Exploiting quasiperiodic electromagnetic radiation using software-defined radio

Christian David O'Connell



University of Cambridge
Computer Laboratory
Girton College

September 2018

This dissertation is submitted for
the degree of Doctor of Philosophy

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text.

This dissertation does not exceed the regulation length of 60 000 words, including tables and footnotes.

Exploiting quasiperiodic electromagnetic radiation using software-defined radio

Christian David O'Connell

Summary

Electronic devices emanate unintentional electromagnetic radiation from which an attacker can extract sensitive information. In video display units these are quasiperiodic: nearly periodic in the short term. Video-eavesdropping attacks on these, a main motivation for the use of TEMPEST shielded equipment in security-critical applications, have evolved little since first publicly demonstrated by van Eck in 1985. I investigate digital signal processing techniques that exploit the quasiperiodic nature of digital video signals, with TMDS-encoded data on HDMI/DVI cables as the main example.

After first discussing the practicalities of intercepting compromising emanations from the UHF frequency band, using a software-defined radio platform to perform IQ down conversion, I outline the process to carry out a video eavesdropping attack, and methods for rasterising intercepted data.

Using a database of video modes, such as VESA and CEA standards, I identify viable eavesdropping targets by fitting likely harmonics of emanating clock signals to a model. Video signals contain blanking intervals that create characteristic periodicities; cepstral features can be used to eliminate false positives, and provide improved performance over autocorrelation as a method of recovering synchronisation frequencies.

The signal-to-noise ratio of intercepted emanations is often very poor. Coherent periodic averaging in the complex domain can suppress noise and uncorrelated background sources. I design a phase-locked loop to perform clock recovery and synchronisation of the video signal, negating the effects of temperature drift in the local oscillators. This permits averaging arbitrary-length recordings, increasing the range at which an attack can be performed. I discuss the implications this may have on existing protection standards.

Finally, I present a method to recover bandwidths higher than that which the SDR front-end hardware is nominally capable of. I use the cross-correlation between multiple overlapping lower-bandwidth recordings to correct time and phase offsets, and a zero-phase Linkwitz-Riley filter pair to combine them. The resulting higher-bandwidth recordings improve raster clarity, and enable use of a hidden Markov model to recover colour information.

Acknowledgments

Markus Kuhn, Ian Wassell, David Oswald, David Greaves, Angelique Faye Loe, Jeunese Adrienne Payne, Khaled Baqer, Franck Courbon, Stephan Kollmann, Laurent Simon, Alexander Vetterl, Barbara Bracher, Tony Clayton, Garth Minton, Lise Gough, Sergei Skorobogatov, and the others who know why.

Contents

Acronyms **9**

Notation **13**

List of Symbols **17**

1 Introduction **19**

1.1 Compromising video emanations 19

1.2 A brief history of TEMPEST 21

1.3 Shielding strategies 22

1.3.1 Physical shielding 22

1.3.2 Device design 22

1.3.3 Jamming 23

1.3.4 Software-based protection 23

1.4 Practical eavesdropping 24

1.5 Motivations and scope 25

1.6 Structure 25

2 Video eavesdropping foundations **27**

2.1 TMDS video signal features 27

2.2 Emanation of information 30

2.3 Emanation capture 32

2.4 Sampling and down conversion 34

2.4.1 Hilbert transform 35

2.4.2 Digital down conversion 36

2.4.3 Software-defined radio front-ends . . . 39
2.5 Resampling . . . 44
2.6 Demodulation and rastering . . . 45
2.6.1 Amplitude information . . . 45
2.6.2 Phase information . . . 49
2.7 Signal-to-noise ratio estimation . . . 51
2.8 Simulating uniform wave propagation . . . 52
2.9 Data sets . . . 55
2.9.1 The VGA data set . . . 56
2.9.2 The Scan data set . . . 56
2.9.3 The Stitch data set . . . 57
2.10 Summary . . . 57

3 Using cepstral features for viable target device detection 59
3.1 Candidate harmonic identification . . . 59
3.2 Manual search . . . 61
3.3 Harmonic discriminant analysis . . . 63
3.4 Identifying periodicity features . . . 65
3.4.1 Autocorrelation . . . 66
3.4.2 Cepstral analysis . . . 68
3.4.3 Periodicity covariance . . . 72
3.4.4 False positive rejection . . . 75
3.5 Harmonic selection . . . 76
3.6 Summary . . . 77

4 PLL-based pixel-clock recovery for long-term complex averaging 81
4.1 Clock signal waveform . . . 81
4.2 Pixel-clock acquisition . . . 84
4.2.1 Direct digital synthesis . . . 86
4.3 Pixel-clock tracking . . . 87
4.3.1 Oscillator stability . . . 91
4.4 Variable-rate Lanczos resampling . . . 94

4.4.1 Resampler stability . . . 95
4.5 Complex-domain averaging . . . 98
4.6 Viable eavesdropping distance . . . 104
4.7 Summary . . . 106

5 Stitching multiple narrowband recordings into a single wideband 109
5.1 Frequency band selection . . . 110
5.2 Stitching . . . 111
5.2.1 Upsampling . . . 112
5.2.2 Frequency, phase, and delay alignment . . . 112
5.2.3 Crossover filtering . . . 115
5.2.4 Summation . . . 117
5.3 Character recovery . . . 119
5.3.1 Hidden Markov model . . . 124
5.3.2 Viterbi decoding . . . 126
5.4 Summary . . . 130

6 Conclusions 131
6.1 Future work . . . 133
6.2 Outlook . . . 136

Bibliography 137

Appendix A Transition-minimized differential signalling 147
A.1 Encoding . . . 148
A.2 Decoding . . . 149
A.3 Example encodings . . . 149

Appendix B Test card 153

Acronyms

ADEV	Allan deviation.
ASIC	Application-specific integrated circuit.
AWGN	Additive white Gaussian noise.
CEA	Consumer Electronics Association.
CNR	Carrier-to-noise ratio.
CRLB	Cramer-Rao lower bound.
DC	Direct current.
DDS	Direct digital synthesiser.
DSP	Digital signal processing.
DUT	Device under test.
DVI	Digital Visual Interface.
EMC	Electromagnetic compatibility.
EmSec	Emission security.
FFT	Fast Fourier transform.
FIR	Finite impulse response.
FPGA	Field-programmable gate array.
GCD	Greatest common divisor.
GUI	Graphical user interface.
HDMI	High-Definition Multimedia Interface.

HMM Hidden Markov model.

HSV Hue-saturation-value.

i.i.d. Independent and identically distributed.

IF Intermediate frequency.

IIR Infinite impulse response.

L-R Linkwitz-Riley.

LO Local oscillator.

LSB Least significant bit.

LVDS Low-voltage differential signalling.

MVU Minimum variance unbiased.

NCO Numerically controlled oscillator.

PAM Pulse-amplitude modulation.

pdf Probability density function.

PID Proportional-integral-derivative.

PLL Phase-locked loop.

ppb Parts per billion.

ppm Parts per million.

PSD Power spectral density.

RF Radio frequency.

SDR Software-defined radio.

SNR Signal-to-noise ratio.

TMDS Transition-minimized differential signalling.

UHD USRP hardware driver.

USRP Universal Software Radio Peripheral.

VDU Video display unit.

VESA Video Electronics Standards Association.

VGA Video Graphics Array.

XNOR Exclusive-NOR.

XOR Exclusive-OR.

Notation

x	A scalar value.
$\boldsymbol{x}$	A vector or time series of values.
$x(t)$	A value in time series $\boldsymbol{x}$ where $t \in \mathbb{R}$.
$x[n]$	A value in vector $\boldsymbol{x}$ where $n \in \mathbb{Z}$.
$\lvert x \rvert$	The absolute value operator such that $x = \lvert x \rvert \cdot \mathrm{e}^{\mathrm{j}\angle x}$.
$\angle x$	The argument of a complex number such that $x = \lvert x \rvert \cdot \mathrm{e}^{\mathrm{j}\angle x}$ and $\angle x \in [-\pi, \pi) \cdot 1$ rad.
$\measuredangle x[n]$	The unwrapped phase angle of an element in a sequence $\boldsymbol{x}$ where $\measuredangle x[n] = \angle x[n] + \left(\sum_{i=1}^{n} \mathbb{1}_{\angle x[i]-\angle x[i-1]\leq-\pi} - \mathbb{1}_{\angle x[i]-\angle x[i-1]>\pi}\right) \cdot 2\pi$ rad.
x^*	The complex conjugate: $x = a + b\mathrm{j} \to x^* = a - b\mathrm{j}$, $\forall a, b \in \mathbb{R}$.
$\Re(\cdot)$	The real component: $x = a + b\mathrm{j} \to \Re(x) = a$, $\forall a, b \in \mathbb{R}$.
$\Im(\cdot)$	The imaginary component: $x = a + b\mathrm{j} \to \Im(x) = b$, $\forall a, b \in \mathbb{R}$.
$\delta(\cdot)$	The Dirac delta function with $x(0) = \int_{-\infty}^{\infty} x(t) \cdot \delta(t)\,\mathrm{d}t$.
$\text{Ш}_\tau(\cdot)$	The Dirac comb function $\text{Ш}_\tau(t) = \sum_{i=-\infty}^{\infty} \delta(t - i \cdot \tau)$.
$\mathcal{F}\{\cdot\}$	The discrete Fourier transform $\mathcal{F}\{\boldsymbol{x}\}[f] = \sum_{n=0}^{N-1} x[n] \cdot \mathrm{e}^{-2\pi\mathrm{j}fn/N}$.
$\mathcal{F}^{-1}\{\cdot\}$	The inverse discrete Fourier transform $\mathcal{F}^{-1}\{\boldsymbol{X}\}[n] = \frac{1}{N}\mathcal{F}\{\boldsymbol{X}^*\}[n]^*$.
$\mathcal{H}\{\cdot\}$	The Hilbert transform $\mathcal{H}\{\boldsymbol{x}\}(t) = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{x(\tau)}{t-\tau}\,\mathrm{d}\tau$.
$\text{H}(\cdot)$	The Heaviside step function $\text{H}(t) = \int_{-\infty}^{t} \delta(s)\,\mathrm{d}s$.

$X(s)$ The Laplace transform of continuous sequence $\boldsymbol{x}$ where $X(s) = \int_{-\infty}^{\infty} x(t)\mathrm{e}^{-st}\,\mathrm{d}t$.

$X(z)$ The z-transform of discrete sequence $\boldsymbol{x}$ where $X(z) = \sum_{n=-\infty}^{\infty} x[n] \cdot z^{-n}$.

$\boldsymbol{X}$ The Fourier transform of $\boldsymbol{x}$ where $\boldsymbol{X} = \mathcal{F}\{\boldsymbol{x}\}$.

$\left|\overline{\boldsymbol{X}}\right|^2$ The power spectral density of $\boldsymbol{x}$.

$\mathbb{C}^{\mathbb{Z}}$ The set of all functions mapping a domain $\mathbb{Z}$ to a codomain $\mathbb{C}$, i.e. $\mathbb{C}^{\mathbb{Z}} = \{f \,:\, f : \mathbb{Z} \to \mathbb{C}\}$.

$\mathbb{Z}_N$ The set of integers in the interval $\mathbb{Z}_N = \{0, 1, \ldots, N-1\}$.

$|\boldsymbol{Y}|$ The cardinality operator of a set $\boldsymbol{Y} = \{Y_1, Y_2, \ldots, Y_N\} \to |\boldsymbol{Y}| = N$.

$\mathbf{LPF}_f(\cdot)$ A low-pass filter with cut-off frequency f (3 dB of attenuation). Optionally the function symbol may further specify the filter type, e.g. cheby2.

$\mathbf{LCM}\,(\cdot,\cdot)$ The lowest-common-multiple function.

$\mathbb{E}\,[\cdot]$ The expected value of a random variable.

$\mathbb{V}\,[\cdot]$ The variance of a random variable.

$Z \sim \mathcal{N}(\mu,\, \sigma^2)$ A normally distributed random variable Z where $\mathbb{E}\,[Z] = \mu$ and $\mathbb{V}\,[Z] = \sigma^2$ with pdf $z(x) = \dfrac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

$Z \sim \mathcal{R}(\nu,\, \sigma)$ A Rician distributed random variable Z with pdf $z(x) = \dfrac{x}{\sigma^2}\mathrm{e}^{-\frac{(x^2+\nu^2)}{2\sigma^2}} I_0\left(\dfrac{x\nu}{\sigma^2}\right)$, where $I_0(\cdot)$ is a zero-order modified Bessel function of the first kind.

$Z \sim \mathcal{U}(a,\, b)$ A uniformly distributed random variable Z with pdf $z(x) = \begin{cases} (b-a)^{-1} & a \le x \le b \\ 0 & \text{otherwise} \end{cases}$.

$\nabla\,\times$ The curl operator.

$\nabla\,\cdot$ The divergence operator.

$\mathbb{1}_x$ The indicator function where $\mathbb{1}_x = \begin{cases} 1 & x \\ 0 & \neg x \end{cases}$.

$x \oplus y$ The exclusive-OR operator defined for $x, y \in \mathbb{Z}_2$ such that $x \oplus y = (x + y) \bmod 2$.

$x \mathbin{\overline{\oplus}} y$ The exclusive-NOR operator defined for $x, y \in \mathbb{Z}_2$ such that $x \mathbin{\overline{\oplus}} y = \neg(x \oplus y)$.

$w_{\mathrm{D}} \times h_{\mathrm{D}} @ f_{\mathrm{v}}$ Displayed video mode. The dimensions of the addressable pixel data in the video signal and the frame rate.

$w_{\mathrm{T}} \otimes h_{\mathrm{T}} @ f_{\mathrm{p}}$ Full video mode. The total dimensions of the video signal, including the blanking interval, and the pixel rate.

