)
Blind range and max. unambiguous range (8.1):	150 m and 5846 m respectively
Max.-detectable speed and resolution (8.10), (8.11):	192 m/s, 12 m/s
Chirp factor (9.19):	$\mu = 0$ MHz^2 ($\tau_{\text{eff}} = 1\ \mu$s)
Sampling interval (9.22):	$t_s = 0.067\ \mu$s ($n_1 = 2$, $n_2 = 15$)

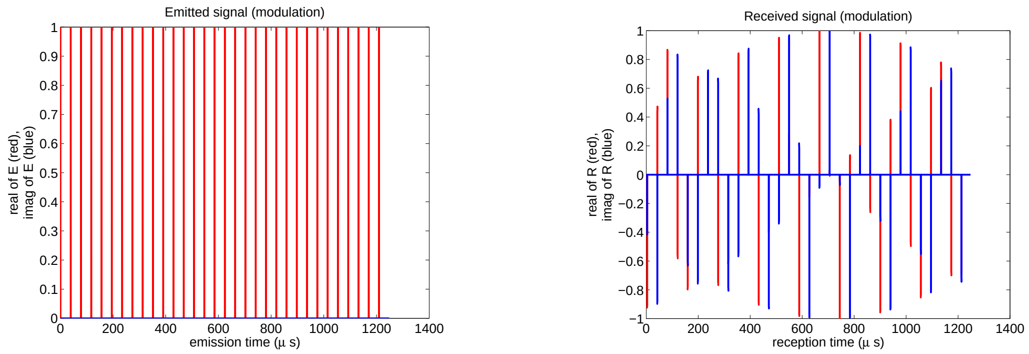
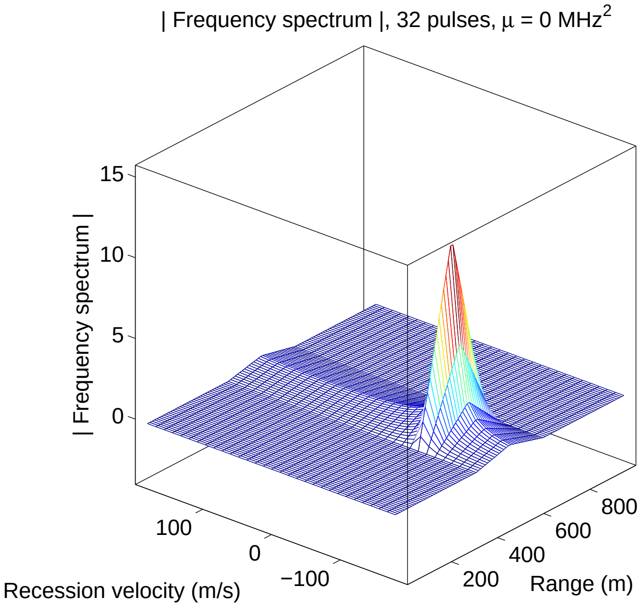


Figure 13: *A CPI of 32 pulses with range resolution of 10 metres, no chirp, PRI = 39 µs (PRF = 26 kHz), pulse width 1 µs, sampling interval 0.067 µs, and no Doppler windowing.* ***Left:*** *modulation of emitted signal.* ***Right:*** *received signal.*

At the bottom of Figure 14 is a range–Doppler plot made of the same target, but now incorporating Doppler windowing. The Doppler tail is now completely absent at the expense of a slight broadening of the Doppler peak. The peak correlation now occurs at range 500 metres. The velocity still peaks at -105 m/s.

This "tidying up" of the range–Doppler plot is the reason why windowing functions are useful, especially when windowing is used in the presence of background noise and other targets.



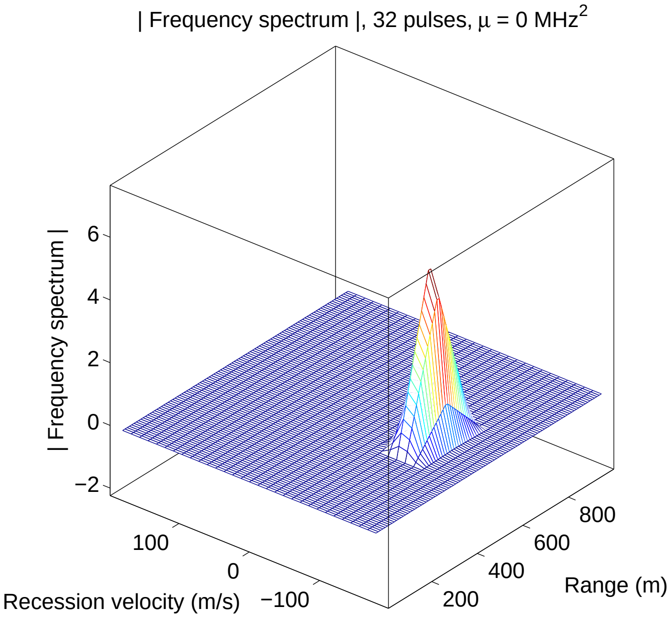


Figure 14: ***Top:*** *The CPI of 32 pulses in Figure 13 gives this range–Doppler plot. No Doppler windowing has been used.* ***Bottom:*** *The same scenario with Doppler windowing included.*

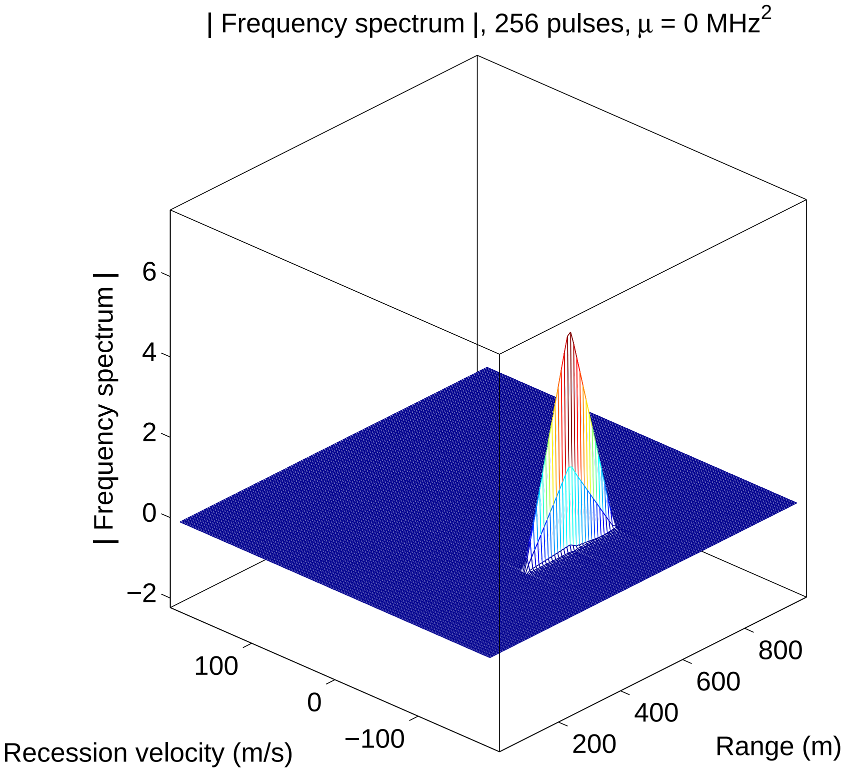


Figure 15: *A CPI with the same parameters used for Figure 14, with Doppler windowing, but now using 256 pulses. The higher number of pulses increases the Doppler resolution.*

Figure 15 shows range–Doppler for the same parameters used for Figure 14, including windowing, but now with 256 pulses per CPI. The range result is again 500 metres, but the higher number of pulses has increased the Doppler resolution, so that the velocity is now much more localised near its peak at -102 m/s.

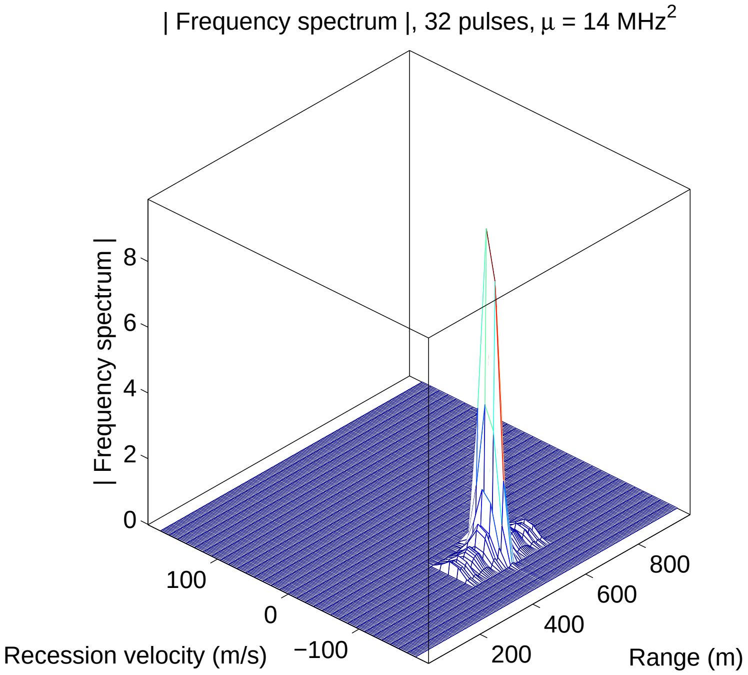
10.2 Series of Chirped Rectangular Pulses with Doppler Windowing

The parameters in the table below have been used to create a series of chirped pulses, again for a target at 500 m approaching at 100 m/s. Equation (8.8) specifies that the PRI should lie between 3.3 and 75 microseconds, so we have used the average value of 39 microseconds.

Target distance:	500 m
Target recession velocity:	-100 m/s
Range-bin width:	$\Delta r_{\text{res}} = 10$ m
Pulse-train length:	$N = 32$ pulses per CPI
Pulse width:	$\tau = 1\ \mu$s
PRI (8.8):	$T = 39\ \mu$s (PRF = 26 kHz)
Blind range and max. unamb. range:	150 m and 5846 m respectively
Max.-detectable speed and resolution (8.10), (8.11):	192 m/s, 12 m/s
Chirp factor (9.19):	$\mu = 14.0$ MHz^2 ($\tau_{\text{eff}} = 1/15\,\mu$s)
Sampling interval (9.22):	$t_s = 0.036\ \mu$s ($n_1 = 2$, $n_2 = 15$)

The effective pulse width is $\tau_{\text{eff}} = 1/15\,\mu$s, giving a chirp factor of $\mu = 14$ MHz^2. Some intuitive feel for the value of μ results from writing it as 14 MHz/μs. Equation (9.6) gives the carrier frequency of the emitted pulse as $f = f_0 + \mu t$, so for the 1 μs duration of the chirp the carrier frequency increases at a rate of $\mu = 14$ MHz/μs, giving a total frequency increase of 14 MHz. This is the swept bandwidth of one pulse. (This, of course, is just a restatement of (9.16), which gives the frequency change over a pulse width as $B_{\text{swept}} = |\mu|\tau$.) Compare this 14 MHz frequency increase with the carrier frequency of 10 GHz: the signal is indeed narrow band.

Figure 16 (top) shows the resulting range–Doppler plot. The peak correlation occurs at range of 498 metres and a velocity of -105 m/s. The chirp gives a better range resolution than the non-chirped pulse trains in Figures 14 and 15, although the fact that we don't quite get a peak at 500 metres is probably just due to a chance placement of the sampling. The bottom plot of Figure 16 shows the same scenario but now using 256 pulses, which gives better Doppler resolution.



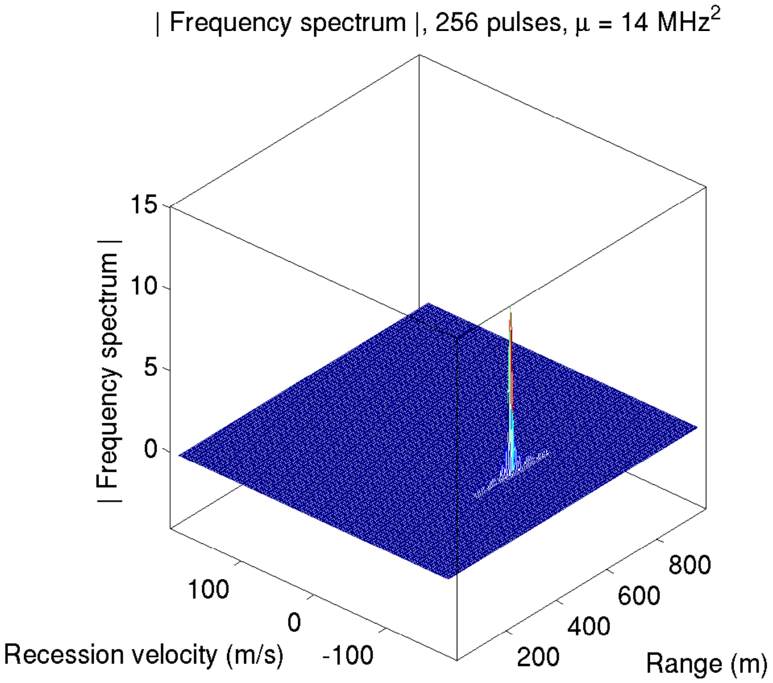


Figure 16: Top: *A CPI of 32 pulses with Doppler windowing, range resolution of 10 metres, chirp μ =14.0 MHz^2, PRI = 39 μs, pulse width 1 μs, sampling interval 0.036 μs. The chirp increases the range resolution.* ***Bottom:*** *The same for a CPI of 256 pulses, showing a clear increase in Doppler resolution.*

11 Creating Range and Velocity Offsets for Use in Jamming

A radar target might actively record the signal and re-emit a copy, perhaps altered, after some time interval. The alteration is accomplished by a set of electronics called a *tap*, which increases the signal strength (wave amplitude), and possibly introduces a time-dependent phase shift. We describe these alterations in this section. As in Section 2.3, the analytic signal emitted by the radar at time t is $u(t)\,\mathrm{e}^{\mathrm{i}\omega_0 t}$, and the target motion is also modelled as in that section. The question: how should the tap alter the signal to offset its range and velocity on the receiver's range–Doppler map?

11.1 Creating a Range Offset

The tap delay T_{tap} that gives a desired range offset is straightforward to compute. Delaying a tap's return by T_{tap} leads the radar to measure the tap to lie a distance of $\Delta r = cT_{\text{tap}}/2$ further away than its true value. So for any given Δr, the required delay is

$$\boxed{T_{\text{tap}} \simeq 2\Delta r/c\,.} \tag{11.1}$$

For example, if we wish the target to appear to be 150 metres further away than what it really is, set $\Delta r = 150$ m in (11.1) and infer that a tap must delay the signal for a time of

$$T_{\text{tap}} = \frac{2 \times 150 \text{ m}}{300 \text{ m}/\mu\text{s}} = 1\ \mu\text{s}\,. \tag{11.2}$$

11.2 Creating a Velocity Offset

A tap can introduce a velocity-offset structure to the range–Doppler plot by injecting a time-dependent phase change to each pulse it returns. We can see this in the following way. Refer to the third option of (2.22):

$$g_c(t) \simeq A(r_0)\,u(t-t_d)\,\mathrm{e}^{\mathrm{i}\omega_0 t}\,\mathrm{e}^{-\mathrm{i}\omega_0 \frac{2}{c}(r_0+vt)}. \tag{11.3}$$

Suppose that a tap multiplies each I/Q number it generates by $\mathrm{e}^{-\mathrm{i}\omega_0 2/c\,\Delta vt}$, which has the effect of replacing v in (11.3) with $v+\Delta v$. We then expect a new velocity of $v+\Delta v$ to be measured. In practice, of course, the tap has a clock of its own which is offset from the receiver's clock by some t_{offset}. So replace the factor of $\mathrm{e}^{-\mathrm{i}\omega_0 2/c\,\Delta vt}$ by

$$\mathrm{e}^{-\mathrm{i}\omega_0 \frac{2}{c}\,\Delta v(t\,+\,t_{\text{offset}})} = \mathrm{e}^{-\mathrm{i}\omega_0 \frac{2}{c}\,\Delta vt} \times \text{a constant phase factor.} \tag{11.4}$$

So this tap's clock offset produces a constant phase shift, which is of no real consequence in the radar signal processing. The additional Doppler shift detected by the radar that arises from Δv follows from (2.26), whose linearity implies that

$$\boxed{\Delta\omega_D = -\omega_0\,2\Delta v/c\,.} \tag{11.5}$$

Here we have a recipe for creating a single Doppler offset from a given tap. But in addition, although each tap is assumed capable of returning only one time-delayed signal to the radar for each pulse incident on it—producing a single major range offset—the tap is assumed to be capable of building this return by summing multiple altered copies of the incident pulse, where each alteration consists of an amplitude/phase variation made to the pulse. Each of these copies that are returned simultaneously will give rise to its own peak in the Doppler spectrum produced by the radar, with all of these peaks at the same range for a given tap.

Returning, say, two altered copies of the incident signal back to the radar from tap n, corresponding to velocity offsets Δv_1 and Δv_2, effectively requires multiplying one copy by $w_n^{(1)} \mathrm{e}^{-\mathrm{i}\omega_0 2/c\,\Delta v_1 t}$ $\big[= w_n^{(1)} \mathrm{e}^{\mathrm{i}\Delta\omega_{D,n}^{(1)} t}\big]$ and another copy by $w_n^{(2)} \mathrm{e}^{-\mathrm{i}\omega_0 2/c\,\Delta v_2 t}$ $\big[= w_n^{(2)} \mathrm{e}^{\mathrm{i}\Delta\omega_{D,n}^{(2)} t}\big]$ for weights $w_n^{(1)}$, $w_n^{(2)}$, and summing these two copies. This corresponds to multiplying the incident signal by the single number

$$w_n^{(1)} \mathrm{e}^{-\mathrm{i}\omega_0 2/c\,\Delta v_1 t} + w_n^{(2)} \mathrm{e}^{-\mathrm{i}\omega_0 2/c\,\Delta v_2 t} \quad \big[= w_n^{(1)} \mathrm{e}^{\mathrm{i}\Delta\omega_{D,n}^{(1)} t} + w_n^{(2)} \mathrm{e}^{\mathrm{i}\Delta\omega_{D,n}^{(2)} t}\big]\,. \tag{11.6}$$

This new factor becomes a generalised "Doppler offset" factor in the I/Q returned by the tap. It multiplies the right-hand sides of the expressions in (2.34) and (2.35). It needn't have unit magnitude. It will produce two peaks in the receiver's range–Doppler map at different Doppler values but at a single range value. Of course, in general there can be any number of terms in the phase sum. In fact we can synthesise an arbitrary Doppler profile for tap n by summing some possibly large number of $w_n^{(j)} \mathrm{e}^{\mathrm{i}\Delta\omega_{D,n}^{(j)} t}$ terms, although whether the tap is capable of generating this large number of terms depends on its hardware.

In practice, only absolute values of the Doppler spectral components are conventionally plotted in each range bin to build the range–Doppler map; phase information is discarded. But inserting arbitrary time-independent phases into the weights $w_n^{(j)}$ should make no difference to the synthesised range–Doppler plot, and indeed they do not, as noted after (11.10) ahead.

So if we require to synthesise the Doppler profile in range bin "n" of an already existing range–Doppler plot that was formed using, say, 32 pulses/CPI (and so has 32 velocity values plotted at each range), then we might be able to use 32 velocity offsets: the values of $\Delta\omega_{D,n}^{(1)}$ to $\Delta\omega_{D,n}^{(32)}$ will be those values corresponding to the velocities on the range–Doppler plot by way of (11.5), and their weights $w_n^{(1)}$ to $w_n^{(32)}$ will be proportional to the values being plotted on the "z axis" of the range–Doppler plot. But given that there is always some velocity broadening along the Doppler axis, we might well need to specify more than 32 velocity offsets to create the desired Doppler profile.

The Taps' Impulse Response

The ship's impulse response appeared in the calculation of Section 3.1. Here we show how to modify that calculation to model the taps' impulse response for including range and velocity offsets. In particular, the return is delayed as per (11.1) (now using a subscript n for tap n), so that a delayed return from tap n increments element number k_n of the impulse-response array $\boldsymbol{h}$, where k_n was calculated using (3.7) for no delay, but is now calculated from

$$k_n - 1 = \text{round}\left[\frac{2(r_n + \Delta r_n - R_{\text{near}})}{c t_s}\right]. \tag{11.7}$$

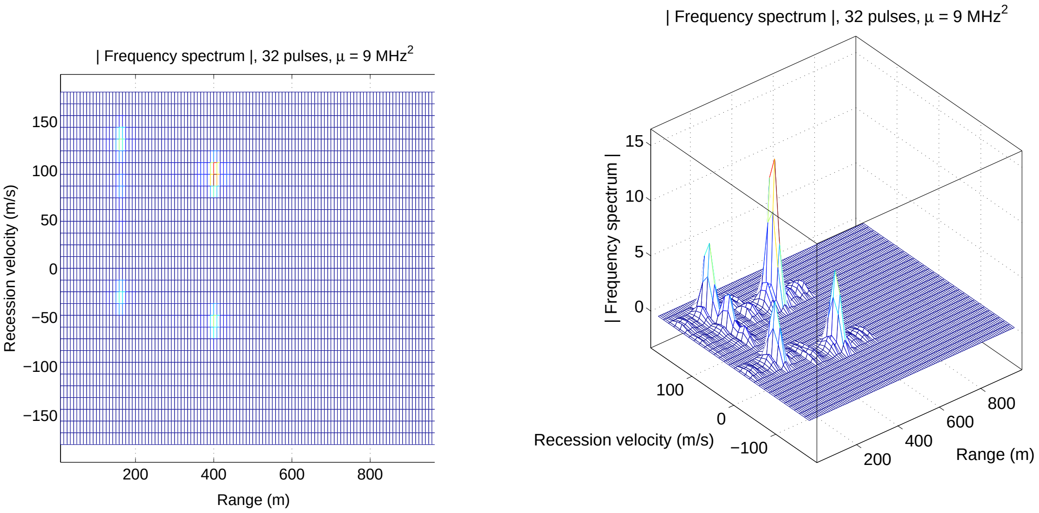


Figure 17: *A two-tap synthesis of two range bins, using the parameters in* (11.9) *and* (11.10)*. The first range bin has four velocity offsets and the second has two velocity offsets.*

The return itself now incorporates the weighted sum of Doppler offsets, such as in (11.6). For tap n, (3.8) becomes

$$\begin{aligned}\text{modulation of return} &= \frac{1}{4\pi r_n^2}\sqrt{\frac{2P_tG_t\sigma_n}{\varepsilon_0 c}}\; e^{i\phi_{0,n}}\; e^{-i\omega_0 \frac{2}{c}\times \text{range at emission}} \sum_j w_n^{(j)} e^{-i\omega_0 \frac{2}{c}\Delta v_j t} \\ &= \frac{1}{4\pi r_n^2}\sqrt{\frac{2P_tG_t\sigma_n}{\varepsilon_0 c}}\; e^{i\phi_{0,n}}\; e^{-i\omega_0 \frac{2}{c}\times \text{range at emission}} \sum_j w_n^{(j)} e^{i\Delta\omega_{D,n}^{(j)} t},\end{aligned} \tag{11.8}$$

where t is the time at the start of the current pulse being emitted.

11.3 Example of Synthesising Range and Doppler Offsets

Figure 17 shows an example of a range–Doppler profile built from a set of velocity offsets $\Delta v_n^{(j)}$ that give rise to corresponding $\Delta\omega_{D,n}^{(j)}$, using two taps, with Blackman windowing in Doppler, and incorporating an arbitrarily chosen signal-to-noise ratio of 20 dB, as shown in the following table. A 10 GHz carrier is used.

Target distance:	500 m
Target recession velocity:	0 m/s
Range-bin width:	$\Delta r_{\text{res}} = 15$ m
Pulse-train length:	$N = 32$ pulses per CPI
Pulse width:	$\tau = 1\ \mu$s
PRI (8.8):	$T = 40\ \mu$s (PRF = 25 kHz)
Blind range and max. unamb. range:	150 m and 5996 m respectively
Max.-detectable speed and resolution (8.10), (8.11):	187 m/s, 12 m/s
Chirp factor (9.19):	$\mu = 9$ MHz^2 ($\tau_{\text{eff}} = 1/10\ \mu$s)
Sampling interval (9.22):	$t_s = 0.1\ \mu$s ($n_1 = 2$, $n_2 = 10$)

Tap 1 offsets range by $\Delta r_1 = -340$ m from the nominal target range of 500 m, and has four Doppler offsets, at velocities of 135, 88, 50, and −24 m/s from the nominal target recession velocity of 0 m/s, using tap weights of 1, 0.5, 0.2, 0.8 respectively:

$$\begin{aligned} \Delta r_1 &= -340 \text{ m}, \\ \left\{\Delta v_1^{(1)},\ \Delta v_1^{(2)},\ \Delta v_1^{(3)},\ \Delta v_1^{(4)}\right\} &= \{135,\ 88,\ 50,\ -24\} \text{ m/s}, \\ \left\{w_1^{(1)},\ w_1^{(2)},\ w_1^{(3)},\ w_1^{(4)}\right\} &= \{1,\ 0.5,\ 0.2,\ 0.8\}\,. \end{aligned} \tag{11.9}$$

The parameters for tap 2 are

$$\begin{aligned} \Delta r_2 &= -100 \text{ m}, \\ \left\{\Delta v_2^{(1)},\ \Delta v_2^{(2)}\right\} &= \{-50,\ 100\} \text{ m/s}, \\ \left\{w_2^{(1)},\ w_2^{(2)}\right\} &= \{1,\ 2\}\,. \end{aligned} \tag{11.10}$$

As expected, if we attach a random time-independent phase to each of the tap weights, these plots are visually unaffected.

12 Final Comments and Acknowledgements

This report has addressed two aims. The first has been to act as a tutorial in the fundamentals of radar signal processing. My approach has been to start from the first principles of electromagnetic theory, build an expression for the I/Q signal that is returned by an arbitrary number of scatterers, then show how this yields range and Doppler information when correlated with the emitted signal. I have also laid out some of the theoretical reasons behind the various bounds placed on standard waveform parameters such as pulse width and chirp bandwidth.

In the appendices that follow, I have analysed the Hilbert transform to show where it comes from and why it's useful. My approach stresses how it can be constructed in a proper mathematical way without the arbitrariness found in the standard literature, where the transform is typically defined as a principal value for no apparent reason other than that this avoids a divergent integral. My approach does produce this same definition, but now the principal value arises quite naturally.

I have also explored a pen-and-paper approach to convolution and correlation in an appendix. The fact that "long multiplication" is actually a convolution is well known, but I hope the appendix gives additional insight, in that it employs a little-known form of long multiplication that matches the convolution routine far more closely than the usual "school" method of long multiplication does.

The second aim of this report has been to describe the mathematics of how a signal can be modified by a jammer so as to produce false targets in the radar receiver.

I wish to thank Roland Keir for his close involvement in the writing of this report. Its contents also benefitted from feedback and discussions with Gavin Dickeson, Len Hall, Stephen Howard, and Warren Marwood.

References

1. M.A. Richards (2005) *Fundamentals of Radar Signal Processing*, McGraw-Hill. See Section 1.3.3.

2. N. Levanon, E. Mozeson (2004) *Radar Signals*, John Wiley and Sons, Inc., New Jersey. See page 35, Properties 1 and 2.

3. M. Cheney, B. Borden (2009) *Fundamentals of Radar Imaging*, Society for Industrial and Applied Mathematics, Philadelphia. See pages 25–26.

4. D. Koks (2006) *Explorations in Mathematical Physics*, Springer, New York. See its equations (3.152) and (3.156) for the discrete Fourier notation, its equations (11.47) and (11.48) for the function sequences yielding $\pi\delta(t)$ and $\sigma(t)$ referred to in Appendix A, and its equation (11.46) for the one-sided Fourier identity referred to in that Appendix.

5. S.W. Smith (1999) *The Scientist and Engineer's Guide to Digital Signal Processing*, 2$^{\text{nd}}$ ed., California Technical Publishing, San Diego. See chapters 14 and 16.

6. D. Koks (2011), A Course of Lectures on Statistical Mechanics. DSTO-GD-0612, Melbourne, Vic., Defence Science and Technology Organisation (Australia). See Section 16.7.

This page is intentionally blank.

Appendix A Some Discussion of the Hilbert Transform

This appendix gives a derivation of the Hilbert transform with substantially more mathematical detail than typically found in radar texts. It is mostly my own analysis of why the Hilbert transform takes the form that it does. I have yet to find a radar textbook that discusses the Hilbert transform in a way that as a mathematical physicist I find illuminating; hence I have taken an approach here that does speak to the mathematical physicist. But additionally, the analysis here explains why the standard approach of radar texts works, when that approach "fixes" a divergent integral in an apparently ad hoc manner.

Given a real signal $s(t)$, we are generally not interested in its carrier; the information in the signal is contained in its running amplitude and phase. The amplitude and phase conventionally define the phasor representation of the signal. We will set $s(t)$ to be the real part of a *complex signal* $s_c(t)$, and then determine the choice of imaginary part that gives a phasor clearly embodying the signal's amplitude and phase. The resulting special choice of complex signal will be called the *analytic signal.*

Suppose, first, that our real signal is $s(t) = \cos 5t$. Set this to be the real part of $s_c(t)$; we desire to have $s_c(t) = \mathrm{e}^{\mathrm{i}5t}$, since this represents a phasor of unit length (corresponding to the unit amplitude of $\cos 5t$), and constant spin rate $\omega = 5$, corresponding to the angular frequency $\omega = 5$ of the real signal. An alternative choice is $s_c(t) = \mathrm{e}^{-\mathrm{i}5t}$, which also has real part $\cos 5t$. By convention, we'll set $s_c(t)$ to have positive frequency: $s_c(t) = \mathrm{e}^{\mathrm{i}5t}$. Notice that this contains *one* "phasor" frequency $\omega = 5$ with *unit weight.* Compare this to the real signal $s(t) = \cos 5t = {}^1\!/\!_2\, \mathrm{e}^{\mathrm{i}5t} + {}^1\!/\!_2\, \mathrm{e}^{-\mathrm{i}5t}$, which is built from *two* Fourier frequencies $\omega = \pm 5$, each with weight 1/2.