List of Symbols

B	Bandwidth.
ς	Carrier-to-noise ratio.
e	Euler's number.
f_{b}	Bit rate.
f_{c}	Centre frequency.
$f_{\mathrm{c,RX}}$	Operating frequency of LO in receiving device.
$f_{\mathrm{c,TX}}$	Operating frequency of LO in transmitting device.
f_{d}	Offset frequency.
f_{e}	Frequency error.
f_{h}	Horizontal refresh rate, also known as the line rate.
f_{o}	LO offset from f_{c}.
f_{p}	Pixel clock frequency.
f_{PLL}	PLL rate.
f_{s}	Sampling rate.
f_{t}	Tracking frequency.
f_{v}	Vertical refresh rate, also known as the frame rate.
h	Harmonic index.
h_{D}	Height of video content in lines, excluding the blanking interval.
h_{T}	Total height of video frame in lines, including the blanking interval.
j	Imaginary unit $\mathrm{j} = \sqrt{-1}$.

ϱ	Signal-to-noise ratio.
τ	Sampling period $\tau = {f_\mathrm{s}}^{-1}$.
τ_h	Duration of the horizontal refresh rate $\tau_\mathrm{h} = {f_\mathrm{h}}^{-1}$.
τ_v	Duration of the vertical refresh rate $\tau_\mathrm{v} = {f_\mathrm{v}}^{-1}$.
w_D	Width of video content in pixels, excluding the blanking interval.
w_T	Total width of video frame in pixels, including the blanking interval.

Chapter 1

Introduction

Information leakage via electromagnetic emanations is a property inherent to all systems that communicate using electrical currents [1, 2]. Changes in electrical current cause the emission of electromagnetic waves which can be intercepted and analysed by an eavesdropper. This creates an exploitable attack vector, even when the device has been air gapped by removing network or direct physical access. The information leaked from side-channel attacks range from a single bit describing operational status [3] to more complex information. Such compromising emanations have been an ongoing source of anxiety for various government organisations for the better part of the last century. Much of the literature in the field is classified because of its often covert nature. Coupled with a reputation for being high-risk, low-reward, eavesdropping attacks have been largely ignored as a practical threat model by most organisations. However, the proliferation of software-defined radio (SDR) due to dropping hardware costs [4] has precipitated a shift to software processing, challenging that notion.

1.1 Compromising video emanations

Unintentional information leakage is far from a new phenomenon; crosstalk on phone lines has been noted as far back as the 19^{th} century [5]. Compromising emanations, a subset of side-channel attacks, focus on the emission of various information carrying signals from a target device. While the term can cover any form of emanation, such as acoustic emissions [6], it is most often used in the context of eavesdropping electromagnetic emanations, sometimes known as radiative coupling [5] to differentiate them from crosstalk [7], known as conductive coupling. Information has been successfully recovered via eavesdropping from a variety of devices including keyboards [3], touchscreens [8], RFID tags in smart cards [9, 10], laser printers [7], dot-matrix printers [6], and cryptographic modules, including those using CMOS [10], application-specific integrated circuits (ASICs), and field-programmable gate arrays (FPGAs) [11] where cryptographic keys were extracted [12].

Devices that emanate periodic signals, for example a keyboard's key-press monitoring loop [3], are common targets as they are easier to observe in the spectrum [2, 13] and offer multiple opportunities to intercept data. Ulaş *et al.* demonstrated the usefulness of this property in [7] where an investigation of laser printer emanations, which do not contain periodic features, required multiple pages of repeating patterns to be fed to the printer to force a locatable periodic signal.

Perhaps one of the most publicly visible types of attack on signals with periodic features is video eavesdropping. Video display units (VDUs) emanate information that allow eavesdroppers to estimate screen content remotely. As a consequence of their secretive nature, literature relevant to video emanations in the public domain is somewhat sparse [14]. Capabilities and protection methods remain locked away from prying enemy eyes, which leaves a rather narrow field for the amateur eavesdropper to consider. What does exist primarily builds upon the work of van Eck [15] of the eponymous *van Eck phreaking* attacks. In 1985 he demonstrated recovery of screen data from a VDU using a black and white TV receiver from a range of around 50 m. He outlined the attack as one that is relatively easy to mount by someone with the appropriate know-how and a few days to put together the necessary equipment. Indeed, following the dissemination of the attack a number of similar receivers were built from only the high-level description [16]. The flurry of papers written in the aftermath of van Eck's work focused on now outdated CRT displays [17–19]. However newer papers investigate more modern VDUs [20–23]. While the eavesdropped information is often presented in the form of an image raster, this is not always the case: *Tempest for Eliza*, famously demonstrated by Thiele in 2001 [24] and based on the 1998 work by Kuhn and Anderson [25], used screen data to modulate audio on a nearby radio. In 2007 another audio-bandwidth radio side channel allowed eavesdroppers to distinguish between candidates displayed on a Dutch voting machine [26] by altering the tone played by a radio when a particular party name was displayed. Video emanations have clear security implications, allowing sensitive material to be stolen without ever 'leaving' a system.

VDUs often emit multiple periodic components such as the pixel, frame, and line frequencies depending on implementation. The video data itself can be considered *quasiperiodic*: it exhibits short-term periodic behaviour. Consecutive frames and lines tend to be highly similar, if not identical, in order to be useful to a human observer. Making multiple measurements of the emanated data presents interesting possibilities for post-processing. Much of the existing literature focuses on the extraction of information from single observations—Hayashi *et al.* [27] describe VDUs as single observation devices—and the quasiperiodic property is ignored. VDU make an ideal candidate for investigating quasiperiodic emanations as they are ubiquitous, inexpensive, and can leak highly sensitive information.

1.2 A brief history of TEMPEST

While there is evidence of eavesdropping attacks going as far back as the First World War, when the German army used conductive emanations to eavesdrop enemy phone lines [28] [29, p.106], protection protocol standardisation did not occur until the 1950s [18]. *NAG1A* outlined acceptable limits on electronic emission levels across the spectrum known as limit curves, as well as criteria for certifying testing services and instrumentation. At some point the word *TEMPEST* became attached. The exact origins of the term TEMPEST remain classified [20, 21]; sources give conflicting definitions as an acronym—if it is indeed an acronym[1]—but there is no consensus, much less a reference[2]. Regardless, in the literature the term is used colloquially to describe both relevant protection standards, and side-channel attacks exploiting electromagnetic emissions [4, 20, 40–42]. Russell and Gangemi [33] outline the goal of TEMPEST: to allow government agencies to take an off-the-shelf approach to the acquisition of equipment shielded to prevent emanations. However the extensive red tape involved in the manufacture and sale of TEMPEST equipment significantly increases costs. Mauriello [34] estimated that making a device TEMPEST compliant would increase costs three to four fold. This is corroborated by Rowe [2], who stated the CIA declared that TEMPEST shielding was not cost effective in 1991. The latest known iteration of the US TEMPEST standard, called *NSTISSAM TEMPEST/1-92* [43], can only be shared with select friendly governments on a need-to-know basis. In 1982, NATO adopted an equivalent standard: *AMSG 720B* [15]. The current version is known as *SDIP-27/1*. While the limit curves remain classified, Le Bail [44] shows some emission limit curves which are possibly based on AMSG 720B, although this is unconfirmed.

There have been tentative studies into a civilian version of TEMPEST. In 1984 the Swedish government published a short report on compromising emanations and how best to protect against eavesdropping [45]. They suggested methods such as room insulation and careful device placement [16]. Although never translated from its native Swedish, similar advice was published online in 1993 in English [46]. A more rigorous approach proposed by Kuhn in 2005 [47] has also not been widely adopted, suggesting many do not consider eavesdropping protection a worthwhile investment.

[1] There appears to be conflict in the literature over whether the term should be presented as *Tempest* or *TEMPEST*. As the capitalised form is used in unclassified [30] and declassified material [31, 32] I will adopt this notation, but note this could be as much a formatting choice as an implication of an acronym.

[2] Proposed definitions include: no particular meaning [33]/not an acronym [34], Transient Electromagnetic Pulse Standard [35], Transient Electromagnetic Pulse Emanation Standard [36], Transient Electromagnetic Pulse Emanation Surveillance Technology [37], Telecommunications Electronics Material Protected from Emanating Spurious Transmissions [13, 38], and other less likely candidates [39].

1.3 Shielding strategies

Protecting against eavesdropping is a difficult task. Finding an eavesdropper is not easy, and even if caught prosecution would be unlikely as it is generally[3] not illegal to passively listen to the radio spectrum [48, 49]. The only viable solution is to reduce the signal-to-noise ratio (SNR) an adversary can receive. Efficacy of a shielding strategy is measured by its ability to attenuate emissions through a combination of absorption and reflection of compromising emissions [50]. One of the simplest approaches is *zoning*, where a device is placed away from uncontrolled space allowing emanations to dissipate before they could be intercepted [33]. Space limitations commonly prevent such a simple approach. Therefore a variety of more pragmatic strategies can be employed.

1.3.1 Physical shielding

The most common method of protection is to apply physical shielding to components that emanate information. The difficulty in shielding stems from the number of different components that can radiate information [1, 17, 51]. Often, the device is placed in a Faraday cage—a metallic enclosure with non-conductive gaps (e.g. for ventilation) kept below at least one tenth of the emanated wavelengths [2]. Weaknesses arise from damage to the enclosure or human error. The latter could occur due to leaving a door or panel open—a particular worry if the device is cumbersome to use or requires regular maintenance. Cracks in the enclosure could not only compromise the system, but assist an eavesdropper [52] by acting as a directional slot antenna [53]. Additionally, a close-fitting container requires devices to have the same shape, which can result in older equipment being used past its natural shelf life to reduce costs. Even without these weaknesses, the effectiveness of shielding is debatable. It is not unheard of for supposedly shielded devices to be exposed as anything but secure [54]. In one experiment, Anderson [53] showed that a *brute force* shielding approach was able to shield a device such that it could only be eavesdropped at point-blank range outside of business hours when the radio frequency (RF) spectrum was quieter, but concluded such cumbersome approaches were impractical.

1.3.2 Device design

For proper shielding the device and circuitry [1] would need to be designed from the ground up [52, 55] to reduce emissions at their source [56], rather than adapting off-the-shelf equipment. Recommendations include the use of ferromagnetic materials to

[3] In the UK it is an offence under the Wireless Telegraphy Act 2006, §48(1)(a) to use 'wireless telegraphy apparatus with intent to obtain information as to the contents, sender or addressee of a message (whether sent by means of wireless telegraphy or not) of which neither he nor a person on whose behalf he is acting is an intended recipient'.

attenuate high frequency components, short cable lengths [1], and balanced signals on twisted cable pairs to reduce the electromagnetic field [13]. However Hayashi *et al.* [57] showed that cable length has a more complex relationship with emanation levels. If disturbed, twisted cables can form a loop which acts as a radiating antenna; and balanced signals are rarely fully balanced in practice. RED/BLACK separation [58], where components handling classified information in their unencrypted forms (RED) should be physically separated from those carrying unclassified and encrypted data (BLACK), must be observed to prevent conductive emanations. Optical fibres can prevent electromagnetic leakage. However, conversion back to electrical impulses or booster units over long distances will provide a weak point in the chain still in need of protection [2]. Clearly this is an expensive, non-trivial task, and it is often simpler to force standards compliance by adding layers of shielding, and in the case of cables: encryption.

1.3.3 Jamming

An alternative approach to reducing the level of signal available to an attacker is to increase the noise level to 'drown out' sensitive information [33]. Also known as *active shielding* [59] or *heterodyning strategy*, jamming masks emanations with noise to reduce the estimation accuracy an eavesdropper can achieve [37, 60]. As with device design this is difficult to accomplish correctly, and in some cases illegal [52] due to transmitting over license-controlled RF bands. Many signals in computers are square waves causing the emanation of a wide bandwidth of constituent frequencies, all of which must be masked to be effective [46]. Simple noise, such as white noise, could be averaged out and so should contain some synchronised component. In 2010, Suzuki and Akiyama [42] injected complex jamming noise into a video cable which was then radiated with the video data; when averaged, the jamming data provided an obscuring interference pattern. Placing similar devices nearby has been proposed. However, Anderson [53] explains that manufacturing tolerances would allow an eavesdropper to discern individual devices. Additionally valuable metadata can still be recovered; for example a jamming signal could reveal the location of sensitive devices and times of use allowing alternative attack vectors to be explored.

1.3.4 Software-based protection

Sometimes called *Soft TEMPEST* [25], software protection methods attempt to ensure emanated data is difficult for an eavesdropper to interpret. In VDUs scrambling the transmission order of lines has been proposed as far back as van Eck's original 1985 demonstration [15, 36]. However this would simply provide security through obscurity and would not stop a reasonably determined eavesdropper from recording emanations and disentangling at their leisure. A mechanism similar to how Kuhn's 1994 AntiSky [61]

project operated, which used cross-correlation to undo horizontal line scrambling, could be used for this task. Rowe [2] suggests encrypting data. While this may help, it is not always an option for the internal signals of devices, such as keyboards and VDUs, as the information needs to be decrypted prior to human use. This may provide an opportunity for the decrypted data to emanate from elsewhere in the system, such as the display controller. In one case this appears to have compromised a *cryptofax* machine [62].