This construction of a useful complex signal from the real one is trivial in this case, but how might we go about it given a more complicated $s(t)$? After all, we can always write $s(t) = \cos 5t$ in arbitrary ways, such as

$$s(t) = A(t)\cos(3t + \sin t) = B(t)\cos(12t + \mathrm{e}^t) \tag{A1}$$

for some real $A(t)$ and $B(t)$, but that does *not* imply that it will be useful to define

$$s_c(t) = A(t)\exp \mathrm{i}(3t + \sin t)\,, \text{ or } s_c(t) = B(t)\exp \mathrm{i}(12t + \mathrm{e}^t)\,. \tag{A2}$$

That these two choices of $s_c(t)$ are generally not equal can be shown as follows. The real parts of the two choices are equal by construction, so we must show that the choices' imaginary parts are generally *not* equal. Do this by using (A1) to solve for $A(t)$ and $B(t)$:

$$A(t) = \frac{s(t)}{\cos(3t + \sin t)}\,, \quad B(t) = \frac{s(t)}{\cos(12t + \mathrm{e}^t)}\,. \tag{A3}$$

The imaginary parts of the two choices of $s_c(t)$ in (A2) must then be, from (A3),

$$A(t)\sin(3t + \sin t) = s(t)\tan(3t + \sin t)\,, \tag{A4}$$

and

$$B(t)\sin(12t + \mathrm{e}^t) = s(t)\tan(12t + \mathrm{e}^t)\,. \tag{A5}$$

These two imaginary parts are generally not equal, so the two choices of $s_c(t)$ in (A2) are also generally not equal. And because the functions $3t + \sin t$ and $12t + \mathrm{e}^t$ were arbitrarily chosen, we see that there are an infinite number of choices of $s_c(t)$ that all have the same real part $s(t)$, but which have different imaginary parts. The question is, which choice of $s_c(t)$ might be "best"?

Different choices of $s_c(t)$ will generally have frequency spectra that don't resemble the spectrum of $s(t) = \cos 5t$. We define the best choice of $s_c(t)$ for $s(t) = \cos 5t$ to be the standard phasor: $s_c(t) = \mathrm{e}^{\mathrm{i}5t}$. The frequency spectrum of this $s_c(t)$ has no negative frequencies, and has twice the amount of the positive frequency that appears in $s(t) = \cos 5t = 1/2\,\mathrm{e}^{\mathrm{i}5t} + 1/2\,\mathrm{e}^{-\mathrm{i}5t}$. This idea can now be applied to *any* real signal $s(t)$: choose $s_c(t)$ to be the function whose frequency spectrum has no negative frequencies, and whose positive frequencies are the same as those of the real signal $s(t)$, but with twice the weighting. Such a function is easily constructed: Fourier-transform $s(t)$ to give its spectrum, remove the negative part, double the positive part, and take the inverse Fourier transform, resulting in $s_c(t)$. This special choice of $s_c(t)$ is called the *analytic signal* corresponding to the real signal $s(t)$. In Fourier language, if the real signal $s(t)$ has spectrum $S(\omega)$:

$$s(t) = \int_{-\infty}^{\infty} S(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega\,, \tag{A6}$$

then the analytic signal is

$$s_c(t) = \int_{0}^{\infty} 2S(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega\,. \tag{A7}$$

Before evaluating this integral, we'll give an alternative discussion of the analytic signal that also leads to (A7).

A.1 Alternative Derivation of the Analytic Signal

Here is another approach to constructing a "best" choice of complex signal $s_c(t)$ that is not related to the discussion above. Given a real signal $s(t)$, suppose we construct a linear operator C that converts $s(t)$ to a complex signal by adding an imaginary part $\mathrm{i}\hat{s}(t)$, where $\hat{s}(t)$ is some special real function to be determined:

$$C\{s(t)\} \equiv s(t) + \mathrm{i}\hat{s}(t)\,. \tag{A8}$$

We require only that for *positive* frequencies Ω (we are taking a minimalist approach by dealing with positive frequencies only), C converts $\cos\Omega t$ and $\sin\Omega t$ into phasors, viz., the complex exponential $\mathrm{e}^{\mathrm{i}\Omega t}$, possibly with some scale factor. Specifically,

$$C\{\cos\Omega t\} \equiv \mathrm{e}^{\mathrm{i}\Omega t}\,, \tag{A9}$$

$$C\{\sin\Omega t\} = \alpha\mathrm{e}^{\mathrm{i}\Omega t} \text{ for some } \alpha\,. \tag{A10}$$

These three requirements (A8)–(A10) will suffice to determine C fully. Write $\alpha = a + \mathrm{i}b$ for a, b real, and apply (A8) to the left-hand side of (A10) to give

$$\sin\Omega t + \mathrm{i}\,\widehat{\sin\Omega t} = (a + \mathrm{i}b)(\cos\Omega t + \mathrm{i}\sin\Omega t)\,. \tag{A11}$$

Equating the real and imaginary parts of (A11) and making use of the linear independence of sine and cosine leads to $\alpha = -\mathrm{i}$, in which case $\widehat{\sin\Omega t} = -\cos\Omega t$. Thus

$$C\{\sin\Omega t\} = -\mathrm{i}\mathrm{e}^{\mathrm{i}\Omega t}\,. \tag{A12}$$

We now have enough information to determine how C acts on sinusoids of both positive *and* negative frequencies ω. Use the fact that cosine is an even function ($\cos\theta = \cos|\theta|$) and sine is odd ($\sin\theta = \operatorname{sgn}\theta\,\sin|\theta|$) to write

$$\begin{aligned} C\{\cos\omega t\} &= C\{\cos|\omega|t\} \overset{\text{(A9)}}{=\!=} \mathrm{e}^{\mathrm{i}|\omega|t}\,, \\ C\{\sin\omega t\} &= C\{\operatorname{sgn}\omega\,\sin|\omega|t\} \overset{\text{(A12)}}{=\!=} -\mathrm{i}\operatorname{sgn}\omega\,\mathrm{e}^{\mathrm{i}|\omega|t}\,, \end{aligned} \tag{A13}$$

and therefore, on adding the first equation above to $\mathrm{i}\times$ the second one,

$$C\{\mathrm{e}^{\mathrm{i}\omega t}\} = (1+\operatorname{sgn}\omega)\,\mathrm{e}^{\mathrm{i}|\omega|t} = 2\theta(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}\,, \tag{A14}$$

where $\theta(\omega)$ is the unit step function:[11]

$$\theta(\omega) \equiv {}^1\!/\!{}_2\,(1+\operatorname{sgn}\omega) = \begin{cases} 1\,, & \omega > 0 \\ 1/2\,, & \omega = 0 \\ 0\,, & \omega < 0 \end{cases}\,. \tag{A15}$$

We can now use the linearity of C to build $C\{s(t)\}$ for a general real signal $s(t)$. Begin with the Fourier decomposition (A6), writing

$$C\{s(t)\} = \int_{-\infty}^{\infty} S(\omega)\,C\{\mathrm{e}^{\mathrm{i}\omega t}\}\,\mathrm{d}\omega = \int_{-\infty}^{\infty} S(\omega)\,2\theta(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega = \int_0^{\infty} 2S(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega\,. \tag{A16}$$

But the last integral above is just the right-hand side of (A7), which shows that $C\{s(t)\}$ is identical to the analytic signal $s_c(t)$ of (A7). So this alternative approach to complexifying $s(t)$ has reproduced the analytic signal described in (A1)–(A7). We need now only evaluate (A7).

Notice that

$$\begin{aligned} \widehat{\cos\Omega t} &\overset{\text{(A9)}}{=\!=} \sin\Omega t = \cos(\Omega t - \pi/2)\,,\ \text{and} \\ \widehat{\sin\Omega t} &\overset{\text{(A12)}}{=\!=} -\cos\Omega t = \sin(\Omega t - \pi/2)\,. \end{aligned} \tag{A17}$$

This means that the imaginary part of the complex signal built from $\cos\Omega t$ is found by simply delaying $\cos\Omega t$ by 90°. The same is true for the imaginary part of the complex signal built from $\sin\Omega t$. Because Fourier analysis states that a signal can be considered as a sum of sines and cosines, we conclude that the complex part of the analytic signal is built by taking each of the sinusoidal components of the real signal and delaying them by 90°, or a quarter period.

A.2 Evaluating the One-Sided Fourier Integral (A7)

To evaluate (A7), begin by incorporating Fourier theory into the current analysis through writing (A7) as an integral from $-\infty$ to ∞:

$$\begin{aligned} s_c(t) &= \int_0^{\infty} 2S(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega = \int_{-\infty}^{\infty} \theta(\omega)\,2S(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega \overset{\text{(A15)}}{=\!=} \int_{-\infty}^{\infty} S(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}(1+\operatorname{sgn}\omega)\,\mathrm{d}\omega \\ &\overset{\text{(A6)}}{=\!=} s(t) + \int_{-\infty}^{\infty} S(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}\operatorname{sgn}\omega\,\mathrm{d}\omega\,. \end{aligned} \tag{A18}$$

[11] I use $\theta(0) = {}^1\!/\!{}_2$, corresponding to the choice $\operatorname{sgn} 0 = 0$. This aligns with the sense of continuity of Fourier sums within the Fourier transform theory of discontinuous functions.

The $\operatorname{sgn}\omega$ in the last integral of (A18) can be converted to another Fourier integral through the following piece of gymnastics. First, use the fact that

$$\int_{-\infty}^{\infty} \frac{\sin \omega\tau}{\tau}\,\mathrm{d}\tau = \pi \operatorname{sgn}\omega\,. \tag{A19}$$

Although $\frac{\sin \omega\tau}{\tau}$ is not defined for $\tau = 0$, it has a "removable discontinuity" there: a single "hole" in the curve depicting the function, as it were. You will sometimes see it written that $\frac{\sin x}{x} = 1$ when $x = 0$. But this is not so; rather, only the *limit* exists: $\lim_{x\to 0} \frac{\sin x}{x} = 1$, and distinguishing between these two statements is precisely why the concept of a limit was invented three centuries ago. The term "removable discontinuity" describes the fact that functions with isolated holes in their otherwise continuous curves still have well-defined integrals. So the presence of a single "hole" in the curve of the sinc function doesn't stop the integral (A19) from being well defined. But that means its *principal value* is also well defined:

$$⨍_{-\infty}^{\infty} \frac{\sin \omega\tau}{\tau}\,\mathrm{d}\tau = \pi \operatorname{sgn}\omega\,, \tag{A20}$$

where the principal value is defined as

$$⨍_{-\infty}^{\infty} f(x)\,\mathrm{d}x \equiv \lim_{\varepsilon\to 0^+} \left[\int_{-\infty}^{-\varepsilon} f(x)\,\mathrm{d}x + \int_{\varepsilon}^{\infty} f(x)\,\mathrm{d}x\right]. \tag{A21}$$

(For example, whereas $\int_{-\infty}^{\infty} \mathrm{d}x/x$ is undefined, the principal value $⨍_{-\infty}^{\infty} \mathrm{d}x/x$ equals zero because $1/x$ is an odd function.)

Writing (A19) in principal-value form (A20) allows us to introduce the following principal-value integral:

$$⨍_{-\infty}^{\infty} \frac{\cos \omega\tau}{\tau}\,\mathrm{d}\tau = 0 \tag{A22}$$

(for which the principal value is *required*) to conclude that

$$⨍_{-\infty}^{\infty} \frac{\mathrm{e}^{\mathrm{i}\omega\tau}}{\tau}\,\mathrm{d}\tau = ⨍_{-\infty}^{\infty} \frac{\cos \omega\tau + \mathrm{i}\sin\omega\tau}{\tau}\,\mathrm{d}\tau = \mathrm{i}\pi \operatorname{sgn}\omega\,. \tag{A23}$$

Equation (A23) is almost a Fourier integral—but only in the principal-value sense. We can eliminate the need for a principal value and convert (A23) fully to a Fourier integral by defining the following *continuous* generalised function:

$$\sigma(t) \equiv \begin{cases} 1/t & t \neq 0 \\ 0 & t = 0\,. \end{cases} \tag{A24}$$

Defining $\sigma(t)$ as continuous makes it a generalised function in the same class as the Dirac delta function, in that both of these functions can only be realised in practice as limiting cases of sequences of functions. Two such sequences are [4]

$$\pi\delta(t) = \lim_{\varepsilon\to 0^+} \frac{\varepsilon}{t^2+\varepsilon^2}\,, \quad \sigma(t) = \lim_{\varepsilon\to 0^+} \frac{t}{t^2+\varepsilon^2}\,, \tag{A25}$$

whose similarity reveals a close relationship between $\pi\delta(t)$ and $\sigma(t)$, which we'll see more of shortly.

Introducing $\sigma(t)$ enables us now to write the principal-value integral (A23) as a Fourier integral which is *not* a principal value, and which therefore can be brought into the fold of Fourier theory:

$$⨍_{-\infty}^{\infty} \frac{\mathrm{e}^{\mathrm{i}\omega\tau}}{\tau}\,\mathrm{d}\tau = \int_{-\infty}^{\infty} \mathrm{e}^{\mathrm{i}\omega\tau}\sigma(\tau)\,\mathrm{d}\tau \quad (= \mathrm{i}\pi \operatorname{sgn}\omega)\,. \tag{A26}$$

(This is an important point; the fact that $\sigma(t)$ is continuous and yet $\sigma(0) = 0$ is what converts the principal-value integral to a normal integral.) Now the last integral in (A18) can be written as (with $\mathrm{d}\omega$ and $\mathrm{d}\tau$ repositioned for readability)

$$\begin{aligned}\int_{-\infty}^{\infty} S(\omega)\, \mathrm{e}^{\mathrm{i}\omega t} \operatorname{sgn}\omega \,\mathrm{d}\omega &= \frac{1}{\mathrm{i}\pi}\int_{-\infty}^{\infty} \mathrm{d}\omega\, S(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{i}\pi \operatorname{sgn}\omega \\ &= \frac{1}{\mathrm{i}\pi}\int_{-\infty}^{\infty} \mathrm{d}\omega\, S(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}\int_{-\infty}^{\infty}\mathrm{d}\tau\,\mathrm{e}^{\mathrm{i}\omega\tau}\sigma(\tau)\,. \qquad \text{(A27)}\end{aligned}$$

Swapping the order of integration yields

$$\begin{aligned}\int_{-\infty}^{\infty} S(\omega)\, \mathrm{e}^{\mathrm{i}\omega t} \operatorname{sgn}\omega \,\mathrm{d}\omega &= \frac{1}{\mathrm{i}\pi}\int_{-\infty}^{\infty}\mathrm{d}\tau\,\sigma(\tau)\int_{-\infty}^{\infty}\mathrm{d}\omega\,S(\omega)\,\mathrm{e}^{\mathrm{i}\omega(t+\tau)} \\ &\overset{\text{(A6)}}{=\!=} \frac{1}{\mathrm{i}\pi}\int_{-\infty}^{\infty}\mathrm{d}\tau\,\sigma(\tau)\,s(t+\tau) \qquad \text{(A28)} \\ &= \frac{1}{\mathrm{i}\pi}\int_{-\infty}^{\infty}\mathrm{d}\tau\,\sigma(-\tau)\,s(t-\tau) \qquad \text{(A29)} \\ &= \frac{\mathrm{i}}{\pi}\int_{-\infty}^{\infty}\mathrm{d}\tau\,\sigma(\tau)\,s(t-\tau) \qquad \text{(A30)} \\ &= \frac{\mathrm{i}}{\pi}\sigma(t) * s(t)\,; \qquad \text{(A31)}\end{aligned}$$

here (A29) follows from (A28) by essentially just a change of variables $\tau \to -\tau$ (alternatively, this expresses the fact that the signed area under any function is unchanged if the function is reversed left–right); also, (A30) follows from (A29) because σ is an odd function; and (A31) is a convolution. Finally, equation (A18) becomes

$$s_c(t) = s(t) + \mathrm{i}\,\sigma(t)/\pi * s(t)\,, \qquad \text{(A32)}$$

which we compare with (A8) to write the *Hilbert transform* of the real signal as

$$\boxed{\mathscr{H}\{s(t)\} \equiv \hat{s}(t) = \sigma(t)/\pi * s(t)\,.} \qquad \text{(A33)}$$

On a side note, it's apparent that any one-sided Fourier transform can be written as a sum of a two-sided Fourier transform and a Hilbert transform. We can see this by writing (A8) as

$$s_c(t) = \big[1 + \mathrm{i}\mathscr{H}\big]\, s(t)\,, \quad \text{or} \quad {}^1\!/\!_2\, s_c(t) = {}^1\!/\!_2\,\big[1 + \mathrm{i}\mathscr{H}\big]\, s(t)\,; \qquad \text{(A34)}$$

now call on (A6) and (A7) to write the last expression above as

$$\boxed{\int_0^{\infty} S(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega = \frac{1}{2}\big[1 + \mathrm{i}\mathscr{H}\big]\int_{-\infty}^{\infty} S(\omega)\,\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega\,.} \qquad \text{(A35)}$$

Radar texts probably universally write the generalised function $\sigma(t)$ defined in (A24) as simply $1/t$. Although $1/t$ differs from $\sigma(t)$ only at a single point, there is a marked difference in using $1/t$ versus using $\sigma(t)$ in Fourier analysis: for example, whereas $\int_{-\infty}^{\infty}\sigma(t)\cos t\,\mathrm{d}t$ exists (and equals zero), $\int_{-\infty}^{\infty}\frac{\cos t}{t}\,\mathrm{d}t$ does *not* exist. Radar texts typically invoke the principal value to "fix" the divergence of an integral such as $\int_{-\infty}^{\infty}\frac{\cos t}{t}\,\mathrm{d}t$, but I don't believe that mathematics should be done by ad hoc fixes when something breaks; there is no a priori reason for why enforcing a symmetrical approach to a singularity in an integration should be a reasonable thing to do. That is, one should not invoke a principal value to "fix" a divergent integral—if an integral diverges, then we should conclude that we've done something wrong, and must go back to study the analysis in more detail.

In contrast, the approach taken in this appendix never invokes a principal value to "fix" a divergent integral, because it has no divergent integrals. Rather, the principal value was deliberately introduced in (A20) as a way of joining $\cos\omega\tau$ to $\sin\omega\tau$ to create a complex exponential that would bring the full power of Fourier theory to bear, enabling us to convert $\operatorname{sgn}\omega$ to a Fourier integral in (A27); this then allowed the one-sided Fourier integral (A7) to be calculated. There is nothing unusual about this use of a generalised function to perform a difficult integral; anyone who is familiar with the delta function $\delta(t)$ in Fourier theory should have no trouble accepting that it has a sister function, $\sigma(t)$. In fact, setting $S(\omega) = 1$ in (A35) gives

$$\begin{aligned}\int_0^\infty \mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega &= \frac{1}{2}\left[1 + \mathrm{i}\mathcal{H}\right] 2\pi\delta(t)\\ &= \pi\delta(t) + \mathrm{i}\,\sigma(t)/\pi * \pi\delta(t)\\ &= \pi\delta(t) + \mathrm{i}\,\sigma(t)\,, \end{aligned} \tag{A36}$$

showing that the two generalised functions appear side by side in this fundamental one-sided Fourier integral.[12] Equation (A36) also shows that $\sigma(t)$ is the Hilbert transform of $\pi\delta(t)$; again, a close relationship.

How do we invert the Hilbert transform? This can be done through the evaluation of $\sigma(t) * \sigma(t)$. First, write the Fourier transform $\mathcal{F}$ of some $f(t)$ as

$$\begin{aligned}\mathcal{F}\{f(t)\} &\equiv \int_{-\infty}^\infty \mathrm{e}^{\mathrm{i}\omega t} f(t)\,\mathrm{d}t = g(\omega)\,,\\ \text{so that }\ \mathcal{F}^{-1}\{g(\omega)\} &= \frac{1}{2\pi}\int_{-\infty}^\infty \mathrm{e}^{-\mathrm{i}\omega t} g(\omega)\,\mathrm{d}\omega = f(t)\,. \end{aligned} \tag{A37}$$

Now, using the idea that the Fourier transform of a convolution equals the product of the individual Fourier transforms, consider

$$\mathcal{F}\{\sigma(t) * \sigma(t)\} = \mathcal{F}\{\sigma(t)\} \times \mathcal{F}\{\sigma(t)\} \overset{\text{(A26)}}{=\!=\!=} (\mathrm{i}\pi \operatorname{sgn}\omega)^2 = \begin{cases} -\pi^2 & \omega \neq 0\\ 0 & \omega = 0\,. \end{cases} \tag{A38}$$

Next, invoke the inverse Fourier transform of both sides of (A38), noting that the removable discontinuity at $\omega = 0$ can be ignored for the purpose of doing the integral:

$$\sigma(t) * \sigma(t) = \mathcal{F}^{-1}\{-\pi^2\} \overset{\text{(A37)}}{=\!=\!=} \frac{-\pi^2}{2\pi}\int_{-\infty}^\infty \mathrm{e}^{-\mathrm{i}\omega t}\,\mathrm{d}\omega \;=\; -\pi^2\delta(t)\,. \tag{A39}$$

(This shows that $\sigma(t)/\pi$ is a "convolutional square root of -1", in the sense that $\delta(t)$ is to convolution what 1 is to multiplication: convolving any function with $\delta(t)$ leaves that function unchanged.) Now, the identity $\sigma(t) * \sigma(t) = -\pi^2\delta(t)$ enables us to invert the Hilbert transform easily. Consider the following, with "(t)" omitted for clarity:

$$\hat{s} = \mathcal{H}\{s\} = s * \sigma/\pi\,, \tag{A40}$$

[12] The identity (A36) was established by an entirely different route that used no Fourier transforms in Chapter 11 of [4].

so that

$$\begin{aligned}\mathcal{H}\{\hat{s}\} &= \hat{s} * \sigma/\pi \;=\; s * \sigma/\pi * \sigma/\pi \\ &= s * -\delta \;=\; -s \;=\; -\mathcal{H}^{-1}\{\hat{s}\}\,. \end{aligned} \tag{A41}$$

We conclude that $\mathcal{H}^{-1} = -\mathcal{H}$. Note also from this analysis that

$$\mathcal{H}\{\delta\} = \sigma/\pi\,, \quad \mathcal{H}\{\sigma/\pi\} = -\delta\,, \tag{A42}$$

which shows more of how the Hilbert transform connects these two generalised functions.

A.3 Numerical Evaluation of the Hilbert Transform

We can express the Hilbert transform as an integral over time in parallel streams here that swap the roles of s and σ, because convolution is commutative:

$$\begin{aligned} \hat{s}(t) &\equiv \sigma(t)/\pi * s(t) & &= s(t) * \sigma(t)/\pi \\ &= \frac{1}{\pi}\int_{-\infty}^{\infty} \sigma(\tau)\, s(t-\tau)\,\mathrm{d}\tau & &= \frac{1}{\pi}\int_{-\infty}^{\infty} s(\tau)\,\sigma(t-\tau)\,\mathrm{d}\tau \\ &= \frac{1}{\pi}\fint_{-\infty}^{\infty} \frac{s(t-\tau)}{\tau}\,\mathrm{d}\tau & &= \frac{1}{\pi}\fint_{-\infty}^{\infty} \frac{s(\tau)}{t-\tau}\,\mathrm{d}\tau\,. \end{aligned} \tag{A43}$$

These integrals are useful in theoretical work to calculate the analytic signal $s + \mathrm{i}\hat{s}$ for a simple real signal $s(t)$. But they can be difficult to apply numerically when $s(t)$ is more complicated. When transforming a signal from samples that have all been collected (i.e. we are *not* examining the problem of how best to transform in real time here), an easier approach is to Fourier-transform any of these integrals in (A43), recognising that they are convolutions, which converts them to a product of Fourier integrals, which is then inverse Fourier transformed. This procedure can use the discrete Fourier transform and its inverse, and is, in fact, completely identical to implementing the frequency-domain procedure that we used to *define* the analytic signal in the paragraph immediately preceding (A6) above.

To reiterate, to form the analytic signal $s_c(t)$ of a real signal $s(t)$, first Fourier-transform $s(t)$ to produce its spectrum. Then set the weights of all negative frequencies to zero, double the weights of all positive frequencies, and, to provide a sense of Fourier continuity, leave the weight of the zero frequency unchanged: this is equivalent to multiplying the spectrum by $\theta(\omega)$ using (A15). Finally inverse Fourier-transform this new spectrum to produce $s_c(t)$.

For a signal composed of a modulated sinusoid, the value of the modulation (i.e. envelope) at any time can be defined as the length of the phasor representing the complex number $s_c(t)$:

$$\text{envelope at time } t = |s_c(t)| = \sqrt{s^2(t) + \hat{s}^2(t)}\,. \tag{A44}$$

The phase $\phi(t)$ can be defined as the argument of the complex number $s_c(t)$:

$$\cos\phi(t) \equiv \frac{s(t)}{|s_c(t)|}\,, \quad \sin\phi(t) \equiv \frac{\hat{s}(t)}{|s_c(t)|}\,. \tag{A45}$$

As referred to in (7.6), at any moment, the instantaneous or "currently dominant" angular frequency is approximately $\phi'(t)$.

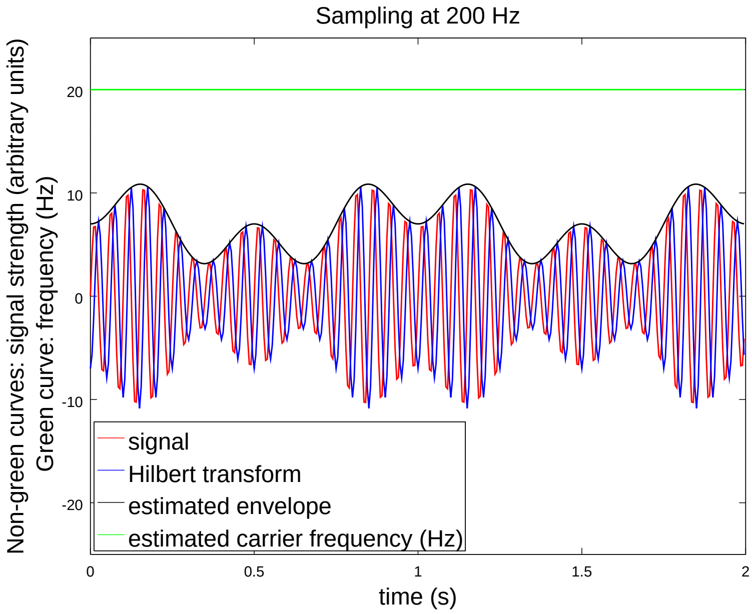


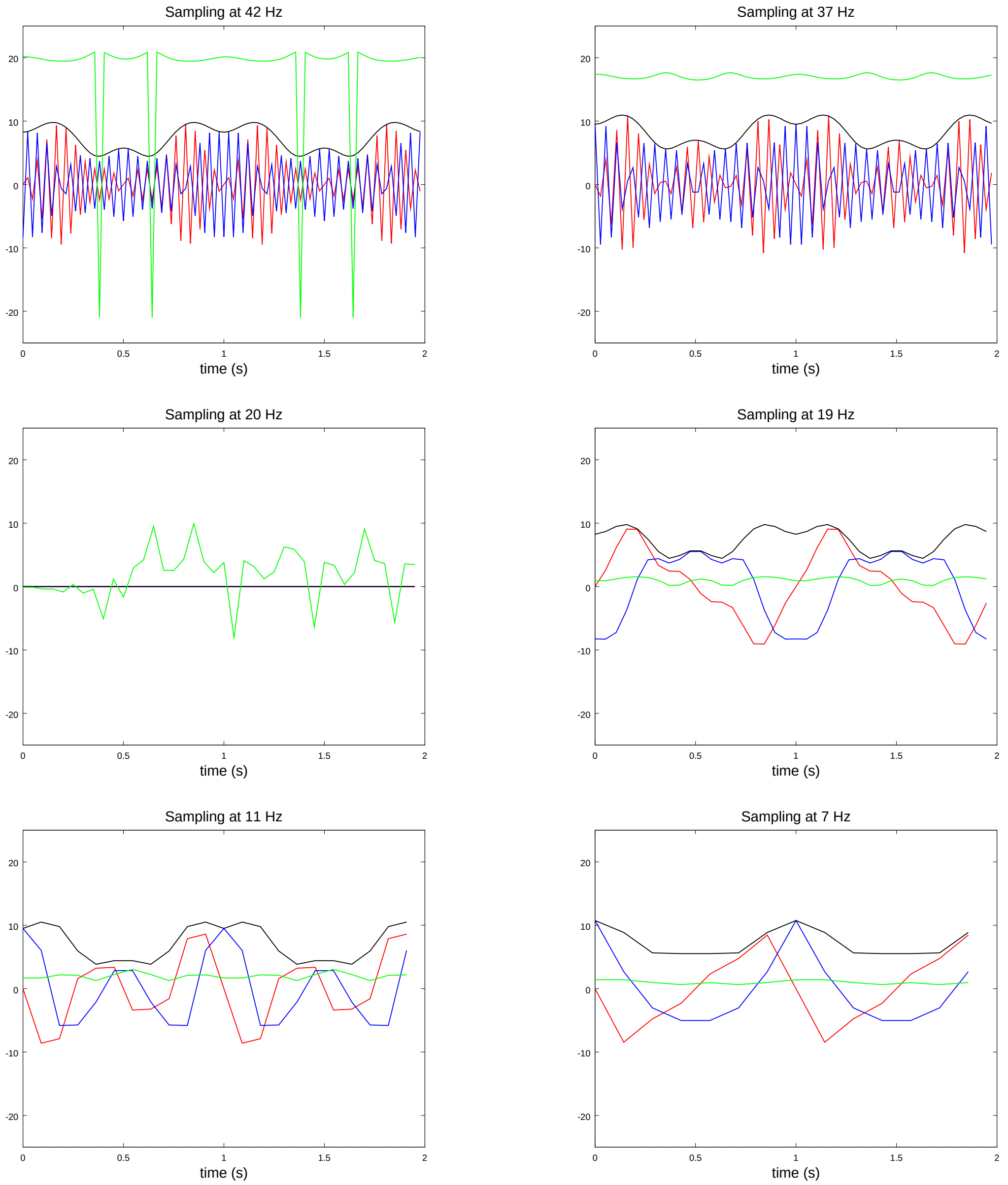
Figure A1: *An example of using the analytic signal to construct the envelope (black) of the real signal in* (A46) *(red), together with the dominant or running frequency at each time step (green, using the same y-axis for convenience but actually different units). The blue curve is the Hilbert transform of the red curve: we get a sense of the 90° lag referred to just after* (A17)*. The 200 Hz sampling rate gives a Nyquist frequency of 100 Hz, more than adequate to reveal the signal's structure.*