1.4 Practical eavesdropping

Radiative-coupled eavesdropping attacks exist in two forms: in passive attacks, the most common approach, an eavesdropper passively records the radio spectrum containing compromising emanations. They are easy to carry out but can suffer from unreliability and low SNR. Active attacks on the other hand can increase the SNR by broadcasting a carrier signal to 'illuminate' the target device; an eavesdropper can then observe how the returned carrier wave has been modulated by the target to extract information [55]. However, this requires more expensive equipment and an eavesdropper will reveal their presence in the radio spectrum. Active eavesdropping attacks may also be less useful now, as increasing clock speeds mean emanations often fall in the microwave range providing better penetration ability and less refraction, making the non-absorbed signals easier to intercept [2]. Our focus will be on passive attack vectors. While cheaper than active attacks, the cost of human operators to carry out the attack can still result in high costs. Experienced operators are required as eavesdropping attacks are notoriously temperamental with environmental factors such as background noise level, time of day, and even season and atmospheric conditions [63], all serving to help or hinder an attack. Anderson [53] proposed a budget of thirty to fifty thousand USD; Watanabe *et al.* [60] put the estimated cost in the millions of dollars in 2010. Much of the existing literature describes simple analogue eavesdropping attacks using TV tuners [15, 52] or oscilloscopes to collect data. Since the 2010s, more modern attacks use SDR platforms [22, 64, 65], which allow collection of data using off-the-shelf equipment with the processing performed in software. This reduces the complexity to simply installing a software package, opening the attack to those without specialist knowledge. As one cannot exert any control over what is displayed on the target, at least not without additional access, eavesdropping is largely opportunistic. Building on this, some have begun to investigate how eavesdropping can act as a component within a larger attack rather than as the sole vector. For example in 2014 Guri *et al.* [66] outlined a method using radios in mobile phones to intercept and exfiltrate data emanated from a VDU modulated by malware. There has even been some research into benign uses for emanations, such as by Yang *et al.* [67] who attempt to use emanations as a replacement for QR codes.

1.5 Motivations and scope

We have seen that, despite a lack of awareness of the potential threat from the public, and questionable protection practices, there is little evidence of compromising emanations being regularly exploited. This implies eavesdropping is a potentially rich source to elicit sensitive information. Electromagnetic compatibility (EMC) standards ensure nearby electronics do not interfere with each other and are often inadvertently the first (and regularly the only) line of defence against eavesdropping. They are insufficient against information leakage as they do not differentiate between noise and data leakage [51]. In 2005 Kuhn [47] concluded the maximally allowed power of signal leakage would need to be several million times lower than what is permitted in contemporary EMC standards in order to be effective.

While the principles of performing a TEMPEST attack against a VDU are well understood, there has been little research beyond basic principles and attacks have hardly evolved beyond producing simple rasters. Thus I was interested in what more could be done with data containing an emanated signal that exhibits quasiperiodic properties. I took the perspective of an attacker and approached the issue from the software side of an SDR system. The described investigations used digital signal processing (DSP) techniques to develop an array of tools to assist with exploiting quasiperiodic signals, and automating eavesdropping on VDUs. I demonstrate the largely untapped potential of this vulnerability.

1.6 Structure

The content of this thesis is categorised into the following chapters:

- **Chapter 2** sets out the steps required to perform a video eavesdropping attack using an SDR system. It describes some of the underlying theory and limitations that SDR presents, and goes on to examine some of the approaches used to analyse and interpret collected data. This chapter encapsulates the current state of video eavesdropping, which is used as a foundation to build on in later chapters.

- **Chapter 3** investigates the spectral features of the quasiperiodic emanations from VDUs. Cepstrum analysis is compared against autocorrelation as a method to extract salient periodicity information, which we can use to identify potential eavesdropping targets and the recovery of their operational parameters. In keeping with the opportunistic nature of eavesdropping, no prior information about what devices may be emanating is assumed.

- The low SNR of detected devices can be increased through the application of periodic averaging. However operational parameters are non-stationary, and while approximate fixed values are sufficient to perform a basic eavesdropping attack, more advanced averaging processing techniques require increased precision. **Chapter 4** tracks changes to parameters using a control mechanism and applies appropriate correcting factors. This has implications for long-period averaging, allowing improved attenuation of uncorrelated background noise by employing complex-domain averaging over arbitrary length periods. This maximises the SNR and increases the distance at which an attack can be carried out.

- **Chapter 5** explores methods to circumvent the bandwidth limitations imposed by an SDR system. Significant information about screen content is lost due to the limited available bandwidth. However, with multiple recordings of a signal with quasiperiodicity one can reconstruct a wide bandwidth attack using hardware with otherwise insufficient capabilities. This allows extraction of data at the bit level and recovery of transmitted symbol information.

- **Chapter 6** summarises the findings and contextualises them in the wider field of side-channel attacks. It provides some outlook and ideas for future work.

Chapter 2

Video eavesdropping foundations

Staging a basic video eavesdropping attack using SDR is relatively straightforward. This chapter outlines the steps to mount the attack, and discusses the principles underpinning video eavesdropping.

2.1 TMDS video signal features

Transition-minimized differential signalling (TMDS) is an 8b/10b digital transmission encoding used in Digital Visual Interface (DVI) and High-Definition Multimedia Interface (HDMI) to transmit video data. It aims to reduce electromagnetic interference by performing two transformations: first, for each byte of video data either an exclusive-OR (XOR) or an exclusive-NOR (XNOR) operation is applied between consecutive bits to reduce the number of bit transitions to between two and six per 10 bit character, acting as a type of first-order derivative. Then the byte is inverted if that helps redress any long-term direct current (DC) imbalance caused by a surplus of either ones or zeros, keeping the mean of the modulated waveform near zero. The encoded 8 bit bytes are appended with two additional bits of information indicating whether an XOR or XNOR was applied, and whether inversion took place, creating 10 bit *characters* [68]. There are 460 unique 10 bit TMDS characters into which an 8 bit intensity value ($\mathbb{Z}_{256}$) can be encoded (Appendix A). When transmitting a constant value, the TMDS balancing algorithm produces cycles of up to nine characters (from a balanced state) to keep the waveform mean at zero. The waveform of a bit sequence $\boldsymbol{p}$ of length N can be seen as a convolution between a series of shifted Dirac delta functions with a rectangular pulse the width of a bit duration τ_b:

$$\boldsymbol{y} = \left\{ \mathrm{rect}\left(\frac{t}{\tau_\mathrm{b}}\right) \right\} * \left\{ \sum_{n=0}^{N-1} p[n] \cdot \delta(t - n\tau_\mathrm{b}) \right\} \tag{2.1}$$

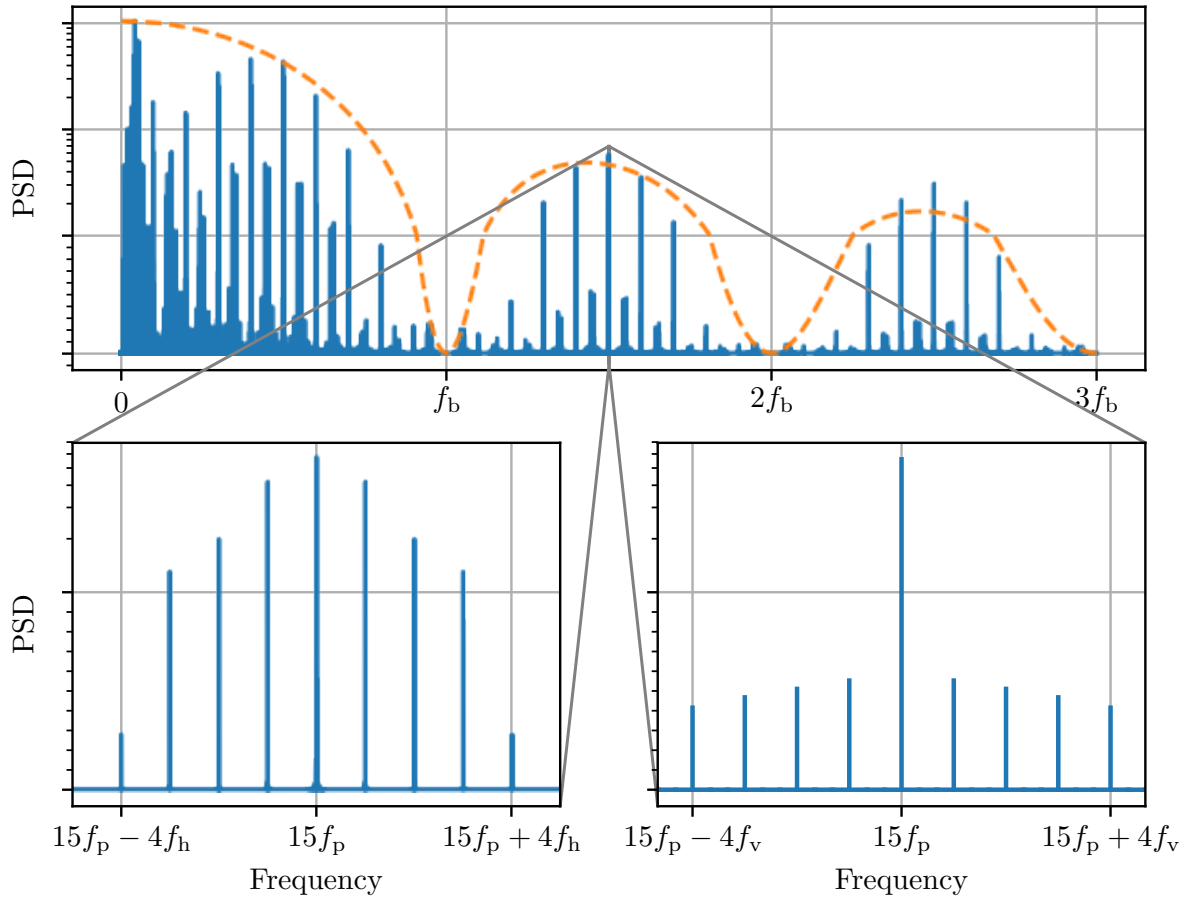


Figure 2.1. The envelope of the signal power spectrum, here created from a simulated TMDS signal, will be proportional to a sinc^2 function (orange) due to the frequency domain shape of the rectangular bit pulses. Thus no power occurs at multiples of the bit rate. Likewise, there is little to no power at DC due to the DC-balancing performed by the TMDS algorithm. Aliases of the video information will create frequencies at multiples of the pixel rate, around which further peaks spaced at the line (lower left) and frame (lower right) rates manifest. A periodic signal creates a line spectrum; a quasiperiodic signal leads to the peaks smearing out.

The power spectrum estimate (computed using Equation 3.16 as the Fourier transform of the autocorrelation function) is therefore

$$|Y|^2(f) = \mathrm{sinc}^2\left(f \cdot \tau_\mathrm{b}\right) \cdot \left| \sum_{k=-\infty}^{\infty} \mathcal{F}\{\boldsymbol{p}\}\left(f - k f_\mathrm{b}\right) \right|^2 . \tag{2.2}$$

Figure 2.1 demonstrates an example of the expected power spectral density (PSD) created using simulated TMDS video data. Most notably, because of the sinc^2 term no power is emitted at multiples of the bit rate. The modulated bit data produces a line spectrum at frequencies present in the video data. Pixel data is transmitted from left-to-right, top-to-bottom, known as raster scanning [20] (Figure 2.2). The video data encoded by TMDS often exhibits very low variability between consecutive frames, especially when text is shown. Fast, prolonged updates are rare as they would not be useful to the human

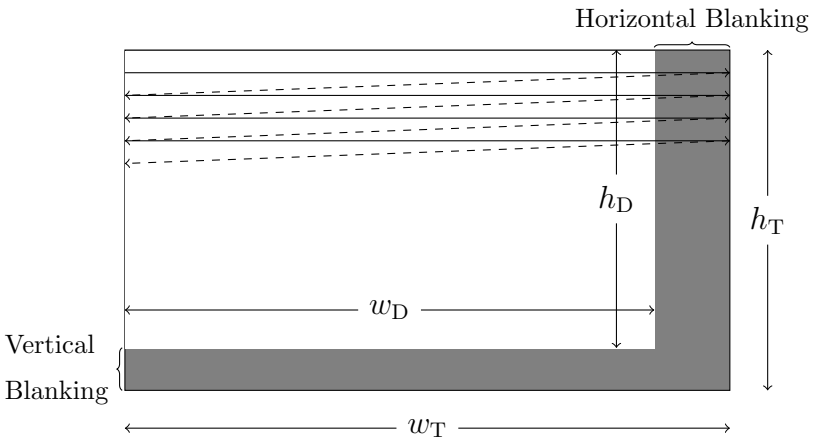


Figure 2.2. A video signal is transmitted from left to right, top to bottom. Video content is interspaced with a blanking interval between each line and frame. When an eavesdropped video signal is rastered the blanking interval appears as an area of largely uniform intensity. It is depicted here as a section of grey surrounding the content area, shown in white, which contains the image data to be displayed on the VDU.

operator. For a graphical user interface (GUI), changes tend to occur relatively slowly and are localised to one area of the screen, for example when editing text. Large changes, such as scrolling or switching to a new window, tend to happen quickly, disturbing only a few frames making them periodic in the short term, for the duration of which the pixel data is a cyclostationary process [69, p.20]. Likewise, the pixel content of successive lines often exhibits short-term periodic behaviour: backgrounds are often solid colours; GUI widgets, such as those created by the Qt or GTK+ tool kits, are commonly rectangular areas; and glyphs contain many vertical components. Thus we consider the video signal to be periodic in the short term, with consecutive frames more highly correlated than consecutive lines. We will also assume signal ergodicity, allowing us to infer signal properties from a single sufficiently long trace, in this case a few frames. In addition to the video data an area of non-addressable data, known as the blanking interval, is transmitted but not usually displayed by the VDU. It is transmitted between every line (the horizontal blanking interval) and frame (the vertical blanking interval). Use varies between devices and protocols. In CRT monitors its purpose was to allow the deflecting magnets to change the field, modulated by a sawtooth wave [70], giving the electron beam time to travel back across the screen [18]. In LCDs it is used for display-internal processing [21]. In HDMI it is sometimes used to transmit audio or for synchronisation purposes [71]. During transmission of the blanking interval the TMDS encoders output one of four control characters with eight bit transitions per character, to make a series of them unique and easy to identify. The blanking interval typically covers between 10–20% of the horizontal area, and up to 5% of the vertical area (in DMT and CVT standards), providing some guaranteed periodic behaviour independent of pixel content. This contributes to the stronger

frequency components at multiples of the frame and line rates we saw in Figure 2.1. If the video signal were otherwise periodic, these frequencies would create a line spectrum. As the signal becomes progressively less periodic the spectral line spacing will begin to 'blur' in the frequency domain.