Examples of constructing the envelope and instantaneous frequency of a modulated signal are shown in Figures A1 and A2. We have used a combination of two modulation frequencies, 1 Hz and 2 Hz, on a carrier with frequency 20 Hz:

$$s(t) = \underbrace{(7 + 5\,\sin 2\pi t\,\sin 2\pi 2t)}_{\text{envelope}} \times \underbrace{\sin 2\pi 20t}_{\text{carrier}}\ . \tag{A46}$$

For Figure A1 we sample at 200 Hz, giving a 100 Hz Nyquist frequency that is of course well able to detect the 20 Hz carrier. The red curve in the plot is $s(t)$ and the blue curve is $\hat{s}(t)$, which is seen to lag the red curve by about 90°, consistent with the comment just after (A17). The black envelope is calculated as the modulus of $s_c(t)$ (A44) at each time, and clearly fits the signal (red) quite well. The phase is calculated from (A45) and then numerically differentiated: the phase increment at each time step is divided by the step's time increment to give the dominant frequency at that time, which is plotted in green (using the same y-axis for convenience but actually different units). As can be seen, this dominant frequency matches that of the 20 Hz carrier very closely.

Figure A2 shows results for various sampling rates that progressively reduce to well below carrier Nyquist. Although the carrier is generally not well detected, the modulation (black curve) is certainly well estimated, showing that the Hilbert transform is quite robust to lower-than-Nyquist sampling rates.



Sampling at 42 Hz

Sampling at 37 Hz

Sampling at 20 Hz

Sampling at 19 Hz

Sampling at 11 Hz

Sampling at 7 Hz

Figure A2: *Results for the signal of Figure A1, but now sampled at progressively lower frequencies. The labels and legends of these plots are the same as those of Figure A1 but are mostly omitted for clarity. Sampling at multiples of the carrier (such as the 20 Hz at mid left) gives no useful result.*

This page is intentionally blank.

Appendix B Tutorial on Correlation and Convolution

The process of correlating two sequences of numbers is central to much of this report, and a good understanding of it is necessary to follow the discussions that accompany equations such as (4.6). This appendix gives step-by-step descriptions of the processes of correlation and convolution. The two procedures are almost identical, so that a knowledge of one is easily transferred to the other. *Correlation* ("$\star$") detects the extent to which a sequence of numbers (the signal) is present inside a possibly longer sequence that is suspected to contain the signal together with noise. In contrast, *convolution* ("$*$") is a procedure that calculates the output of a linear time-shift-invariant system given some input, and is heavily involved in Fourier transform theory. Rewriting a correlation as a convolution enables us to search for one signal inside another, by using a standard technique for convolving quickly and efficiently that is based on the discrete Fourier transform.

Although correlation is defined as essentially convolution below, there is no reason that it has to be; convolution is simply the standard "matched filter" choice of correlation procedure, chosen because convolution turns out to maximise the ratio of what results from correlating something with a signal, compared to what results from correlating with the noise that is invariably attached to that signal.

B.1 The Correlation Procedure

Consider searching for a sequence $[-1,\ 2,\ 3]$, which constitutes a signal, within a sequence that might contain the signal together with some noise. We'll take this sequence to be $[0.1,\ 0.2,\ -1,\ 2.1,\ 3,\ 0.1]$.

Table B1 shows the steps of the procedure. It consists of moving $[-1,\ 2,\ 3]$ in from the left to overlap $[0.1,\ 0.2,\ -1,\ 2.1,\ 3,\ 0.1]$, one element at a time, and at each step summing the products of pairs of numbers that overlap: these pairs are shown in red. At each step, this sum is output as the next element of the correlation sequence. The process is identical to finding the euclidean dot product, at each step, of the two currently overlapping arrays of red numbers. We write the correlation as

$$\begin{aligned}&[-1,\ 2,\ 3]\ \star\ [0.1,\ 0.2,\ -1,\ 2.1,\ 3,\ 0.1]\ = \\ &\qquad\qquad [0.3,\ 0.8,\ -2.7,\ 4.1,\ 14.2,\ 4.2,\ -2.8,\ -0.1]\,. \end{aligned} \tag{B1}$$

The numbers to be correlated will generally be complex, and a complex conjugate must be used in the correlation process. The reason was given on page 17: only by using a complex conjugate will this procedure give a meaningful result: a peak when the signal matches up with itself. Note that by convention, no explicit conjugation is written with the "$\star$" notation. For example when correlating $[5,\ 6+2i,\ 3+8i]$ with $[1+i,\ 3,\ 5-4i,\ 7]$, we write the result as

$$[5,\ 6+2i,\ 3+8i]\ \star\ [1+i,\ 3,\ 5-4i,\ 7]\,, \tag{B2}$$

but the correlation is done by moving the *conjugated* sequence $[5,\ 6-2i,\ 3-8i]$ in from the left over $[1+i,\ 3,\ 5-4i,\ 7]$.

The correlation in (B1) has a maximum 14.2 as its fifth element. As can be seen from the steps of the procedure, it follows that the best estimate of where the signal sequence begins within the signal-plus-noise sequence is the third entry of the latter. Locating the best estimate like this is the idea behind the analysis of Section 4.2.

Table B1: *The array* [−1, 2, 3] *is correlated with* [0.1, 0.2, −1, 2.1, 3, 0.1] *through the following steps. The result is the bottom row of numbers* [0.3, 0.8, −2.7, 4.1, 14.2, 4.2, −2.8, −0.1]. *At each step, the number pairs to be multiplied are written in red.*

Correlation step 1:	−1	2	3							
			0.1	0.2	−1	2.1	3	0.1		
			0.3							
Correlation step 2:		−1	2	3						
			0.1	0.2	−1	2.1	3	0.1		
			0.3	0.8						
Correlation step 3:			−1	2	3					
			0.1	0.2	−1	2.1	3	0.1		
			0.3	0.8	−2.7					
Correlation step 4:				−1	2	3				
			0.1	0.2	−1	2.1	3	0.1		
			0.3	0.8	−2.7	4.1				
Correlation step 5:					−1	2	3			
			0.1	0.2	−1	2.1	3	0.1		
			0.3	0.8	−2.7	4.1	14.2			
Correlation step 6:						−1	2	3		
			0.1	0.2	−1	2.1	3	0.1		
			0.3	0.8	−2.7	4.1	14.2	4.2		
Correlation step 7:							−1	2	3	
			0.1	0.2	−1	2.1	3	0.1		
			0.3	0.8	−2.7	4.1	14.2	4.2	−2.8	
Correlation step 8:								−1	2	3
			0.1	0.2	−1	2.1	3	0.1		
The final result:			0.3	0.8	−2.7	4.1	14.2	4.2	−2.8	−0.1

B.2 The Convolution Procedure

The series of multiplications that implement a correlation ("$\star$") is in fact just a re-ordered version of the set that implement a convolution ("$*$"). Convolution is simply polynomial multiplication, and the way it relates to correlation is via (4.3):

$$\boldsymbol{A} \star \boldsymbol{B} = \boldsymbol{A}^\dagger * \boldsymbol{B}. \tag{B3}$$

There is no mystery here; correlation and convolution are just two different procedures that happen to be related by reversing and conjugating one of the sequences. Equation (B3) expresses the correlation in (B1) with a convolution:

$$[-1,\ 2,\ 3] \,\star\, [0.1,\ 0.2,\ -1,\ 2.1,\ 3,\ 0.1] = [3,\ 2,\ -1] \,*\, [0.1,\ 0.2,\ -1,\ 2.1,\ 3,\ 0.1]\,. \tag{B4}$$

Table B2 shows the steps followed in the convolution on the right-hand side of (B4). The result is

$$\begin{aligned} [3,\ 2,\ -1] \,*\, [0.1,\ 0.2,\ -1,\ 2.1,\ 3,\ 0.1] \ &= \\ &[0.3,\ 0.8,\ -2.7,\ 4.1,\ 14.2,\ 4.2,\ -2.8,\ -0.1]\,, \end{aligned} \tag{B5}$$

just as was obtained for the correlation in (B1).

We are working with real numbers only for convenience, but as an example of using complex numbers, (B2) becomes

$$\begin{aligned} [5,\ 6{+}2\mathrm{i},\ 3{+}8\mathrm{i}] \,\star\, [1{+}\mathrm{i},\ 3,\ 5{-}4\mathrm{i},\ 7] &= [3{-}8\mathrm{i},\ 6{-}2\mathrm{i},\ 5] \,*\, [1{+}\mathrm{i},\ 3,\ 5{-}4\mathrm{i},\ 7] \\ &= [11{-}5\mathrm{i},\ 17{-}20\mathrm{i},\ 6{-}53\mathrm{i},\ 58{-}90\mathrm{i},\ 67{-}34\mathrm{i},\ 35]\,. \end{aligned} \tag{B6}$$

The numbers in (B5) are, by design, the coefficients in the following polynomial multiplication:

$$\begin{aligned} (3 + 2z - z^2)\,(0.1 + 0.2z - z^2 + 2.1z^3 + 3z^4 + 0.1z^5) &= \\ 0.3 + 0.8z - 2.7z^2 + 4.1z^3 + 14.2z^4 &+ 4.2z^5 - 2.8z^6 - 0.1z^7\,. \end{aligned} \tag{B7}$$

These polynomials are defined to be the *z-transforms* of the original sequences (z is replaced by $1/z$ in an alternative definition of the transform). This means that convolving two sequences is equivalent to multiplying their z-transforms and *inverse* z-transforming the result, which converts the polynomial product back to a sequence of numbers—its coefficients. This procedure forms the *Convolution Theorem* for the z-transform. The polynomial formed from $\boldsymbol{A} \equiv \{A_0, A_1, A_2, \dots\}$ is the z-transform of $\boldsymbol{A}$:

$$\mathcal{Z}\{\boldsymbol{A}\} \equiv A_0 + A_1 z + A_2 z^2 + \dots \tag{B8}$$

and similarly for $\boldsymbol{B}$. The procedure in (B7) becomes

$$\mathcal{Z}\{\boldsymbol{A}\} \times \mathcal{Z}\{\boldsymbol{B}\} = \mathcal{Z}\{\boldsymbol{A} * \boldsymbol{B}\}\,, \tag{B9}$$

which is the Convolution Theorem. This correspondence between convolution and multiplication under some transform is a central theme of convolution theory.

The z that appears in the z-transform's polynomial is not normally given any value; it serves solely to convert a sequence to a function. But if we set z equal to 10 in the polynomials of (B7), the convolution becomes a straightforward base-10 multiplication—except that the digits in a base-10 multiplication are usually positive.

Table B2: *The array* [3, 2, −1] *is convolved with* [0.1, 0.2, −1, 2.1, 3, 0.1] *by the procedure below. The result of each step is shown in red under the solid line for that step, so that the last line becomes the convolution:* [0.3, 0.8, −2.7, 4.1, 14.2, 4.2, −2.8, −0.1]. *All numbers that are multiplied above the line—in pairs, from outside inwards—are written in red. We append 2 zeroes to the right-hand sequence (2 being one less than the length of the left-hand sequence); the only function of these is to facilitate the pairwise-multiplication procedure when using pen and paper.*

Convolution step 1:	3	2	−1	*	0.1	0.2	−1	2.1	3	0.1	0	0
					0.3	$(= 3 \times 0.1)$						
Convolution step 2:	3	2	−1	*	0.1	0.2	−1	2.1	3	0.1	0	0
					0.3	0.8	$(= 3 \times 0.2 + 2 \times 0.1)$					
Convolution step 3:	3	2	−1	*	0.1	0.2	−1	2.1	3	0.1	0	0
					0.3	0.8	−2.7	$(= 3 \times -1 + 2 \times 0.2 - 1 \times 0.1)$				
Convolution step 4:	3	2	−1	*	0.1	0.2	−1	2.1	3	0.1	0	0
					0.3	0.8	−2.7	4.1	$(= 3 \times 2.1 + \text{etc.})$			
Convolution step 5:	3	2	−1	*	0.1	0.2	−1	2.1	3	0.1	0	0
					0.3	0.8	−2.7	4.1	14.2			
Convolution step 6:	3	2	−1	*	0.1	0.2	−1	2.1	3	0.1	0	0
					0.3	0.8	−2.7	4.1	14.2	4.2		
Convolution step 7:	3	2	−1	*	0.1	0.2	−1	2.1	3	0.1	0	0
					0.3	0.8	−2.7	4.1	14.2	4.2	−2.8	
Convolution step 8:	3	2	−1	*	0.1	0.2	−1	2.1	3	0.1	0	0
				The final result:	0.3	0.8	−2.7	4.1	14.2	4.2	−2.8	−0.1

In fact, the digits needn't be positive; there is nothing wrong with writing e.g. 28 as $3\overline{2}$, meaning $30 - 2$, so that a multiplication such as 54×28 becomes $54 \times 3\overline{2} = 152\overline{8} = 1512$. Multiplication is often done this way by fast human mental calculators, because it converts the "difficult" multiplication by 8 into an easy multiplication by 2, with the only price being a minor amount of subtraction needed.

The carrying step necessary for base-10 multiplication is not a part of the convolution of course; rather, it's simply a consequence of the fact that any base-n system uses single digits to represent numbers less than n.

Conventionally of course, the digits making up a base-10 number are written in the reverse direction to that of the coefficients in (B7), so that the convolution that comprises a base-10 multiplication ends up being done from right to left. You can see this in more detail when multiplying 1234 by 567, following the steps in Table B3. We first express the numbers as sequences in base-10 by z-transforming them (with, effectively, z set equal to 10, although we are so familiar with mentally equating a number with its "10-transform" that we might not be explicitly aware of doing this step). Then we convolve the sequences, and finally we inverse z-transform the resulting sequence back to a number—again, a step that we might not explicitly be aware of doing, since we take for granted the actually deep notion of equality between a number and its base-10 representation. In other words, we multiply numbers—in principle a difficult task that involves lots of counting—by convolving their "10-transforms", which is a much easier task. We see then, that what we're accustomed to think of as multiplying two numbers written with digits is in fact a convolution of those digits—with added "carrying steps" that are necessary when using a finite set of digits to represent the numbers.

Convolution and the Discrete Fourier Transform

Equation (B9) holds fairly trivially. A similar expression results when the z-transform is replaced by the discrete Fourier transform $\mathcal{D}$, which transforms a sequence of complex numbers to another sequence of the same length. If the sequences $\boldsymbol{A}$ and $\boldsymbol{B}$ have the same length, then the Convolution Theorem states

$$\mathcal{D}\{\boldsymbol{A}\} \times \mathcal{D}\{\boldsymbol{B}\} = \mathcal{D}\{\boldsymbol{A} * \boldsymbol{B}\}\,, \tag{B10}$$

where the multiplication "×" of sequences $\boldsymbol{P}$ and $\boldsymbol{Q}$ is defined *pointwise*: $(\boldsymbol{P} \times \boldsymbol{Q})_n \equiv P_n Q_n$. When $\boldsymbol{A}$ and $\boldsymbol{B}$ have different lengths, a slight alteration to (B10) is needed. Their convolution $\boldsymbol{A} * \boldsymbol{B}$ has length $L = \text{length}(\boldsymbol{A}) + \text{length}(\boldsymbol{B}) - 1$. Append zeroes to $\boldsymbol{A}$ and $\boldsymbol{B}$ to make new arrays $\boldsymbol{A}'$ and $\boldsymbol{B}'$ respectively that both have length L. (Note, this is *not* related to the mostly outdated practice of zero padding, discussed ahead in Appendix D.) With the definition of the DFT used in this report [equations (6.1), (6.2), and Table C1 in Appendix C], the Convolution Theorem becomes

$$L\mathcal{D}\{\boldsymbol{A}'\} \times \mathcal{D}\{\boldsymbol{B}'\} = \mathcal{D}\{\boldsymbol{A} * \boldsymbol{B}\}\,. \tag{B11}$$

The Matlab functions `fft` and `ifft` differ in their normalisation from my `Dft` and `DftInverse` of Table C1; Matlab's normalisation makes (B11) slightly simpler:

$$\mathcal{D}_{\text{Matlab}}\{\boldsymbol{A}'\} \times \mathcal{D}_{\text{Matlab}}\{\boldsymbol{B}'\} = \mathcal{D}_{\text{Matlab}}\{\boldsymbol{A} * \boldsymbol{B}\}\,. \tag{B12}$$

On the other hand, `fft` doesn't immediately give the correct peak heights to spectra made from it, whereas `Dft` does. Always check the normalisation used in your own choice of Fast Fourier Transform routine.

Table B3: *Multiplying 1234 × 567 to get 699,678 by recognising that the process is a convolution plus the mechanism of "carrying" which is necessary to represent the result in a base-10 system. We prepend the left-hand factor with a number of zeroes that is one less than the number of digits in the right-hand factor. This procedure is the mirror image of the procedure in Table B2.*

Convolution step 1:	0	0	1	2	3	4	×	5	6	7
					$4 \times 7 = 28$:	$^{2}8$				
Convolution step 2:	0	0	1	2	3	4	×	5	6	7
				$3 \times 7 + 4 \times 6 +$ carried $2 = 47$:	$^{4}7$	8				
Convolution step 3:	0	0	1	2	3	4	×	5	6	7
			$2 \times 7 + 3 \times 6 + 4 \times 5 +$ carried $4 = 56$:	$^{5}6$	7	8				
Convolution step 4:	0	0	1	2	3	4	×	5	6	7
		$1 \times 7 +$ etc. $= 39$:	$^{3}9$	6	7	8				
Convolution step 5:	0	0	1	2	3	4	×	5	6	7
	$0 \times 7 +$ etc. $= 19$:	$^{1}9$	9	6	7	8				
Convolution step 6:	0	0	1	2	3	4	×	5	6	7
The final answer:	6	9	9	6	7	8				

Equation (B11) is a highly efficient method for convolving $\boldsymbol{A}$ and $\boldsymbol{B}$. For example, we can use it to reproduce the result from Table B2 with this Matlab code:

```
A = [3 2 -1];
B = [0.1 0.2 -1 2.1 3 0.1];
A_prime = [A zeros(1,length(B)-1)];
B_prime = [B zeros(1,length(A)-1)];
(length(A)+length(B)-1) * DftInverse( Dft(A_prime) .* Dft(B_prime) )
% The above line is identical to
ifft( fft(A_prime) .* fft(B_prime) )
```

We get the expected result of

```
[0.3  0.8  -2.7  4.1  14.2  4.2  -2.8  -0.1]
```

For very long arrays, convolution never uses the expensive procedure of Table B2. Instead, it always uses the Convolution Theorem, specifically (B11) or (B12). But understanding the procedure of Table B2 will make you a very adept convolver.

On a final note, this replacing of a difficult *convolution* by an easy *pointwise multiplication* via the discrete Fourier transform is the reverse idea to the multiplication example in Table B3. When learning arithmetic, we effectively replace difficult *multiplication* by easy *convolution* via a "10-transform"—although few would ever describe it that way.

Appendix C Sample Discrete Fourier Transform Code

Here we list two examples of Matlab and Mathematica code to implement a DFT. Our convention is to plot only frequencies with absolute value less than the Nyquist frequency; thus negative frequencies are plotted as negative frequencies, and not wrapped around to become positive frequencies greater than the Nyquist frequency. In particular, our convention allows a plot of the DFT of a gaussian function to be another gaussian. This mimics the fact that the continuous Fourier transform of a gaussian function is another gaussian.

Table C1 shows an example of Matlab code that implements the DFT in (6.1) and inverse DFT in (6.2). It might appear that the code here for a matrix could also be used for an array, thus eliminating the need for the "`if`–`else`" block. But that isn't so, because the ", 1" in the `fftshift` command is really only appropriate for matrices; when used on an array, it gives a different result for a row array than for a column array.

Table C2 has an example of Mathematica code that does the same job.

Table C1: *An example of Matlab code that implements the DFT in* (6.1) *and inverse DFT in* (6.2). *See the text above for the use of ", 1" in the* `fftshift` *command.*

The DFT can be implemented in Matlab with

```
function X = Dft(x)

% If x is an array:
if isvector(x) == 1
  X = fftshift(fft(x))/length(x);
else
  % x is a matrix, so act on each column separately.
  X = fftshift(fft(x),1)/size(x,1);
end
```

The inverse DFT can be implemented with

```
function x = DftInverse(X)

% If X is an array:
if isvector(X) == 1
  x = ifft(ifftshift(X)) * length(X);
else
  % X is a matrix, so act on each column separately.
  x = ifft(ifftshift(X,1)) * size(X,1);
end
```

Table C2: *An example of Mathematica code that implements the DFT in* (6.1) *and inverse DFT in* (6.2)

```
(* These do what Matlab's fftshift and ifftshift do.
For x any list,
   Fftshift[Ifftshift[x]] = Ifftshift[Fftshift[x]] = x. *)
Fftshift[x_List]  := ( NN = Length[x];
      Flatten[{Take[x, -Floor[NN/2]], Take[x, Ceiling[NN/2]]}] )
Ifftshift[x_List] := ( NN = Length[x];
      Flatten[{Take[x, -Ceiling[NN/2]], Take[x, Floor[NN/2]]}] )

Begin["Private`"]
 (* These do what my Matlab functions Dft and DftInverse do,
    but just on an array.  They are defined them here solely
    for use in building Dft and DftInverse in a moment.
    For x any list, DftInverseVector[DftVector[x]] =
       DftVector[DftInverseVector[x]] = x.*)
  DftVector[x_List] := Fftshift[Fourier[x,
                                  FourierParameters -> {-1, -1}]]
  DftInverseVector[X_List] := InverseFourier[Ifftshift[X],
                                  FourierParameters -> {-1, -1}]
End[] (* "Private`" *)


(* These do what my Matlab functions Dft and DftInverse do,
   on arrays and on matrices. *)
Dft[x_List] :=
 If[Length[Dimensions[x]] == 1,
   (* x is an array *)
   Private`DftVector[x],
   (* else x is a matrix *)
   Transpose[
    Table[Private`DftVector[ x[[All,i]] ], {i, Length[ x[[1]] ]}]
   ]
 ]

DftInverse[X_List] :=
  If[Length[Dimensions[X]] == 1,
    (* X is an array *)
    Private`DftInverseVector[X],
    (* else X is a matrix *)
    Transpose[
     Table[Private`DftInverseVector[ X[[All,i]] ],
          {i, Length[ X[[1]] ]}]
    ]
  ]
```

Appendix D Zero Padding in the Discrete Fourier Transform

The appending with zeroes that was required in the analysis around (B11) to convolve two sequences with the DFT is an exact procedure: it gives the exact same result that's obtained when we convolve the two sequences using the approach of Table B2. This appending is distinct from the historical zero padding of data mentioned in Section 6. Older implementations of the FFT *required* the length of the data set to be a power of 2, and so a data set with length other than this would have zeroes appended to create one whose length was a power of 2. This zero padding changed the data and so introduced spurious frequencies to the spectrum. These unwanted frequencies were tolerated because only by zero padding to a power-of-2 length could the FFT be used at all. But current FFT algorithms are very fast even when the data length is not a power of 2, so this zero padding is no longer necessary.

To see why zero padding changes a Fourier transform, recall the central fact that the DFT is a set of amplitudes of sinusoids. Because sinusoids are periodic, the DFT effectively "assumes" that the sequence of numbers being transformed is periodic. The result is that the corresponding Fourier series replicates the original sequence endlessly forward and backward in time. For example, the 5-element sequence $[1, 2, 3, 4, 5]$ is interpreted by the DFT to be the infinite data set

$$\ldots 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, \ldots \tag{D1}$$

What we mean here is that the Fourier series that returns the numbers $[1, 2, 3, 4, 5]$ as a function of times equal to, say, $1, 2, \ldots, 5$ will be the same as the series that returns the numbers $[1, 2, 3, 4, 5, 1, 2, 3, 4, 5]$ as a function of times $1, 2, \ldots, 10$. The DFTs of these two sequences differ only by the presence of interspersed zeroes in the DFT of the second sequence; but those zeroes are weights of sinusoids, and so the end result is that both sequences give rise to the same non-zero sinusoids: the same Fourier series. (Verify this using `Dft`—but not `fft`, which doesn't normalise appropriately.) But zero padding $[1, 2, 3, 4, 5]$ to the next power-of-2 length alters it to $[1, 2, 3, 4, 5, 0, 0, 0]$, which is then interpreted by the DFT as

$$\ldots 1, 2, 3, 4, 5, 0, 0, 0, 1, 2, 3, 4, 5, 0, 0, 0, 1, 2, 3, 4, 5, 0, 0, 0, \ldots \tag{D2}$$

The spectrum of this sequence is certainly different to that of (D1). In essence, the triplets of zeroes in (D2) are treated as a constant stream of data that requires comparatively many frequencies to generate it, because many sinusoids are required to cancel each other to produce a "zero-slope" sequence. The spectra generated from $[1, 2, 3, 4, 5]$ and $[1, 2, 3, 4, 5, 0, 0, 0]$ are shown in Figure D1 on the following page. See the end of this appendix for some discussion of the right-hand plot.

Some users of the DFT object that the DFT algorithm effectively replicates their data into the past and future. They wish to let values of zero represent unknown values of the system under observation, and so will append many zeroes to their data before Fourier transforming it. The length of the padded sequence need not be a power of 2, so this zero padding is not related to the historical zero padding required by early FFT algorithms. When this zero-padded sequence is effectively replicated by the DFT, the zeroes have the effect of pushing the data replications very far into the past and future from the sequence

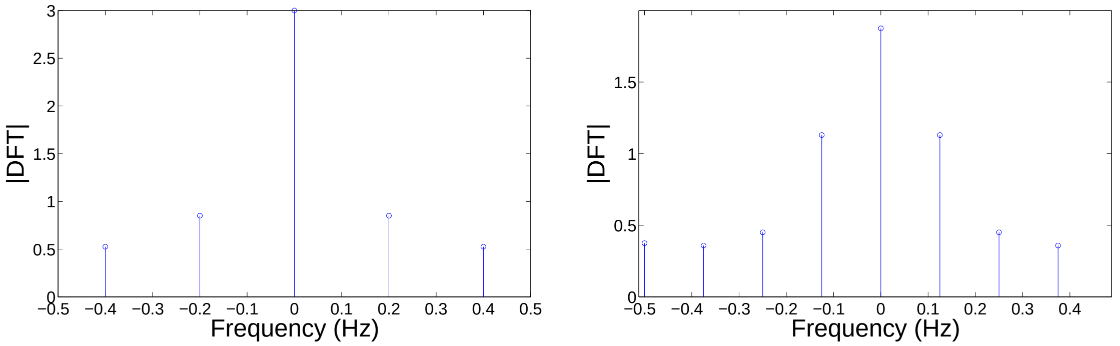


Figure D1: ***Left:*** *Spectrum of* $[1, 2, 3, 4, 5]$, *sampled once per second.* ***Right:*** *Spectrum of* $[1, 2, 3, 4, 5, 0, 0, 0]$, *sampled once per second. More and different frequencies are necessary to construct the Fourier series that represents the second data set.*

that is "located near the present", so to speak. But there is no reason why this zero estimate of what was never measured should be any more realistic than the replicating alternative. Indeed, if we really don't want the Fourier transform to treat our data as periodic, then perhaps we shouldn't be using a Fourier transform in the first place.

Nevertheless, suppose we append 1000 zeroes to $[1, 2, 3, 4, 5]$ and then transform the resulting long array. The resulting spectrum is shown in Figure D2. (The high density of plot points in that figure obliges us to omit the stems that were used to highlight the plot points of Figure D1.) The amplitudes being plotted are now quite small. This is reasonable, given that each amplitude is the amount present of its corresponding sinusoid, and the more sinusoids we have, the less we'll generally need of each.