In addition to the video data, an accompanying pixel clock[1] [68, p.25] of frequency f_p indicates transitions between pixels, fixed at one tenth of the bit rate f_b. The TMDS clock signal is a square wave (§4.1). As such it decomposes into constituent frequencies known as harmonics, in particular at odd multiples $h \in \{1, 3, 5, \ldots\}$ of the base frequency f_p. The TMDS data is sent using pulse-amplitude modulation (PAM) and thus will also have aliases throughout the frequency spectrum centred around the clock harmonics. This gives multiple locations in the spectrum from which to observe emanated data [19].

2.2 Emanation of information

The motivation behind TMDS is for EMC reasons rather than emission security (EmSec) [68, p.10]; attenuation of emanations is insufficient to reliably prohibit eavesdropping. Three simultaneous TMDS encoders—for the red, green, and blue colour channels—transmit across three serial lanes, along with the pixel clock, using current-mode logic. Differential signalling is common in modern high-speed serial communication interfaces. The waveform to be transmitted is encoded in the difference in current between two complementary wires, often a twisted pair, or pathways on a printed circuit board. This provides advantages in the context of electromagnetic immunity as incoming interference affects both lines similarly, minimising the common-mode noise. It also reduces outgoing interference: A current I on a wire has a current density

$$\boldsymbol{J} = \frac{\partial \boldsymbol{I}}{\partial A} \tag{2.3}$$

over area A. From Ampere's law we know $\boldsymbol{J}$ gives rise to a magnetic field (H-field) $\boldsymbol{H}$, proportional to the magnitude of the current:

$$\nabla \times \boldsymbol{H} = \frac{\partial \boldsymbol{D}}{\partial t} + \boldsymbol{J}\,, \tag{2.4}$$

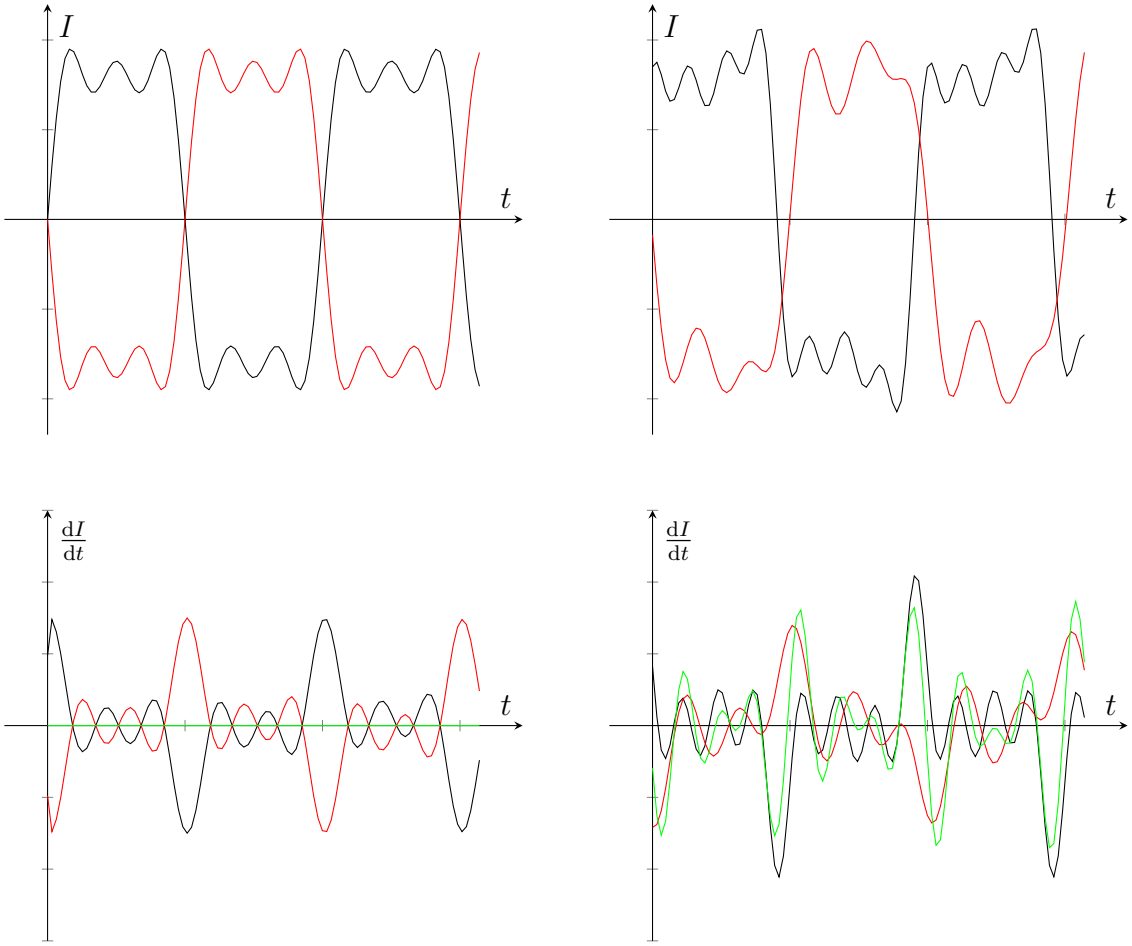
where $\boldsymbol{D}$ is the electric flux density (Equation 2.8). The magnetic flux density

$$\boldsymbol{B} = \mu \boldsymbol{H} \tag{2.5}$$

for some material with permeability μ has a divergence of zero (Gauss' law of magnetic flux):

$$\nabla \cdot \boldsymbol{B} = 0\,, \tag{2.6}$$

[1] In some literature known as the *character clock* or *dot clock*.



(a) Ideal **(b)** In practice (simulation)

Figure 2.3. In differential signalling the waveform is transmitted by two wires with complementary currents (black and red, top). In theory, the H-fields created by the derivative of the currents (black and red, bottom) should be of equal magnitude but opposite polarity (a), cancelling each other out (green). In reality, manufacturing tolerances result in imbalances and some residual field will remain (b). From Maxwell's equations we know this imbalance will cause a propagating electromagnetic wave of proportional magnitude to be emitted.

meaning the H-field will circulate the wire. As each twisted pair of wires has a current of the opposite sign, so too will the generated H-fields be of opposite polarity. In theory, these will be of equal magnitude and cancel each other out, eliminating electromagnetic emission [72]. In reality, the unequal impedance of the wires can cause asymmetry [23] in the rising and falling edges of the current waveform. This can be for a number of reasons such as simultaneous switching noise, load capacitance, duty cycle distortion [73], etc. The change in the residual H-field over time t will in turn create a changing electric field (E-field) $\boldsymbol{E}$ according to Farady's law:

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}\,, \tag{2.7}$$

which again induces a magnetic field with the electric flux density term in Ampre's law with

$$\boldsymbol{D} = \epsilon \boldsymbol{E}\,, \tag{2.8}$$

where ϵ is the material permittivity, and so on, propagating as an electromagnetic wave proportional to the size of the imbalance. The highest level of emission will be at the edges of the square wave, emanating the derivative of the signal (Figure 2.3).

2.3 Emanation capture

The emanated waves allow us to eavesdrop on a TMDS device in much the same manner as a Video Graphics Array (VGA) cable [74] or a CRT monitor, where the emanations were produced by the currents controlling the electron beam [75]. The surface integral of the H-field around a circle C of radius d enclosing a wire equals the current contained within C

$$\begin{aligned} I &= \oint_C \boldsymbol{H}\,\mathrm{d}\boldsymbol{s} \\ &= 2\pi d H\,. \end{aligned} \tag{2.9}$$

Rearranged, we can see the magnitude H of the H-field created by I is inversely proportional to distance d

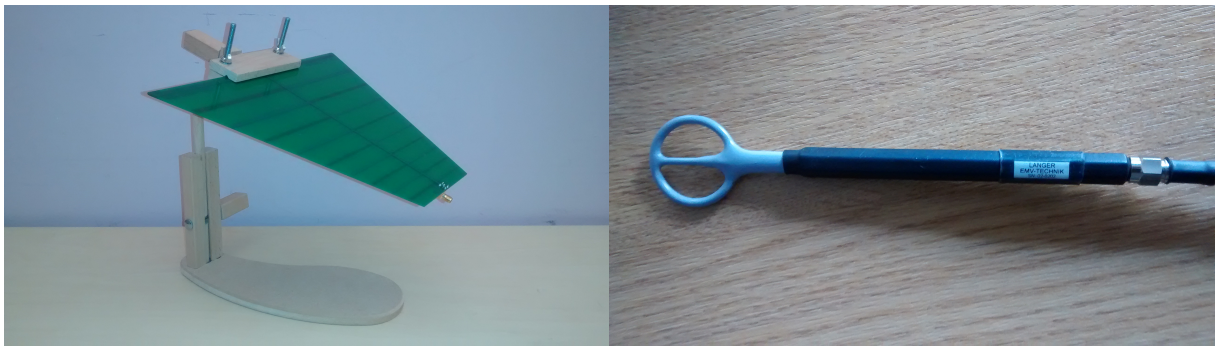
$$H = \frac{I}{2\pi d}\,. \tag{2.10}$$

Invoking Faraday's law again (in this greatly simplified outline of EM wave propagation) we see the propagating wave will induce an electrical current in a dipole antenna at distance d from the source. I captured emanations in the far field using a log-periodic antenna [10] directed towards the location of the target device. A log-periodic antenna is a directional antenna [41] consisting of an array of active dipole antennas of increasing lengths. A dipole antenna is most responsive to frequencies with twice the wavelength of its own length, and a suitable combination of many dipoles creates an antenna responsive over a wide frequency band [73]. A wide frequency range is needed to capture the wideband emanated video content. I used a WA5VJB PCB log-periodic antenna[2] (Figure 2.4a) which has a frequency range of 400–1000 MHz. Its small size and low cost (around 30 USD) make it a suitable antenna for amateur eavesdroppers. For experimentation over larger frequency ranges I used a Schwarzbeck VUSLP 9111B log-periodic antenna[3] (Figure 2.4c), which is calibrated over a frequency range of 200–4000 MHz. For recordings made in the near field I used a Langer XF-R 400-1 near-field probe[4] (Figure 2.4b) placed over the HDMI connector. This has a reasonably flat response to the H-field over

[2] http://www.wa5vjb.com/pcb-pdfs/LogPerio400.pdf

[3] http://schwarzbeck.de/Datenblatt/K9111B.pdf

[4] https://www.langer-emv.com/en/product/xf-passive-30-mhz-6-ghz/34/xf1-set-near-field-probes-30-mhz-up-to-6-ghz/278/xf-r-400-1-h-field-probe-30-mhz-up-to-6-ghz/21



(a) WA5VJB PCB log-periodic antenna

(b) Langer XF-R 400-1 near-field probe



(c) Schwarzbeck VUSLP 9111B log-periodic antenna.

Figure 2.4. A selection of antennas I used to collect data. The log-periodic antennae are directional and operate over a large range of frequencies and measure the E-field. The near-field probe is designed to operate in the near field of a transmitting device, measuring the H-field.

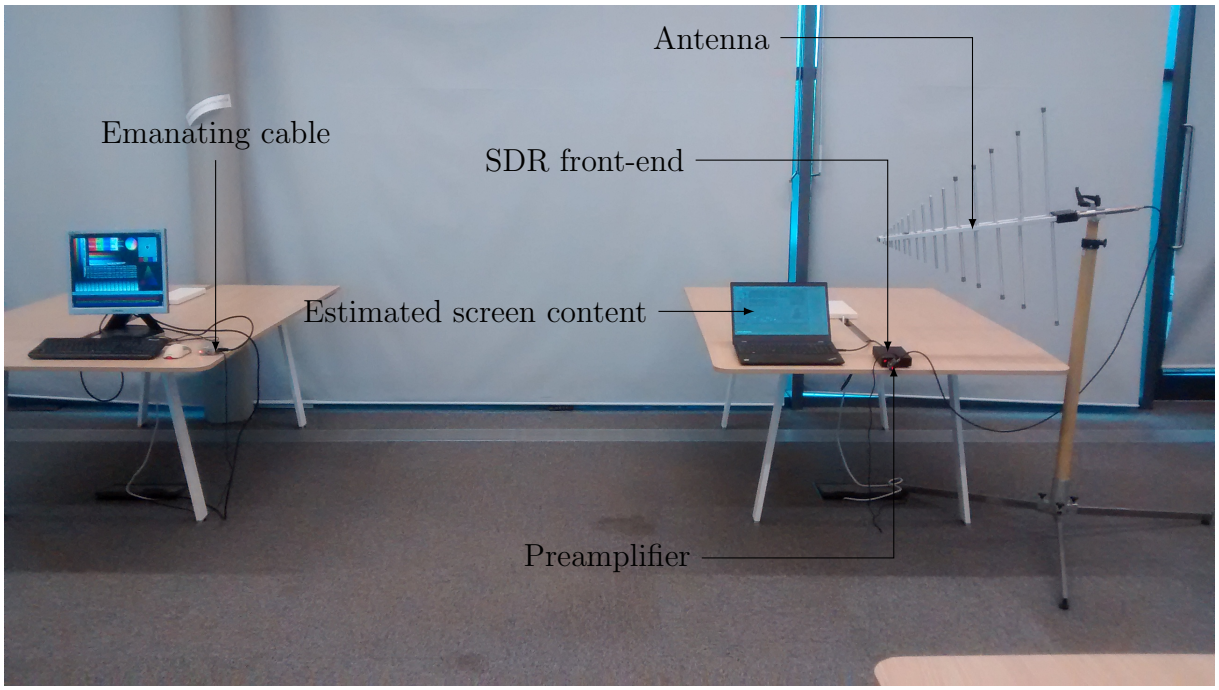


Figure 2.5. A video eavesdropping setup showing placement of the antenna relative to the target device. On the left, the DVI cable connected to the target device emanates information into the surrounding field. The log-periodic antenna on the right captures this information and passes it through a preamplifier feeding into the SDR front-end. The digitised samples are passed from the front-end to the computer, where further processing takes place in software and an estimate of the video information is displayed.

a range of 30–6000 MHz. A coaxial cable connects the antenna (optionally) to a Langer PA 303 preamplifier, which in turn connects to an SDR front-end. A typical setup of an eavesdropping attack is shown in Figure 2.5.