You can think of zero-padding data as equivalent to multiplying a longer set of data by a top-hat function that zeroes the last segment of the data. Fourier-transforming this product of data and top hat is identical to convolving the Fourier transforms of the data and top hat. The top hat transforms to a sinc function, so the effect is to convolve the correct spectrum with a sinc. This smears the spectrum out, introducing the spurious frequencies mentioned at the start of this appendix. (But remember that this smearing

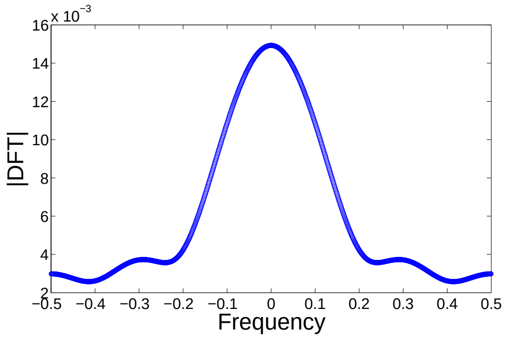


Figure D2: *The DFT of the array* $[1, 2, 3, 4, 5, 0, \ldots, 0]$, *where 1000 zeroes are appended after the 5. The spectrum is still discrete, but the "stems" that are used in Figure D1 are omitted here.*

has no relevance if we are using the FFT as a fast way of convolving two different-length arrays.)

Zero padding is sometimes described as a way of returning a finer or better frequency resolution. But the DFT cannot work magic: it cannot somehow go back out into the field and return a whole new set of data points. Zero padding smoothens the spectrum with a sinc function and returns a denser set of frequencies, but that doesn't imply that these frequencies actually exist in the data.

You can find zero padding described in the literature as relatively harmless because the zeroes are "not new data" as they "don't contribute to any sums in the FFT". But, of course, the FFT doesn't know which of its input numbers are data and which are not data; it simply processes an array that it can only treat uniformly. So the padded zeroes effectively become new data—but false data. And that the zeroes don't contribute to any sums is quite irrelevant. The FFT requires as input the total number of data points, and that number certainly increases when zeroes are appended, which changes the FFT output non-trivially. Analogously, if we were to form a sum to calculate the mean of that data and then do the same with appended zeroes included, the zeroes wouldn't contribute to that sum either—but they certainly *would* change the mean, as indeed they should.

Another way of seeing that zero padding has no a priori validity in building a spectrum is to consider two researchers who independently collect identical data: say 100 numbers. Their FFT algorithm requires its data set to have length a power of 2, so Researcher A pads with 28 zeroes. Researcher B, on the other hand, collects 28 more data points and is astonished to find that they're all zeroes. A and B will end up Fourier transforming identical sets of 128 numbers. But B has measured something that A has not, and so B should surely produce a spectrum with more information than that obtained by A, and yet does not; instead, Researcher A has by sheer coincidence or luck produced the exact same spectrum obtained by B.

The acid test of any FFT algorithm is "Sample one or more noiseless sinusoids in the way described for Figure 5 on page 26, and then plot the spectrum produced by the FFT. This spectrum should be non-zero *only* for each of the frequencies used, as in Figure 5". The reason why no extra frequencies should appear is because that procedure samples the sinusoids in such a way that when the DFT effectively strings copies of the data together, what results still represents one or more pure sinusoids. (More generally we might choose fractional frequencies, but the sampling required to preserve each as a single sinusoid in the replication becomes more delicate then.)

For example, in Matlab's notation, choose a sampling frequency `sf` that is a natural number (i.e. anything in $1, 2, 3, \ldots$) and form a set of data that oscillates at, say, 4 Hz:

```
data = sin(2*pi * 4 * linspace(0, 1-1/sf, sf))
```

where the sampling frequency is above Nyquist's 8 Hz. (The sampling frequency must be a natural number in this case to ensure that Matlab builds an array of samples precisely corresponding to each time in the array `0 : 1/sf : 1-1/sf`. That is, the final value `1-1/sf` must equal a natural number of increments `1/sf`.) Yes, the sampling required by this test is quite artificial, but that has no bearing on the use of this test to ascertain whether an FFT algorithm works or not. The FFT of `data` should take on the value of exactly 0.5 at the frequencies of exactly 4 and -4, and be *exactly zero* for all other frequencies. Zero-padding `data` before FFT-ing it will fail this test. (Strangely, some zero-padders will insist that the two frequencies here of exactly 4 and -4 are incorrect. But those frequencies are certainly correct because they represent a sine function of frequency exactly 4 Hz.)

A final comment: the right-hand spectrum in Figure D1 includes the negative Nyquist frequency (-0.5) but not the positive one. Is this correct? It certainly is; the DFT has 8 elements here, and because the zero frequency is *always* present for both even- and odd-length data sets, only the negative Nyquist frequency will be represented in the DFT when the data has even length. But the weighting of this frequency includes the weighting of the positive Nyquist frequency. For a closer inspection of this figure, use the DFT (6.1) and its inverse (6.2). The array of data is $[1, 2, 3, 4, 5, 0, 0, 0]$. Call these data elements x_k for $k = 0$ to 7 respectively. The DFT of this array is $[X_{-4}, X_{-3}, \ldots, X_3]$, whose elements are respectively

$$\begin{gathered} 0.375, \quad -0.3232 + 0.1553\,\mathrm{i}, \quad 0.375 - 0.25\,\mathrm{i}, \quad -0.6768 + 0.9053\,\mathrm{i}, \quad 1.875, \\ -0.6768 - 0.9053\,\mathrm{i}, \quad 0.375 + 0.25\,\mathrm{i}, \quad -0.3232 - 0.1553\,\mathrm{i}\,. \end{gathered} \tag{D3}$$

The absolute values of these numbers are plotted in the right-hand spectrum of Figure D1, versus their frequencies, which are the set of n/N in (6.1), namely $-4/8, -3/8, -2/8, \ldots, 3/8$. These frequencies now insert into (6.2) to give the inverse transform, for $k = 0$ to 7:

$$\begin{aligned} x_k &= \sum_{n=-4}^{3} X_n\, \mathrm{e}^{\mathrm{i}2\pi kn/N} \\ &= 0.375\,\mathrm{e}^{\frac{-4}{8}\mathrm{i}2\pi k} + (-0.3232 + 0.1553\,\mathrm{i})\,\mathrm{e}^{\frac{-3}{8}\mathrm{i}2\pi k} + (0.375 - 0.25\,\mathrm{i})\,\mathrm{e}^{\frac{-2}{8}\mathrm{i}2\pi k} \\ &\quad + (-0.6768 + 0.9053\,\mathrm{i})\,\mathrm{e}^{\frac{-1}{8}\mathrm{i}2\pi k} + 1.875\,\mathrm{e}^{\frac{0}{8}\mathrm{i}2\pi k} + (-0.6768 - 0.9053\,\mathrm{i})\,\mathrm{e}^{\frac{1}{8}\mathrm{i}2\pi k} \\ &\quad + (0.375 + 0.25\,\mathrm{i})\,\mathrm{e}^{\frac{2}{8}\mathrm{i}2\pi k} + (-0.3232 - 0.1553\,\mathrm{i})\,\mathrm{e}^{\frac{3}{8}\mathrm{i}2\pi k}. \end{aligned} \tag{D4}$$

Setting $k = 0$ to 7 here returns the numbers $[x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7]$ as $[1, 2, 3, 4, 5, 0, 0, 0]$, as expected.

Appendix E Calculating a Signal-to-Noise Ratio

Here we calculate the signal-to-noise ratio (SNR) that results from a given set of radar and pulse parameters. Knowledge of the SNR is necessary to allow us to add realistic noise when modelling a returned signal.

Consider a coherent processing interval made of N emitted pulses. These pulses are effectively combined, or *integrated coherently*, by the operation of the Fourier transform of Sections 5 and 6 that produces Doppler information from those pulses.

The central equation that governs the combination of the pulses is the DFT (6.1). When N is, say, 32, the DFT is a weighted sum of the samples $x_0, \ldots, x_{31}$, and returns the numbers $X_{-16}, \ldots, X_{15}$. The weights are unit-magnitude complex numbers. When N complex noise samples are transformed in this way, their weighted sum can be treated as a “random walk”. It’s well known that the root-mean-square distance from start to end of N unit-length steps in a random walk equals $\sqrt{N}$. Now use the fact that the power of an electromagnetic wave is proportional to the square of its amplitude, so:

$$\begin{aligned}\text{mean power of } N \text{ noise samples} &\propto \text{mean of (displacement due to } N \text{ steps)}^2\\ &= \text{(root-mean-squared displacement due to } N \text{ steps)}^2\\ &= (\sqrt{N} \times \text{distance covered by 1 step})^2\\ &\propto N \times \text{power of 1 noise sample.} \qquad \text{(E1)}\end{aligned}$$

The two proportionality constants above are reciprocals of each other, so we arrive at

$$\text{mean power of } N \text{ noise samples} = N \times \text{power of 1 noise sample.} \qquad \text{(E2)}$$

This might be expected: it says that 10 light bulbs combine to give 10 times the brightness of one light bulb, because the non-coherent light from an incandescent bulb can be treated as noise.

Adding *signals* instead of noise is a different matter, because the phasors that describe the signals are coherent: adding them is essentially like constructing a path from the steps of a sober man, not a drunk one. The phasors, like the sober man’s steps, all point in about the same direction, so that the distance from start to end of N of these signal phasors is N times the length of one signal phasor. That means the power present in the sum of N signals is N^2 times the power of one signal. In other words, 10 “phase-locked” lasers add their light to produce a central spot whose brightness is 100 times the brightness of the spot due to a single laser—albeit that bright spot is narrower than the spot due to a single laser, as expected from energy conservation.

The purpose of the weights $\mathrm{e}^{-\mathrm{i}2\pi kn/N}$ in the DFT (6.1) is to cancel the phase increases that result from any particular frequency. For example, if we wish to extract from a signal the sinusoidal component whose phase is increasing (its phasor is rotating) at one radian per second, then we can *subtract* from each sample a phase that grows by one radian per second, and add the results. Only the sought-after component of the signal will then yield a series of complex numbers of constant phase, so that they add coherently; all other components will yield a series of complex numbers of varying phases, and these will produce some cancellation when they add. The result is that the sought-after component is amplified, while other components are suppressed. In this way, the DFT adds signals and noise. The peaks that result in the range-Doppler plot have their signal increased by a factor of N^2 and their noise increased by a factor of N. So the signal-to-noise ratio of N pulses is N^2/N, or N, times the SNR of one pulse.

The term *coherent pulse integration* denotes this N-fold gain from the Fourier transform that generates Doppler information. The signal-to-noise ratio of the coherently integrated power in N pulses is

$$\text{SNR of } N \text{ pulses} = N \times \frac{\text{signal power in one pulse}}{\text{noise power in one pulse}}\,. \tag{E3}$$

We need the signal and noise powers in one pulse.

Signal power in one pulse:

$$\text{signal power in one pulse} = \frac{\text{energy received in one pulse}}{\text{width of correlation peak}}\,. \tag{E4}$$

The radar has gain G, carrier wavelength λ, losses L, and the scatterer has cross section σ and distance r. The two-way radar equation says

$$\text{energy received in one pulse} = \begin{bmatrix}\text{energy emitted}\\ \text{in one pulse}\end{bmatrix} \times \frac{G^2\sigma\lambda^2}{(4\pi)^3 r^4 L}\,. \tag{E5}$$

The width of the correlation peak is $1/B$ where B is the emitted pulse's bandwidth.

Noise power in one pulse: This is kT_sB, where k is Boltzmann's constant, T_s is the radar's *system temperature* (which is its noise factor times the actual temperature), and B is the receiver bandwidth [6], which equals the emitted pulse's bandwidth B if there is no reason to make it smaller or larger than B: smaller would prevent the receiver from receiving all that it's required to, and larger would make the receiver produce excess noise. Equation (E3) becomes

$$\begin{aligned}\text{SNR of } N \text{ pulses} &= N \times \frac{\text{energy received in one pulse}}{\text{width of correlation peak} \times kT_sB}\\ &= N \times \begin{bmatrix}\text{energy emitted}\\ \text{in one pulse}\end{bmatrix} \times \frac{G^2\sigma\lambda^2}{(4\pi)^3 r^4 L \frac{1}{B}\, kT_sB}\\ &= \begin{bmatrix}\text{energy emitted}\\ \text{in one CPI}\end{bmatrix} \times \frac{G^2\sigma\lambda^2}{(4\pi)^3 r^4 L\, kT_s}\,.\end{aligned} \tag{E6}$$

The gain of a basic "one-lobe" radar is

$$G = \frac{4\pi}{\text{solid angle of beam}}\,. \tag{E7}$$

The solid angle of, say, a beam of rectangular cross section just equals the product of its two defining angles, each in radians.

E.1 Simulating Receiver Noise

Equation (E6) can be used to derive an expression for the noise which, when added to a pulse, simulates the noise generated in a receiver.

First, consider two sources injected into a narrow-band IF amplifier: a zero-mean gaussian noise voltage of standard deviation ν (we have already used "σ" for cross section), and a sinusoidal signal voltage of amplitude A. The signal-to-noise ratio for one pulse is

$$\mathrm{SNR} = \frac{\text{mean signal power}}{\text{mean noise power}} = \frac{\text{mean of square of signal voltage}}{\text{mean of square of noise voltage}} = \frac{A^2/2}{\nu^2}\,. \tag{E8}$$

Referring to (E6) for $N = 1$, we then have, for a transmitted power P_t (considered constant over one pulse width τ, not one PRI),

$$\frac{A^2}{2\nu^2} = \text{SNR for one pulse} = P_t\tau \times \frac{G^2\sigma\lambda^2}{(4\pi)^3 r^4 L\, kT_s}\,, \tag{E9}$$

from which it follows that

$$\nu = \frac{Ar^2}{G\lambda}\sqrt{\frac{(4\pi)^3 L\, kT_s}{2P_t\tau\sigma}}\,. \tag{E10}$$

From this ν a complex-number noise can now be generated: it has an amplitude drawn from the gaussian distribution $\mathcal{N}(0,\nu^2)$, and a random phase uniformly distributed between 0 and 2π.

E.2 Maximum Detectable Range of a Target

Suppose we know the signal-to-noise ratio $\mathrm{SNR}_{\mathrm{given}}$ that is expected for our radar, based on standard theory in radar texts that combines the probability of false alarm, probability of detection, and model of the background clutter such as a Swerling model. Equation (E6) can use $\mathrm{SNR}_{\mathrm{given}}$ to calculate the maximum range at which our radar will detect a target. To see how, note that in (E6) as the target–radar separation r reduces, the SNR increases, which is quite reasonable. If we now increase r again, the SNR will drop until it reaches $\mathrm{SNR}_{\mathrm{given}}$. This value of r is the maximum range r_{max} at which the target can be detected for the given signal-to-noise ratio $\mathrm{SNR}_{\mathrm{given}}$. We require to find r, given $\mathrm{SNR}_{\mathrm{given}}$. Call the energy emitted in one CPI "E_{CPI}", so that (E6) becomes

$$\mathrm{SNR}_{\mathrm{given}} = \frac{E_{\mathrm{CPI}}\, G^2\sigma\lambda^2}{(4\pi)^3 r_{\mathrm{max}}^4 L\, kT_s}\,. \tag{E11}$$

The losses L might be a function of range, so rearrange (E11) to solve for r_{max}:

$$r_{\mathrm{max}}^4 L(r_{\mathrm{max}}) = \frac{E_{\mathrm{CPI}}\, G^2\sigma\lambda^2}{(4\pi)^3\, \mathrm{SNR}_{\mathrm{given}}\, kT_s}\,. \tag{E12}$$

As an example, we calculate the maximum range at which a target of cross section $\sigma = 5000$ m^2 can be seen by a radar with the following parameters:

$$\begin{gathered}\mathrm{SNR}_{\mathrm{given}} = 63\,, \quad \text{carrier} = 9.3\text{ GHz (i.e. } \lambda = 0.0322\text{ m)}\,, \quad P_t = 300\,\mathrm{W}\,, \\ \tau = 16\,\mu\mathrm{s}\,, \quad 16\text{ pulses/CPI}\,, \quad G = 500\,, \quad T_s = 710\text{ K}\,, \quad \mathrm{PRI} = 500\,\mu\mathrm{s}\,.\end{gathered} \tag{E13}$$

Note that a power of $P_t = 300\,\text{W}$ means the 300 W is treated as constant over τ, the pulse width; P_t is *not* the mean power over a PRI.

For the losses L, we'll include only atmospheric attenuation. A textbook figure for this is 0.022 dB/km at 10 GHz, which is close enough to the carrier for our calculation. We must be very careful with the units. I think it's a very under-utilised observation about units that just as "6/2" means "the number of 2s in 6", or 3, the expression "$2r/(1\text{ km})$" means "the number of kilometres in $2r$", which is another way of saying "$2r$ expressed in kilometres". This observation allows a sentence about units to be expressed in concrete mathematics. First, convert the decibels here to bels to write the attenuation as 0.0022 bels per kilometre. This figure then allows (actually *defines*) the attenuation over our two-way radar trip of distance of $2r$ to be written as

$$L(r) = 10^{0.0022 \frac{2r}{1\text{ km}}}. \tag{E14}$$

Equation (E12) becomes

$$\begin{aligned} r_{\max}^4\, 10^{0.0022\frac{2r_{\max}}{1\text{ km}}} &= \frac{\overbrace{300\text{ J/s} \times 16\,\mu\text{s/pulse} \times 16\text{ pulses/CPI}}^{\text{this is } E_{\text{CPI}}} \times 500^2 \times 5000\text{ m}^2 \times 0.0322^2\text{ m}^2}{(4\pi)^3 \times 63 \times 1.38 \times 10^{-23}\text{ J/K} \times 710\text{ K}} \\ &\simeq 8.13 \times 10^{19}\text{ m}^4. \end{aligned} \tag{E15}$$

Take the fourth root of each side to give

$$r_{\max}\, 10^{0.0011\frac{r_{\max}}{1\text{ km}}} \simeq 95\text{ km}\,, \tag{E16}$$

and then divide both sides by 1 km:

$$\frac{r_{\max}}{1\text{ km}}\, 10^{0.0011\frac{r_{\max}}{1\text{ km}}} \simeq 95\,. \tag{E17}$$

As expected, this equation is dimensionless: we can set a dimensionless $x \equiv r_{\max}/(1\text{ km})$ ("x is $r_{\max}$ in kilometres") and solve $x\,10^{0.0011x} = 95$. This equation is easy to solve by trial and error; however, for a sophisticated approach, try using the Method of False Position. More generally, False Position never fails, unlike the more well-known but potentially chaotic Newton-Raphson method, which can fail catastrophically in general use.

The result is $x \simeq 78$: the maximum detectable range of the target is $r_{\max} \simeq 78\,\text{km}$.

Appendix F How to Amalgamate the Cross Sections of Many Scatterers

The previous appendix's calculation of the maximum range at which a ship can be seen required knowledge of the ship's cross section σ. In practice we might know the cross sections of a thousand individual scatterers that are used to model the ship. How can we amalgamate these to form a good estimate of the total cross section σ of the ship?

Refer to page 12, and consider the electric field of the returned wave that is incident on the radar. As usual, we assume that a single polarisation is being used. If the electric field's amplitude and phase are represented by the complex number A, then (2.29) and the comment after it say that the average areal power density is $\langle S\rangle = \varepsilon_0 c\,|A|^2/2$. This is the average power per unit area incident on the receiver. The radar equation tells us that this power density is proportional to the target cross section, so write

$$|A|^2 = K^2\sigma \tag{F1}$$

for some positive constant K that depends on the parameters present in the radar equation (which is easily calculated but which we don't need to know).

But the same argument applies to the power density received from the individual scatterers that make up the target. Writing the electric field amplitude and phase from the j^{th} scatterer as complex A_j, we have

$$A = \sum_j A_j\,, \quad \text{and} \quad |A_j|^2 = K^2\sigma_j\,. \tag{F2}$$

This last equation allows us to write $A_j = K\sqrt{\sigma_j}\,\mathrm{e}^{\mathrm{i}\phi_j}$ for some phase ϕ_j. Combining (F1) and (F2) gives

$$K^2\sigma = |A|^2 = \Big|\sum_j A_j\Big|^2 = \Big|\sum_j K\sqrt{\sigma_j}\,\mathrm{e}^{\mathrm{i}\phi_j}\Big|^2. \tag{F3}$$

The total effective cross section of the target is then

$$\begin{aligned}\sigma &= \Big|\sum_j \sqrt{\sigma_j}\,\mathrm{e}^{\mathrm{i}\phi_j}\Big|^2 = \sum_j \sqrt{\sigma_j}\,\mathrm{e}^{\mathrm{i}\phi_j} \sum_k \sqrt{\sigma_k}\,\mathrm{e}^{-\mathrm{i}\phi_k} \\ &= \sum_{jk} \sqrt{\sigma_j\sigma_k}\,\mathrm{e}^{\mathrm{i}(\phi_j-\phi_k)} = \sum_j \sigma_j + \sum_{\substack{jk\\ j\neq k}} \sqrt{\sigma_j\sigma_k}\,\mathrm{e}^{\mathrm{i}(\phi_j-\phi_k)}.\end{aligned} \tag{F4}$$

For a target composed of many scatterers, the phases $\phi_j - \phi_k$ vary randomly over all values 0 to 2π. It follows that if the scatterers all have similar cross sections, the last sum over j and k is approximately zero. Hence to a high approximation,

$$\sigma \simeq \sum_j \sigma_j\,. \tag{F5}$$

That is, the total cross section of the target is effectively just the sum of the cross sections of the individual scatterers that make it up. This is a simple result, but it requires a large number of scatterers of roughly similar cross sections that are all returning signals with uncorrelated phases.

This page is intentionally blank.

Appendix G Examples of Calculating Ambiguity Functions

The ambiguity function introduced in Section 4.1 is nothing more nor less than the correlation of an emitted signal with its return, when that return has acquired some Doppler shift. The function can be considered as a stacking of correlation plots, one for each value of the target Doppler, where this stacking creates a surface in three dimensions.

Specifically, each slice of this surface at some target Doppler ω_D is the absolute value of the correlation of the signal with a copy of itself that has been Doppler shifted—meaning each of the copy's elements has been multiplied by a Doppler factor $e^{i\omega_D t}$, where t increments from one element to the next within the copy. For example, the zero-Doppler slice of the ambiguity function is the absolute value of the correlation of the signal with an unaltered copy of itself; i.e., the slice is the signal's autocorrelation.

In principle, the signals that will most clearly show the target are those whose ambiguity functions have a high central peak and low side lobes. For example in the next few pages, examining whatever central peak may be present, and whether it's surrounded by other peaks, shows why rectangular non-chirped pulses are not as useful for radar signals as are more structured pulses, such as those that have been Barker coded.

Each of the following plots shows the ambiguity function of the chosen signal type. For each frequencies in a set of Doppler frequencies, we plot the absolute value of correlation versus sample number using the ideas of Section 4.2, and then we stack each of these plots together along the frequency axis. This produces a surface plot which is the ambiguity function for the chosen waveform. The correlation is that of the pulse with a copy of itself that has been Doppler shifted.

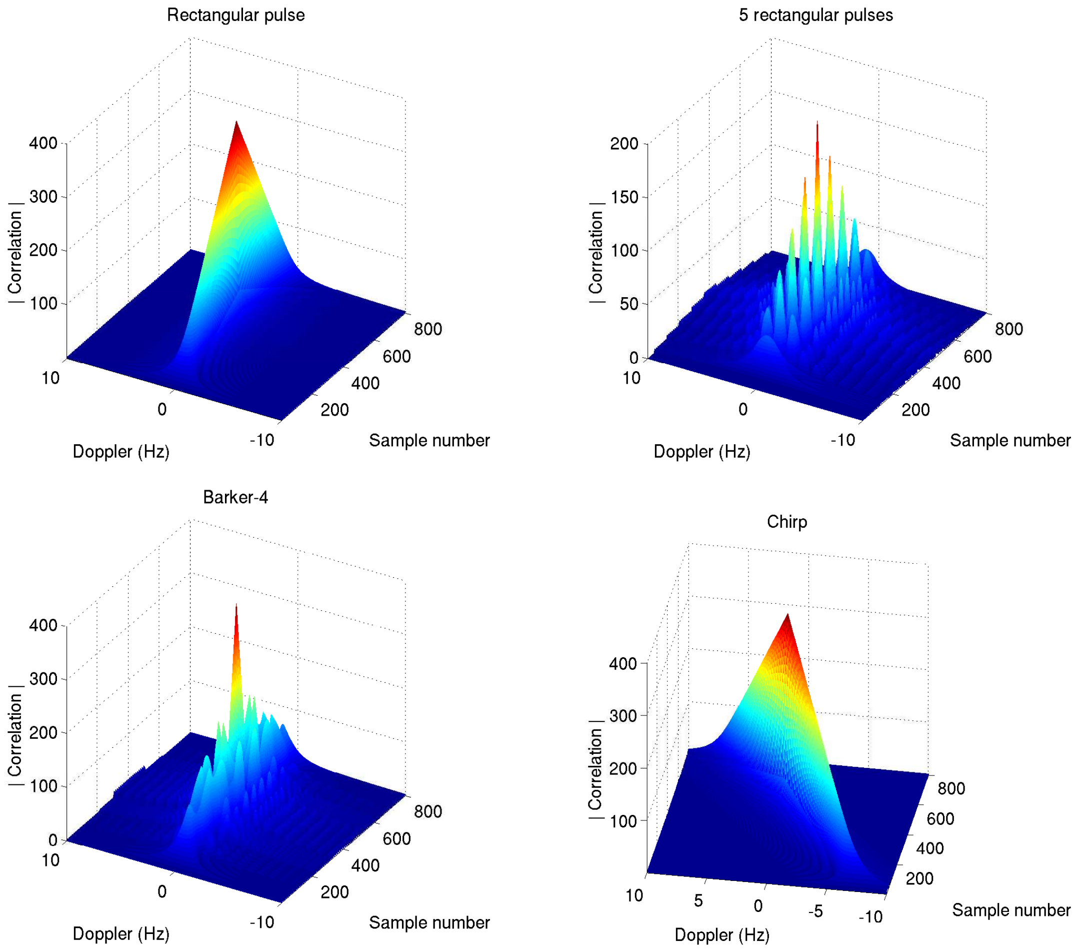
The kernel of creating these ambiguity functions is the following Matlab code, shown here for the case of a single rectangular pulse (although the actual code used for the following plots contained more detail):

```
t = 0 : 0.1 : 4;
signal = ones(1,length(t));

dopplerFrequencies = -5 : 0.1 : 5;

% Initialise.
correlations = zeros(length(dopplerFrequencies), 2*length(signal)-1);

rowNumber = 0;
for angularDoppler = 2*pi * dopplerFrequencies
  rowNumber = rowNumber + 1;
  correlation(rowNumber,:) = ...
      Correlate(signal, exp(1i*angularDoppler*t).*signal);
end
```

The matrix `abs(correlation)` can now be plotted. The above code calls on the one-line function `Correlate`, which gives its output in the order used in this report, being the conjugated reverse of Matlab's `xcorr` output without the unnecessary extra zeroes present in the output of `xcorr`:

```
function correlation = Correlate(array1, array2)

% Correlates array1 with array2.
correlation = conv(fliplr(conj(array1)), array2);
```

Figure G1 plots the ambiguity functions of various pulse types. In particular, the frequency of the chirp has been made to change very quickly throughout the pulse to highlight the rotation of the main lobe of the surface, as compared to the lobe of the single rectangular pulse. The Barker pulse has a well-defined central peak, but at a cost of added range side lobes that will, of course, add some noisy structure to a range–Doppler plot.


Rectangular pulse

5 rectangular pulses

Barker-4

Chirp

Figure G1: *Ambiguity functions for various pulse types: single rectangular pulse, a series of five rectangular pulses (whose PRI is twice the pulse width), a Barker-4 pulse and a fast-changing chirp.*

Index

Italicised numbers denote where the entry is defined.

This page is intentionally blank.

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA	1. CAVEAT/PRIVACY MARKING

2. TITLE How to Create and Manipulate Radar Range–Doppler Plots	3. SECURITY CLASSIFICATION Document (U) Title (U) Abstract (U)
4. AUTHOR Don Koks	5. CORPORATE AUTHOR Defence Science and Technology Organisation PO Box 1500 Edinburgh, SA 5111, Australia

6a. DSTO NUMBER DSTO–TN–1386	6b. AR NUMBER AR–016–185	6c. TYPE OF REPORT Technical Note	7. DOCUMENT DATE December 2014

8. FILE NUMBER 2012/1057198/1	9. TASK NUMBER 07/139	10. TASK SPONSOR DMO/MEWSPO	11. No. OF PAGES 85	12. No. OF REFS 6

13. URL OF ELECTRONIC VERSION `http://www.dsto.defence.gov.au/publications/scientific.php`	14. RELEASE AUTHORITY Chief, Cyber & Electronic Warfare Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for Public Release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS

No Limitations

18. DSTO RESEARCH LIBRARY THESAURUS

Radar jamming	Electronic countermeasures
Electronic warfare	Cross correlation
Signal processing	Fourier analysis

19. ABSTRACT

This report lays out the mathematical framework and reasoning involved in addressing the question of how to produce false targets in both range and Doppler to jam a radar. The explanations here are given from first principles in order to clarify the approach used in writing software to address the task. They constitute standard radar theory as seen through the eyes of a physicist, and are recorded as background to the approach taken in addressing the jamming problem in a later report, to be published.

We discuss how a radar generates a range–Doppler plot, using a set of parameters that describe the outgoing radar signal and the relevant characteristics of the target whose impulse response is known. We also explain necessary concepts from a mathematical physicist's viewpoint: bounds on pulse parameters, correlation/convolution, the theory of Hilbert transforms, the relevant Fourier analysis, and general concepts of radar signal processing such as ambiguity functions and the maximum detectable range of a target. This entire discussion is aimed at indicating the approach and philosophy used to solve the various problems encountered while working on the task.