2.4 Sampling and down conversion

The SDR front-end converts the antenna signal from the continuous to the discrete domain. While digitising the signal provides benefits, such as the ability to process data in software using general purpose hardware, it also brings a set of associated costs, notably the introduction of quantisation error and aliasing. The level of quantisation error is determined by the resolution of the analog-to-digital converters in the hardware used. Aliasing on the other hand can be handled by down converting and filtering the appropriate band of the frequency spectrum. The optimal harmonic to eavesdrop will depend on factors such as the hardware, nearby sources of interference, environmental factors, etc. We select a suitable harmonic as our target frequency

$$f_\mathrm{c} \approx h \cdot f_\mathrm{p}\,. \tag{2.11}$$

2.4.1 Hilbert transform

A bandpass signal $\boldsymbol{v_R} \in \mathbb{R}^{\mathbb{R}}$, such as a voltage varying with time, is the result of modulating a sinusoid by amplitude $A(t)$ and either phase $\phi(t)$ or its first-order derivative frequency $f(t)$ around a centre frequency f_c

$$v_R(t) = A(t) \cdot \cos\left(2\pi \int_0^t (f_c + f(s))\,\mathrm{d}s\right). \tag{2.12}$$

Its Fourier transform is an even function, symmetric around DC (0 Hz), as negative frequencies are indistinguishable from phase-shifted positive frequencies. A single data point gives no information regarding the modulation parameters.

The Hilbert transform $\mathcal{H}\{\boldsymbol{v_R}\}$ is an all-pass filter with a linear phase response which shifts the phase at each frequency of $\boldsymbol{v_R}$ by $\frac{\pi}{2}$ rad. When added to $\boldsymbol{v_R}$ as an imaginary component it creates the complex-valued analytic form of the signal $\boldsymbol{v_C} \in \mathbb{C}^{\mathbb{R}}$. This can also be computed by multiplying the Fourier transform with the Heaviside step function H (doubled to maintain signal power)

$$\begin{aligned} v_C(t) &= v_R(t) + \mathrm{j} \cdot \mathcal{H}\{\boldsymbol{v_R}\}(t) \\ &= v_R(t) + \mathrm{j} \cdot \mathcal{F}^{-1}\{-\mathrm{j} \cdot \mathrm{sgn}(f) \cdot \mathcal{F}\{\boldsymbol{v_R}\}(f)\}(t) \\ &= \mathcal{F}^{-1}\{2\mathrm{H}(f) \cdot \mathcal{F}\{\boldsymbol{v_R}\}(f)\}(t). \end{aligned} \tag{2.13}$$

The analytic signal contains no negative frequency components meaning its Fourier transform is asymmetric. This has important implications as it gives us the ability to distinguish between positive and negative frequencies; information can now be represented across the full bandwidth of the signal rather than just half. We can more conveniently express $\boldsymbol{v_C}$ using a complex exponential with Euler's identity[5]

$$v_C(t) = A(t) \cdot \mathrm{e}^{2\pi\mathrm{j}\int_0^t (f_c + f(s))\,\mathrm{d}s} \tag{2.14}$$

such that $v_R(t) = \Re(v_C(t))$ and $\mathcal{H}\{\boldsymbol{v_R}\}(t) = \Im(v_C(t))$. A complex phasor $\mathrm{e}^{2\pi\mathrm{j}f't}$ of frequency $f' \in \mathbb{R}$ makes $\frac{|f'|}{1\ \mathrm{Hz}}$ complete rotations every second. When plotted with the real component on the x-axis and imaginary component on the y-axis (Figure 2.6), the sign of f' can be discerned by the direction of the rotation: positive frequencies rotate anticlockwise, and negative frequencies clockwise. If we multiply $\boldsymbol{v_C}$ with a phasor of frequency f', we shift the frequency information content in the spectrum by f'

$$v_C(t) \cdot \mathrm{e}^{2\pi\mathrm{j}f't} = A(t) \cdot \mathrm{e}^{2\pi\mathrm{j}\int_0^t (f_c + f' + f(s))\,\mathrm{d}s}. \tag{2.15}$$

This is known as down conversion when shifting information down through the frequency spectrum ($f' < 0$), and up conversion when shifting up ($f' > 0$).

[5] $\mathrm{e}^{\mathrm{j}\theta} = \cos(\theta) + \mathrm{j} \cdot \sin(\theta)$. Here the imaginary sine component is a $\frac{\pi}{2}$ rad delayed version of the real cosine component i.e. its Hilbert transform.

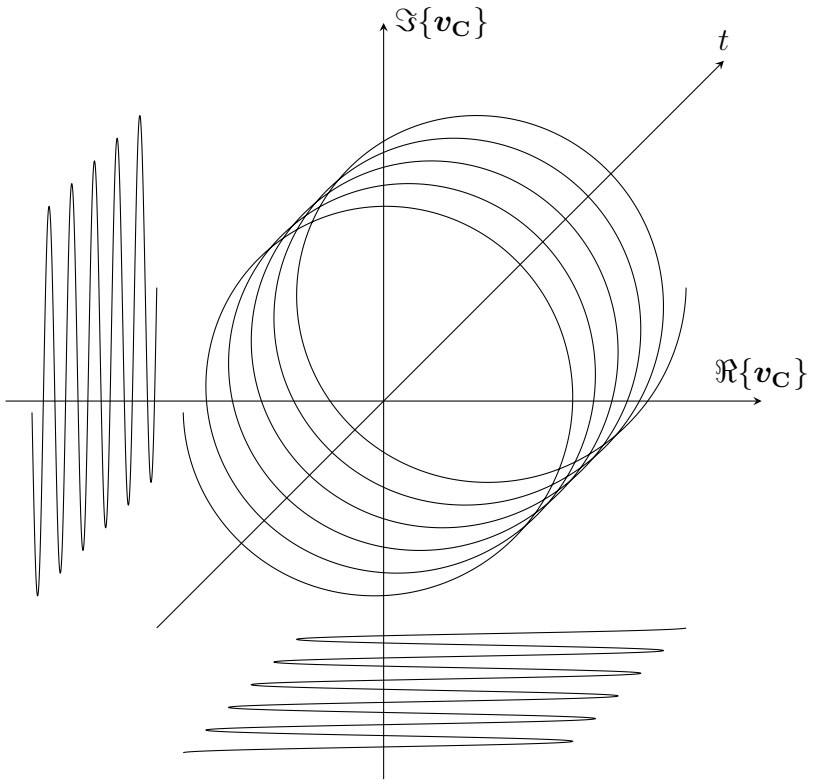


Figure 2.6. A complex sinusoid, also known as a phasor, rotates in two dimensions; anticlockwise for positive frequencies and clockwise for negative frequencies. This distinction allows $B = f_\mathrm{s}$ frequency information to be represented by a complex signal at a cost of storing the imaginary component. When projected into orthogonal dimensions the complex sinusoid decomposes into two real-valued sinusoids $\frac{\pi}{2}$ rad out of phase.

2.4.2 Digital down conversion

The conversion from the continuous real domain to the sampled complex domain is in practice often accomplished using a digital down converter. Sampling theory states that to reconstruct frequencies from 0 Hz ($-\hat{f}$ when the samples are complex) to $\hat{f}$ it must be sampled by at least

$$f_\mathrm{s} = 2 \cdot \hat{f}\,, \tag{2.16}$$

widely known as the Nyquist frequency [76], to prevent aliasing—the case where a lack of enough information to reconstruct one frequency causes it to appear as another. Sampling the RF spectrum requires a very high and often intractable sampling rate to fulfil the Nyquist criterion. However, as we are interested in a narrower frequency band, digital down conversion circumvents this problem by shifting the band of interest to baseband (the region around 0 Hz) in the analogue domain prior to sampling.

A phasor is generated by a local oscillator (LO) at the centre frequency f_c of our band of interest

$$w(t) = \mathrm{e}^{-2\pi \mathrm{j} f_\mathrm{c} t}\,. \tag{2.17}$$

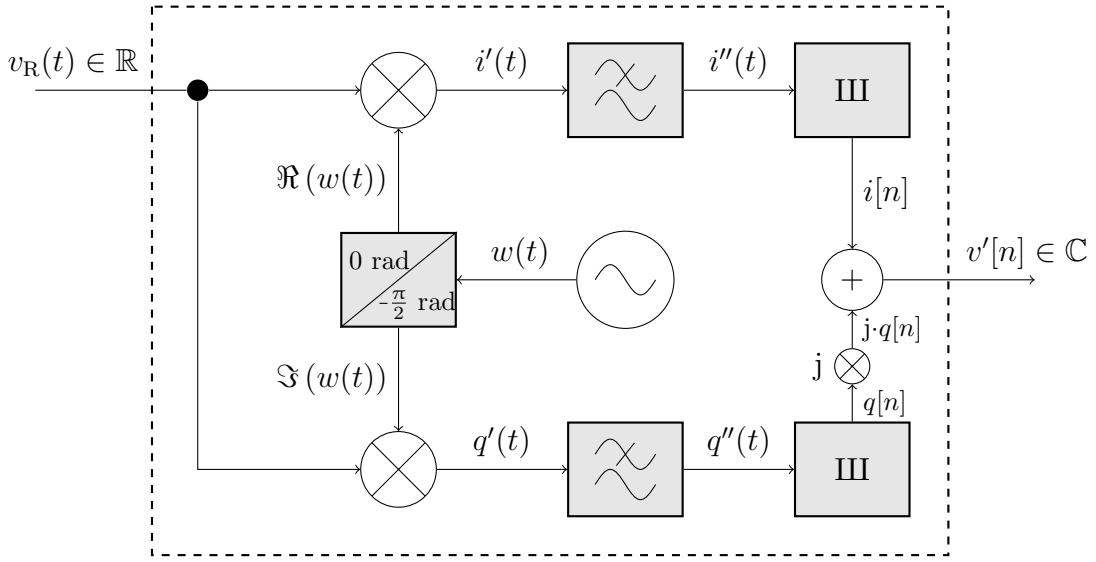


Figure 2.7. Block diagram demonstrating the main stages of how a real-valued analogue signal is converted to a complex-valued discrete signal by a digital down converter. The analogue signal is mixed with a pair of orthogonal sinusoids creating two phase-shifted versions of the signal with copies of the frequency information at the sum and difference frequencies (heterodynes). The sum heterodynes are filtered out and the signals are sampled. The two digital signals are combined to create the complex-valued output. This the equivalent of concatenating the real-valued signal with its Hilbert transform, with an additional stage of down conversion.

$\boldsymbol{v_{\mathrm{R}}}$ is mixed with the real and imaginary components[6] of the LO

$$\begin{aligned} i'(t) &= v_{\mathrm{R}}(t) \cdot \Re\left(w(t)\right) \\ &= v_{\mathrm{R}}(t) \cdot \cos(2\pi f_{\mathrm{c}} t)\,, \end{aligned} \tag{2.18}$$

$$\begin{aligned} q'(t) &= v_{\mathrm{R}}(t) \cdot \Im\left(w(t)\right) \\ &= -v_{\mathrm{R}}(t) \cdot \sin(2\pi f_{\mathrm{c}} t)\,, \end{aligned} \tag{2.19}$$

to create in-phase and quadrature components. Multiplication between two real-valued signals creates copies of the frequency content centred at the sum and difference of the frequencies [77], known as heterodynes. The sum heterodynes are removed with a low-pass filter with impulse response $\boldsymbol{h}$ for some bandwidth B over the range $f_{\mathrm{c}} \pm \frac{B}{2}$:

$$i''(t) = \left\{\boldsymbol{i'} * \boldsymbol{h}\right\}(t)\,, \tag{2.20}$$

$$q''(t) = \left\{\boldsymbol{q'} * \boldsymbol{h}\right\}(t)\,. \tag{2.21}$$

The $\boldsymbol{i''}$ and $\boldsymbol{q''}$ signals contain $\frac{\pi}{2}$ rad phase-shifted copies of the frequency band of interest centred at 0 Hz. They can now be sampled at more reasonable rates $f_{\mathrm{s}} \geq B$ to create

[6] In practice any orthogonal basis $\mathrm{e}^{\mathrm{j}(2\pi f_{\mathrm{c}} t + \phi)}$ where $\phi \in [-\pi, \pi)$ ($\frac{\pi}{2}$ rad phase-shifted sinusoids, said to be in quadrature) would suffice [77].

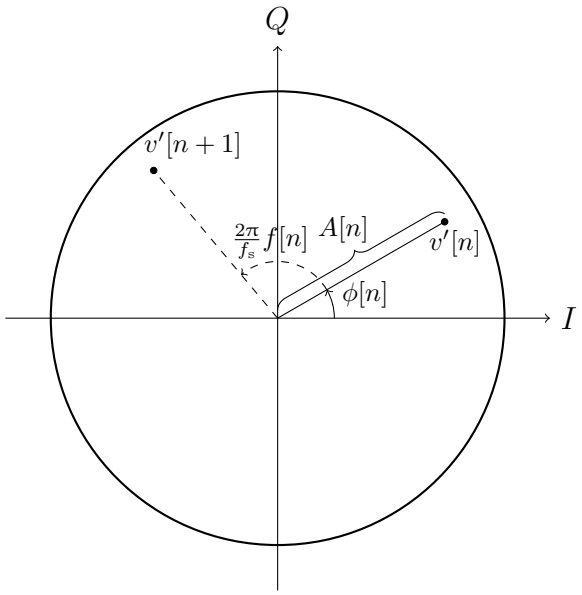


Figure 2.8. A constellation diagram demonstrating how IQ samples encode modulation parameters. A sample is plotted with the in-phase component (I) on the real axis and the quadrature component (Q) on the imaginary axis. The euclidean distance of the point from the origin gives the instantaneous amplitude $A[n]$ of the signal envelope. The angle relative to the positive real axis gives the instantaneous phase $\phi[n]$. A change in phase over time indicates the frequency $f[n]$ is non-zero, i.e. a deviation from f_c.

discrete-time signals

$$i[n] = i''(n\tau)\,, \tag{2.22}$$
$$q[n] = q''(n\tau)\,, \tag{2.23}$$

and combined into a complex result $\boldsymbol{v'} \in \mathbb{C}^{\mathbb{Z}}$

$$v'[n] = i[n] + \mathrm{j} \cdot q[n]\,. \tag{2.24}$$

The process is summarised in Figure 2.7. The complex samples are often known as IQ samples, where the I and Q refer to the in-phase and quadrature components respectively:

$$i[n] = \Re\left(v'[n]\right)\,, \tag{2.25}$$
$$q[n] = \Im\left(v'[n]\right)\,. \tag{2.26}$$

Plotted on a constellation diagram, we can see how they encode the modulation parameters (Figure 2.8). As Cartesian coordinates relative to the origin, the IQ samples encode the instantaneous amplitude of the signal envelope in their magnitude, and the phase $\phi \in [-\pi, \pi)$ relative to that of the LO in the angle from the real axis. The modulation

parameters are recovered from the IQ samples with:

$$A[n] = |v'[n]| = \sqrt{i[n]^2 + q[n]^2}\,, \tag{2.27}$$

$$\phi[n] = \angle v'[n] = \operatorname{arctan2}\left(q[n],\, i[n]\right)\,, \tag{2.28}$$

$$f[n] = \angle \frac{v'[n]}{v'[n-1]} \cdot \frac{f_\mathrm{s}}{2\pi \text{ rad}}\,, \tag{2.29}$$

where

$$\operatorname{arctan2}\left(y,\, x\right) = \begin{cases} 2 \cdot \arctan\left(\frac{y}{x+\sqrt{x^2+y^2}}\right) & x > 0 \vee y \neq 0 \\ -\pi & x < 0 \wedge y = 0 \\ \text{undefined} & x = 0 \wedge y = 0 \end{cases} \tag{2.30}$$

is the two-argument variant of the inverse tangent function for $x, y \in \mathbb{R}$ such that $\operatorname{arctan2}\left(y,\, x\right) \in [-\pi,\, \pi)$. Modulation of frequency around f_c causes the samples to orbit the origin. We can now see how if $|f[n]| > 0.5 \cdot f_\mathrm{s}$ aliasing will occur as the samples will advance by more than π rad/sample and appear to rotate at a different frequency $2\pi(\hat{f} \cdot {f_\mathrm{s}}^{-1} + k)$ where $k \in \mathbb{Z} \setminus 0$. But by appropriately filtering the bandwidth prior to sampling we do not need to concern ourselves with aliases. Down conversion is a linear process i.e. it works the same for sums of frequencies.

2.4.3 Software-defined radio front-ends

An SDR front-end implements digital down conversion, usually using an analogue RF front-end, analogue-to-digital converters, and an ASIC or FPGA. Let us define an SDR in mathematical terms as a linear operator

$$\mathrm{D} : \mathbb{R}^\mathbb{R} \to \mathbb{C}^\mathbb{Z}\,, \tag{2.31}$$

where a real function of continuous time is transformed to a complex function of discrete time. For some centre frequency f_c and bandwidth B, we can see this mathematically as the equivalent to

$$\mathrm{D}\{\boldsymbol{v_\mathbf{R}}\} = \text{Ш}_{\boldsymbol{\tau}} \cdot \left(\mathrm{LPF}_{0.5B}\left\{\{\boldsymbol{v_\mathbf{R}} + \mathrm{j} \cdot \mathcal{H}\left\{\boldsymbol{v_\mathbf{R}}\right\}\} \cdot \boldsymbol{w}\right\}\right) \tag{2.32}$$

for some sampling period τ.

I used Ettus Research's Universal Software Radio Peripheral (USRP) platform with the USRP hardware driver (UHD) on a Linux platform to collect data for experiments. To identify the practical considerations presented by using the hardware I used a Hewlett Packard E4432B ESG-D Series RF signal generator connected by coaxial cable to the SDR device under test (DUT). To ensure there is no discrepancy between the LOs in the signal generator and DUT, the DUT shares the signal generator's 10 MHz reference clock. Sharing a reference clock will ensure both devices measure frequencies identically during testing. No preamplifier was used.

Front-end hardware

I inspected two SDR front-ends which I used to collect data. The B200 is an SDR front-end that can be used to perform low-cost video eavesdropping. It can record up to $B = 56$ MHz and has a tuning range of 70 MHz to 6 GHz. It is powered by, and communicates with the host over, a USB 3.0 connection. It cost circa 500 GBP, making it of particular interest for amateur eavesdroppers. Its portable form factor makes it a suitable device for mobile eavesdropping.

The X300 is a high-end SDR front-end. It has a maximum sample clock rate of 200 MHz, of which the selected f_s must be some integer factor. I used a UBX-160 daughterboard with gives up to $B = 160$ MHz. It communicates with the host over a 10 Gbit/s fibre-optic Ethernet connection.

Magnitude calibration

The UHD returns packets of timestamped[7] complex samples $v'[n]$ as pairs of 16 bit integers. While it can additionally output floating-point numbers, these are simply the short integer values divided by 2^{15}. USRP devices are uncalibrated [78] and the UHD does not provide a unit for the complex samples it outputs. To obtain a real-world unit for discussion the devices must be calibrated manually. I set the generator to output a sinusoid with power $P = -30$ dBm at frequency matching the DUT receive frequency $f_{\mathrm{c,TX}} = f_{\mathrm{c,RX}}$. We compute

$$V_\mathrm{rms} = \sqrt{P \cdot Z} \tag{2.33}$$

for our system with impedance $Z = 50\ \Omega$. This is divided by the mean amplitude value to calculate the voltage per unit of magnitude

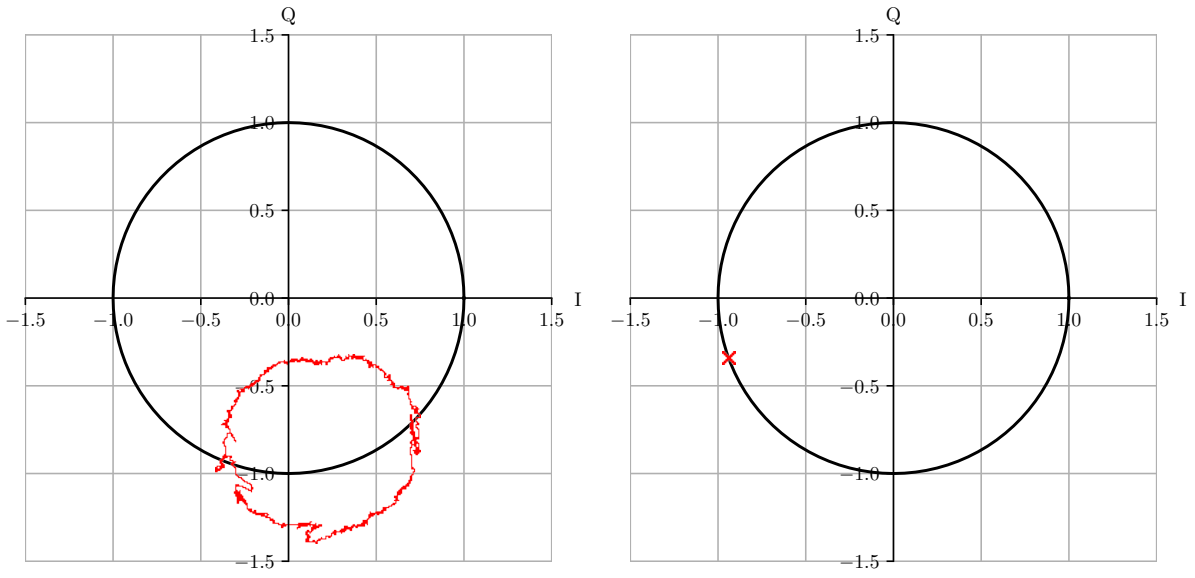
$$a = N \cdot \frac{V_\mathrm{rms}}{\sum_{n=0}^{N-1} |v'[n]|}\,. \tag{2.34}$$

Now a can be used as a multiplier for samples output by the DUT to obtain

$$v[n] = a \cdot v'[n]\,, \tag{2.35}$$

which is defined in terms of a real-world unit, in my case microvolts. Multiplication between a complex number and a scalar is a linear operation, so this only affects the magnitude of the complex sample. I found the value for a can vary as the front-end's parameters are adjusted (e.g. f_c). Therefore I performed magnitude calibration for each experimental configuration.

[7]The early USRP1 device fakes the timestamps by using the computer clock. As such it is not recommended for time-sensitive experiments.



(a) $f_o = 0$ Hz **(b)** $f_o = 1$ MHz

Figure 2.9. When there is no frequency offset between the front-end's LO and the frequency of interest, noise from the LO in the SDR front-end leaks into the signal introducing erroneous phase information. This adversely affects applications where phase is important. When down converting via an offset f_o, shifting to baseband is performed in two stages: the LO is used to down convert f_c to $-f_o$, then a second round of frequency conversion in software shifts $-f_o$ to DC. This introduces no new noise, and the noise component introduced by the LO has been moved to f_o, away from our frequency of interest where it can be filtered out.

LO leakage

By default both SDR front-ends perform direct down conversion. That is, the LO operates at a frequency close to the frequency of interest instead of down converting via some intermediate frequency (IF). Let us plot the constellation diagram of $\boldsymbol{v}$ after applying a Chebyshev Type II (to minimise passband amplitude distortion) low-pass filter[8] using zero-phase filtering with cut-off frequency $\overline{f} = 20$ Hz

$$\widetilde{\boldsymbol{v}} = \text{cheby2}_{\overline{f}}\{\boldsymbol{v}\} \tag{2.36}$$

and normalise by the median amplitude

$$\mathring{v}[n] = \frac{\widetilde{v}[n]}{\text{median}\{|\widetilde{v}[m]|\}}\,. \tag{2.37}$$

The constellation diagram shows unusual behaviour around DC, where the frequency of interest is located. I expected $\mathring{\boldsymbol{v}}$ to produce a stationary point with some arbitrary phase offset as the filter removes noise around the test sinusoid. Instead, I observed an unusual

[8] I implemented this using three cascaded biquadratic (second-order) filters to minimise numerical instability.

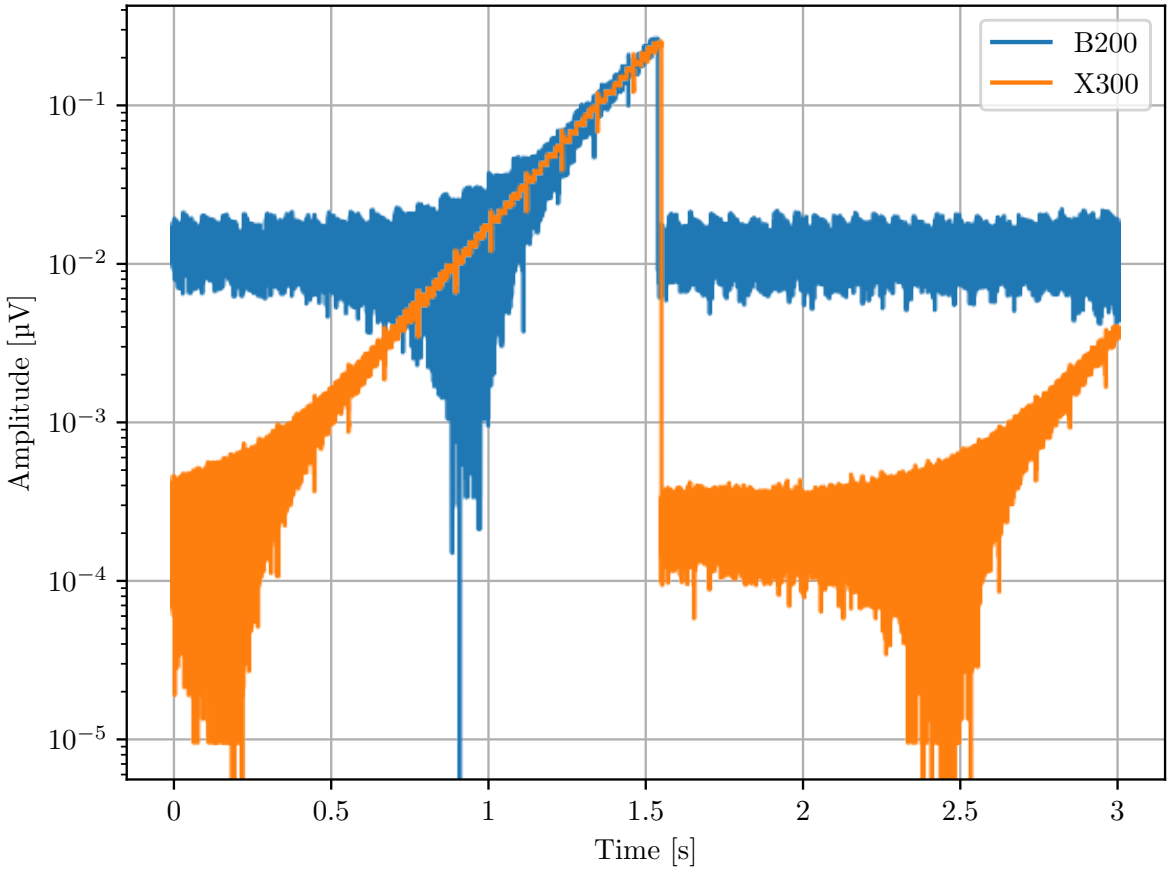


Figure 2.10. A linear magnitude response by the front-end amplifier is desirable. Plotting the magnitude of the IQ samples from the Ettus USRP B200, and X300 with UBX-160 daughterboard ($f_c = 428$ MHz, $B = 10$ MHz, 0 dB gain) for a range of input amplitudes, there is a linear response of the curves produced by the front-ends, showing distortion is not introduced. This means we do not need to correct amplitudes for different input powers. We can also see the much greater dynamic range afforded by the USRP X300 when using the UBX-160 daughterboard.

circular trajectory (Figure 2.9a). I determined the artefact was the DUT's LO leaking into $\boldsymbol{v}$ during the analogue mixing stage, an effect known as *reciprocal mixing* [79]. To counter this we can down convert via an IF. I offset the LO's operating frequency from f_c by $f_o = 1$ MHz[9]. The spectrum is first down converted via the IF $f_{c,RX} = f_c + f_o$ and digitised. The front-end then performs a second stage of frequency conversion in software using the inbuilt DSP, shifting the information contained at $-f_o$ to 0 Hz, and the LO leakage to f_o. The DSP frequency shift does not introduce new artefacts. When I repeated the experiment, the LO phase noise, now contained outside of the bandwidth at f_o, was attenuated by the low-pass filter, resulting in the expected stationary point in the constellation diagram (Figure 2.9b). Therefore when performing operations where phase information is important, one must ensure the front-end's LO frequency is appropriately offset from the frequency of interest.

[9]UHD function *set_rx_freq* with *tune_request_t* object argument.

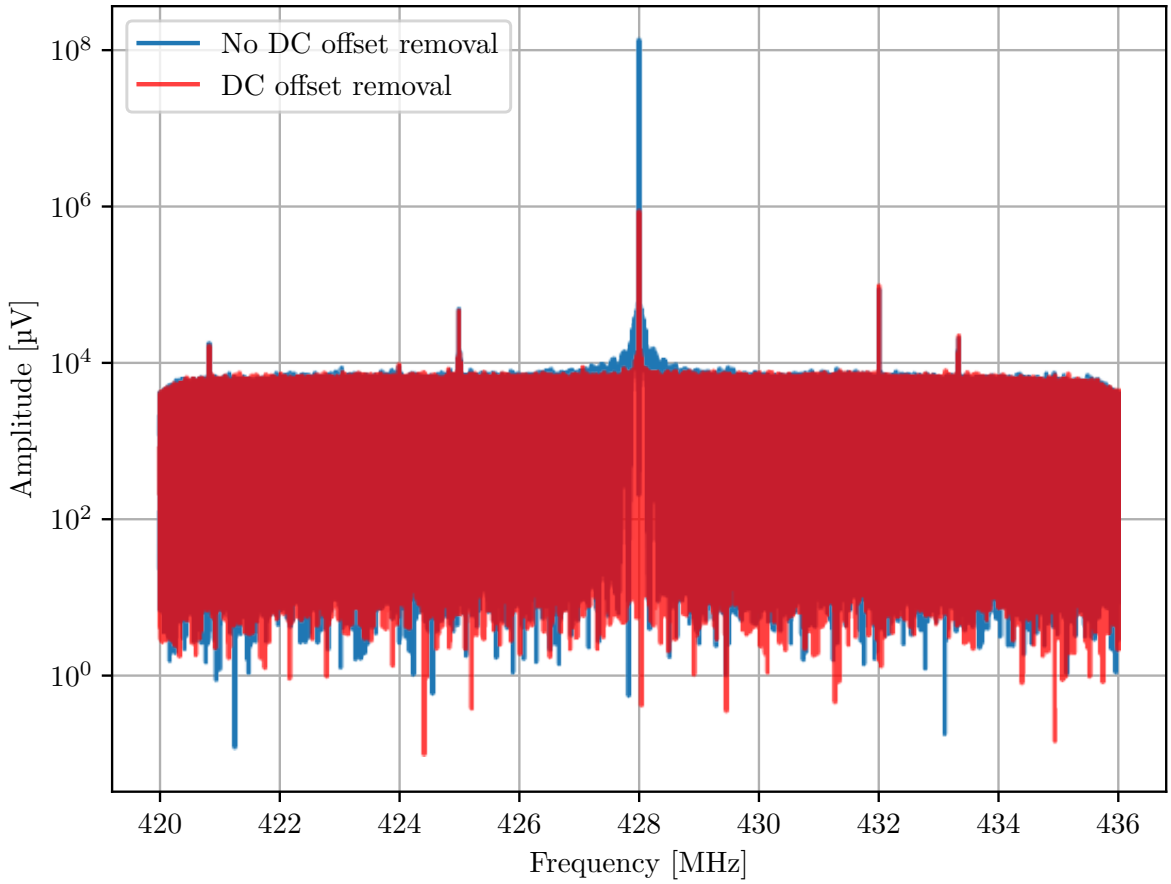


Figure 2.11. Enabling automatic DC offset removal subtracts out the long-term average of the signal to improve the SNR. However, it distorts frequencies at f_c. Therefore when making use of offset removal we must ensure the frequency of interest is offset from the centre frequency.

Amplifier linearity

To investigate how the amplifier in the front-end responds over different amplitudes I performed amplitude sweeps over the range:

$$P = \{-100, -99, \ldots, 0\} \text{ dBm}\,, \tag{2.38}$$

with a dwell time between amplitude changes of 10 ms. I selected this range to include feasible amplitudes for the test environment, from the noise floor of approximately −60 dBm, up to the maximum input for which the DUT is rated. When plotted, the recorded magnitude (Figure 2.10) demonstrates a linear curve rising out of the noise floor on a logarithmic scale, showing the amplifier exhibits a linear response over the region of interest. This is desirable behaviour as it means we do not need to perform correction.

DC offset removal

By default the UHD performs DC offset removal by subtracting out the long-term average of the signal to reduce noise near DC, and thus improve the SNR. This heavily distorts the

amplitude at $f_{\mathrm{c,RX}}$, as can be seen in Figure 2.11, where the peak at DC and the sidebands are significantly attenuated when offset removal is turned on. The spurs elsewhere in the frequency range remain unattenuated. Disabling the option in the driver[10] reduces the SNR. Therefore we should ensure the frequency of interest is not exactly at f_{c}.

2.5 Resampling

A VDU has three operational parameters most relevant to eavesdroppers: the rate of pixel transmission—the pixel clock $f_{\mathrm{p}} \in \mathbb{R}^{+}$, the transmission rate for a complete row of pixels—the line rate $f_{\mathrm{h}} \in \mathbb{R}^{+}$, and the frame rate $f_{\mathrm{v}} \in \mathbb{R}^{+}$. They are usually related via integer ratios:

$$f_{\mathrm{h}} = \frac{f_{\mathrm{p}}}{w_{\mathrm{T}}}\,, \tag{2.39}$$

$$f_{\mathrm{v}} = \frac{f_{\mathrm{h}}}{h_{\mathrm{T}}}\,, \tag{2.40}$$

where a frame of video signal, including the blanking interval, is $w_{\mathrm{T}} \in \mathbb{N}$ pixels across and $h_{\mathrm{T}} \in \mathbb{N}$ lines high. As f_{p}, f_{h}, and f_{v} are real-valued parameters subject to error and drift (Chapter 4), I select f_{p} together with the constant terms w_{T} and h_{T} to characterise a device using the notation:

$$w_{\mathrm{T}} \otimes h_{\mathrm{T}} @ f_{\mathrm{p}}\,, \tag{2.41}$$

denoted the *full video mode*, from which one can accurately recover f_{h} and f_{v} using Equations 2.39 and 2.40. This notation is distinct from the displayed video mode, presented using the notation:

$$w_{\mathrm{D}} \times h_{\mathrm{D}} @ f_{\mathrm{v}}\,, \tag{2.42}$$

defined using the width w_{D} and height h_{D} of the content area—the addressable subset of the full video mode excluding the blanking interval. This notation is often used to describe screen resolution and refresh rate. For example the Video Electronics Standards Association (VESA) SMT mode 640×480@60 Hz has the full video mode 800⊗525@25.2 MHz.

To reconstruct a raster of emanated screen content within $\boldsymbol{v}$, we must first resample from f_{s} to f_{p} (some other integer multiple of f_{h} would also suffice) with the resampling ratio:

$$R = \frac{f_{\mathrm{s}}}{f_{\mathrm{p}}}\,. \tag{2.43}$$

When $R \notin \mathbb{N}$ or $(f_{\mathrm{s}} < f_{\mathrm{p}})$, we first upsample by a factor[11]

$$L = \frac{\mathrm{LCM}\,(f_{\mathrm{s}}, f_{\mathrm{p}})}{f_{\mathrm{s}}} \tag{2.44}$$

[10] UHD function *set_rx_dc_offset*.

[11] When the lowest-common-multiple of f_{s} and f_{p} is very large or does not exist a more tractable approximation can be used. Precise correction of the resampling ratio is discussed in §4.4.

by zero stuffing the signal

$$\hat{v}[n] = \begin{cases} v\left[\frac{n}{L}\right] & n \bmod L = 0 \\ 0 & \text{otherwise} \end{cases}. \tag{2.45}$$

This increases the number of samples whilst keeping the spectral data within the bandwidth the same. It does however create copies (aliases) of the spectral data at higher frequencies, which we remove with a zero-phase finite impulse response (FIR) low-pass filter:

$$\check{v}[n] = \mathrm{FIR}_{0.5f_\mathrm{s}}\left\{\hat{\boldsymbol{v}}\right\}[n]. \tag{2.46}$$

Now, if $L \cdot f_\mathrm{s} > f_\mathrm{p}$ for $\check{\boldsymbol{v}}$, we downsample by

$$M = \frac{\mathrm{LCM}(f_\mathrm{s}, f_\mathrm{p})}{f_\mathrm{p}} \tag{2.47}$$

to obtain

$$w[n] = \check{v}[nM], \tag{2.48}$$

where now $f_\mathrm{s} = f_\mathrm{p}$, and each sample corresponds approximately to one pixel in the video signal[12]. In this case we are estimating the true value of f_p, which can be affected by various physical factors (Chapter 4). The error in the estimate will cause the video content to be resampled into an incorrect number of pixels. Instances of incorrect [18, 20, 22, 64], or lack of [15], resampling have distorted the results of many published experiments. A small error will cause the content to drift over multiple frames at a velocity proportional to the error magnitude unless corrected (Chapter 4), while a large error will sheer or distort the frame completely (Chapter 3).

2.6 Demodulation and rastering

We can now demodulate and raster $\boldsymbol{w}$. The digital down conversion process bandpass filters the emanated video content. The characters output by the TMDS algorithm will form stable cycles over regions of uniform intensity as TMDS attempts to DC balance the signal (Appendix A), hence we can expect the modulation parameters to be periodic over uniform regions. Let us define the image buffer[13] $\boldsymbol{b} \in \mathbb{R}^{w_\mathrm{T} \times h_\mathrm{T} \times 3}$ as a $w_\mathrm{T} \times h_\mathrm{T}$ array of red, green, and blue values.

2.6.1 Amplitude information

The simplest approach to visualising the received and rasterised signal is to take the absolute value of $\boldsymbol{w}$ as a brightness value, discarding the phase information. This provides

[12] In practice, polyphase filter banks are used to avoid (2.46) producing samples which (2.48) then discards.

[13] I used Matplotlib as the rendering library.

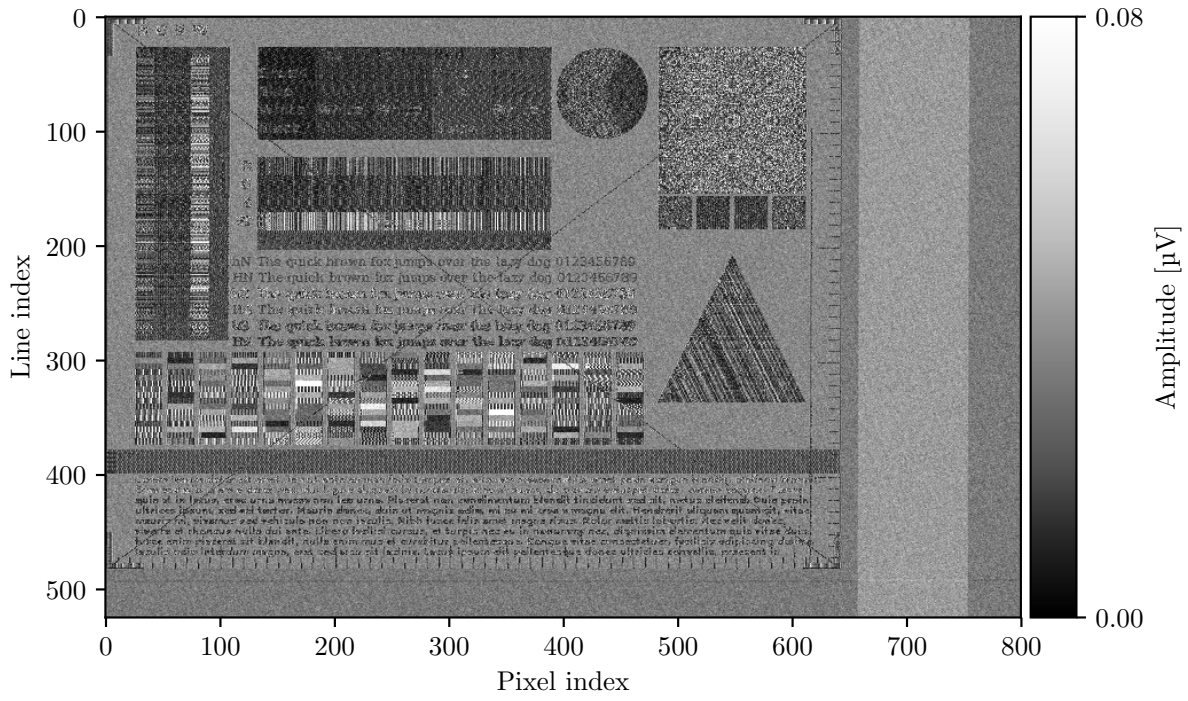


Figure 2.12. By discarding the phase information and rastering the amplitude of the IQ samples we obtain a reasonable approximation of screen content, here of the test card in Appendix B. Colour information is lost as we record the superposition of the red, green, and blue channel emanations. Areas with uniform colour and intensity are best approximated as the TMDS algorithm produces character cycles, whereas gradients tend to produce barcode like patterns due the changing signal. Areas where there is a colour change, but no overall intensity change, are often difficult to discern from each other. The content area is bordered by the blanking interval. A *data island* can be seen as a brighter band running the length of the blanking interval in which additional data, notably 16 bit audio encoded using TERC4, can be transmitted [71]. The content area exhibits a very slight slanting effect due to the parameters used during the resampling stage being inexact.

a reasonable approximation of transmitted content for areas with uniform intensity as the TMDS DC-balancing cycles form short periodic regions of bit transitions. We scale the demodulated brightness value by some user-defined contrast U_C and brightness U_B parameters:

$$C_\mathrm{V}[n] = |w[n]| \cdot U_\mathrm{C} + U_\mathrm{B}\,, \tag{2.49}$$

and set pixel at coordinates (x, y), relative to the top left-hand corner, to

$$b[x, y, i] = C_\mathrm{V}[y \cdot w_\mathrm{T} + x] \quad \forall i \in \{\mathrm{R},\, \mathrm{G},\, \mathrm{B}\} \tag{2.50}$$

creating a greyscale representation. In Figure 2.12 we can see how the brightness tends to remain constant over areas of uniform intensity, or form short cycles. Colour information is lost as the emanations from the three RGB transmission lines form a superposition [20, 23, 27, 80], making the specific colour indistinguishable upon reception at such narrow

bandwidths. The emanated amplitude is a complex function of the voltages expressed by the transmission lines [81]. We can see this in the RGB colour text boxes, where the sum of the colour intensities have roughly uniform brightness in the raster making it difficult to discern the text from the background, despite the change in colour in the test card. Relative intensities and colour gradients too are lost, rastering as barcode-like patterns. I draw a comparison here with the constant-colour triangle. Like the coloured text boxes it has uniform intensity, but gradiated colouring. It rasters as a striped pattern instead of a uniform area. This is due to the higher emanated power from the red transmission lines. Other red features, such as the gradient bars and the random brightness boxes, are more distinct than the blue and green counterparts. In the constant-colour triangle the barcode pattern moves away from the red corner as the level of red diminishes, changing the transmitted bit pattern. The stronger changing emanations are easier to see in the rastered feature. The stronger emanations from the red channel are an artefact of this particular hardware setup and may vary between devices.

In addition to HDMI/DVI cables using TMDS, I also performed eavesdropping experiments on VDUs using MIPI Display Serial Interface, VGA cables, FPD-Link with low-voltage differential signalling (LVDS), and cathode ray tube VDUs. I found they could all be eavesdropped in a similar manner using amplitude rasters to estimate screen content.

Dynamic range compression

The dynamic range of a signal defines the ratio between the smallest and largest amplitude values

$$\breve{w} = 20 \text{ dB} \cdot \log_{10} \left(\frac{\max \{|w[n]|\}}{\min \{|w[n]|\}} \right) . \tag{2.51}$$

The UHD returns 15 bits of magnitude information per sample from the SDR front-end, giving an available dynamic range of 90.3 dB. An example recording $\boldsymbol{w}$ has a dynamic range of $\breve{w} = 85.62$ dB, showing most of the dynamic range provided by the SDR front-end has been utilised. The rasterisation process necessitates the re-quantisation of values to 8 bit values, giving 2^8 greyscale values, a dynamic range of just 48.2 dB. This can result in loss of information. In Figure 2.13 I plot a histogram of the sample amplitudes in $\boldsymbol{w}$. The majority of the signal information is contained at lower amplitudes. A few outliers artificially inflate the dynamic range of the signal. This will increases the level of quantisation noise introduced during rasterisation as more greyscale values will be used to represent the infrequent higher brightnesses. Furthermore, the blanking interval forms an area of generally higher amplitudes, with some overlap between the two. To improve the visualisation we can compress the range of amplitudes represented in the signal to better represent relevant information at a coarser level of quantisation. For amplitude-demodulated samples taken from within the content frame

$$\boldsymbol{w_c} = \{|w[n]| \, : \, n \bmod w_\text{T} < w_\text{D} \wedge n \bmod w_\text{T} \cdot h_\text{T} < w_\text{T} \cdot h_\text{D}\} \, , \tag{2.52}$$

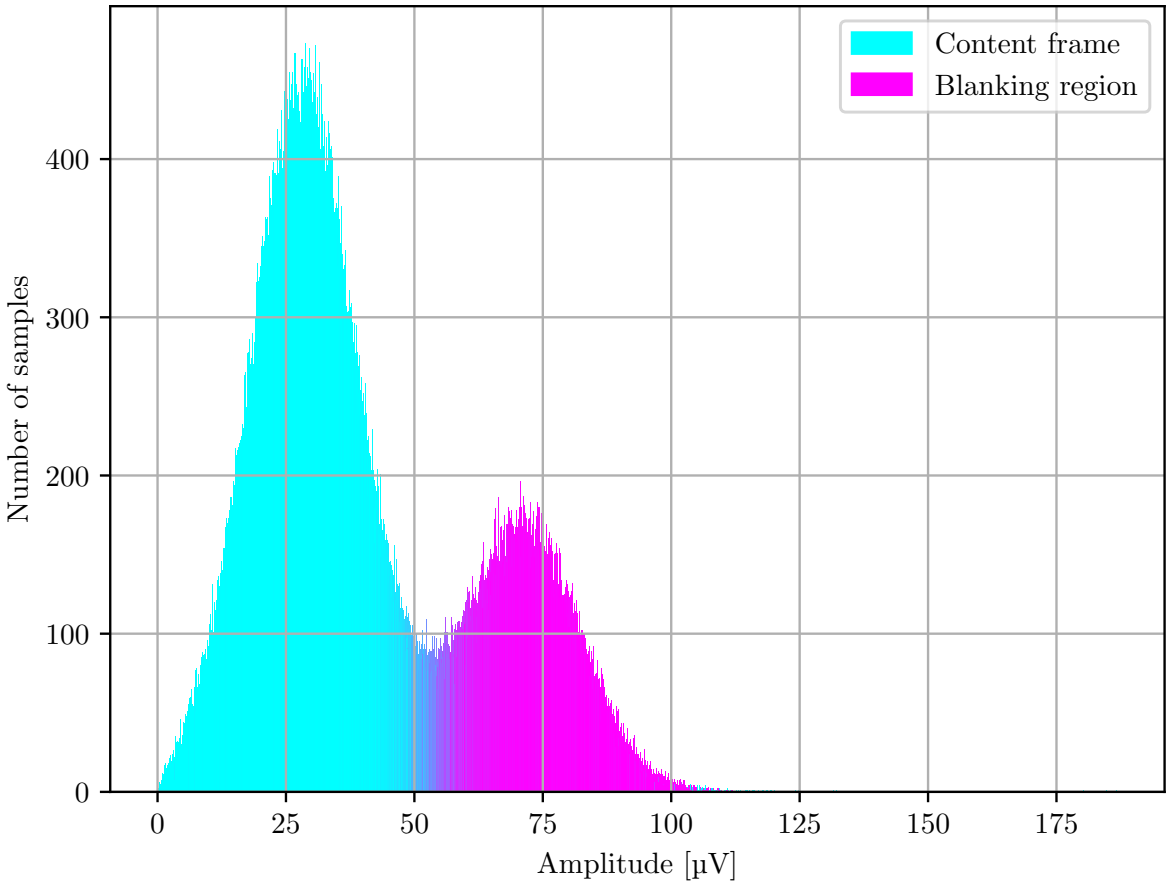


Figure 2.13. The histogram of amplitudes present in a frame of an eavesdropped video signal forms a bimodal distribution. The blanking interval is encoded using TMDS characters with more transitions, resulting in generally increased emanated power and hence amplitude in the down converted and bandpassed IQ samples. There is significant overlap between the two modes. This can be seen here, where bins have been coloured according to the proportion of each class they contain. Occasional outliers produce many bins at higher amplitudes with little-to-no data. Rastering the amplitude information for this signal will result in many of the 2^8 greyscale values being assigned to represent infrequent or not visually useful information. By measuring the variance of the content samples, we can clip the amplitudes to better utilise the limited number of brightness values available.

where $w[0]$ corresponds to the first top left sample of the content area, we compute the sample variance of $\boldsymbol{w}_\mathbf{c}$:

$$s_\mathrm{c}^2 = \frac{1}{N}\sum_{n=0}^{N-1} w_\mathrm{c}[n]^2 - \overline{w}_\mathrm{c}^2\,, \tag{2.53}$$

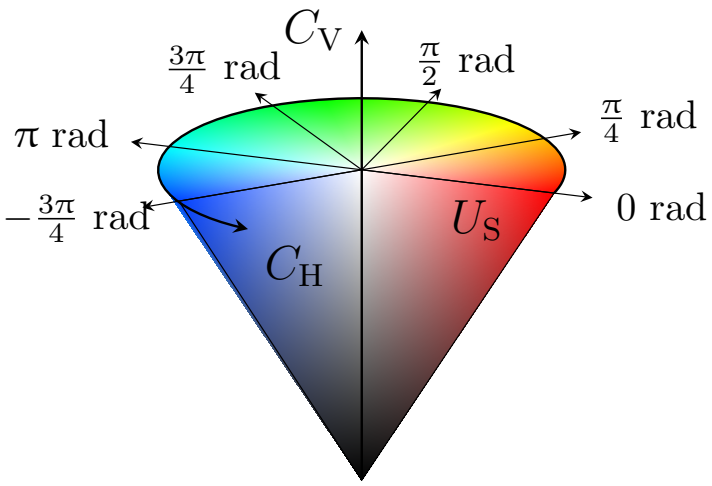


Figure 2.14. The HSV colour space can be used to represent both the amplitude and phase information encoded by a complex-valued sample in a single pixel. The hue parameter is used to represent the phase component whilst the value parameter encodes the amplitude. Saturation can be set according to user preference, and when set to zero the encoding scheme reduces to the amplitude-only scheme.

where $\overline{w}_{\mathrm{c}}$ is the sample mean of $\boldsymbol{w}_{\mathbf{c}}$. We then clip samples that fall further than three standard deviations from the mean of the content region:

$$w'[n] = \begin{cases} \overline{w}_{\mathrm{c}} + 3s_{\mathrm{c}} & w_{\mathrm{c}}[n] > \overline{w}_{\mathrm{c}} + 3s_{\mathrm{c}} \\ \overline{w}_{\mathrm{c}} - 3s_{\mathrm{c}} & w_{\mathrm{c}}[n] < \overline{w}_{\mathrm{c}} - 3s_{\mathrm{c}} \\ w[n] & \text{otherwise} \end{cases} \tag{2.54}$$

Assuming the brightness values follow a Gaussian distribution, this would clip to the upper and lower 0.03% of amplitudes in the content frame, and remove unused data bins. This reduces the dynamic range of the example recording to $\breve{w}' = 76.32$ dB. The data can then be scaled linearly to fill the available greyscale values using

$$U_{\mathrm{C}} = \frac{255}{6s_{\mathrm{c}}} \tag{2.55}$$

$$U_{\mathrm{B}} = \frac{-255\left(\overline{w}_{\mathrm{c}} - 3s_{\mathrm{c}}\right)}{6s_{\mathrm{c}}}, \tag{2.56}$$

reducing the level of data loss caused by quantisation to 8 bit.

2.6.2 Phase information

We optionally can also incorporate the phase information $\angle w[n]$ into the raster using the hue-saturation-value (HSV) colour model [82], represented by a cone (Figure 2.14). Relative to the central axis the hue $C_{\mathrm{H}} \in [-\pi, \pi) \cdot 1$ rad is determined by the angle; the saturation level $S \in [0, 1)$ by the radial distance; and the brightness $C_{\mathrm{V}} \in [0, 1)$ by the

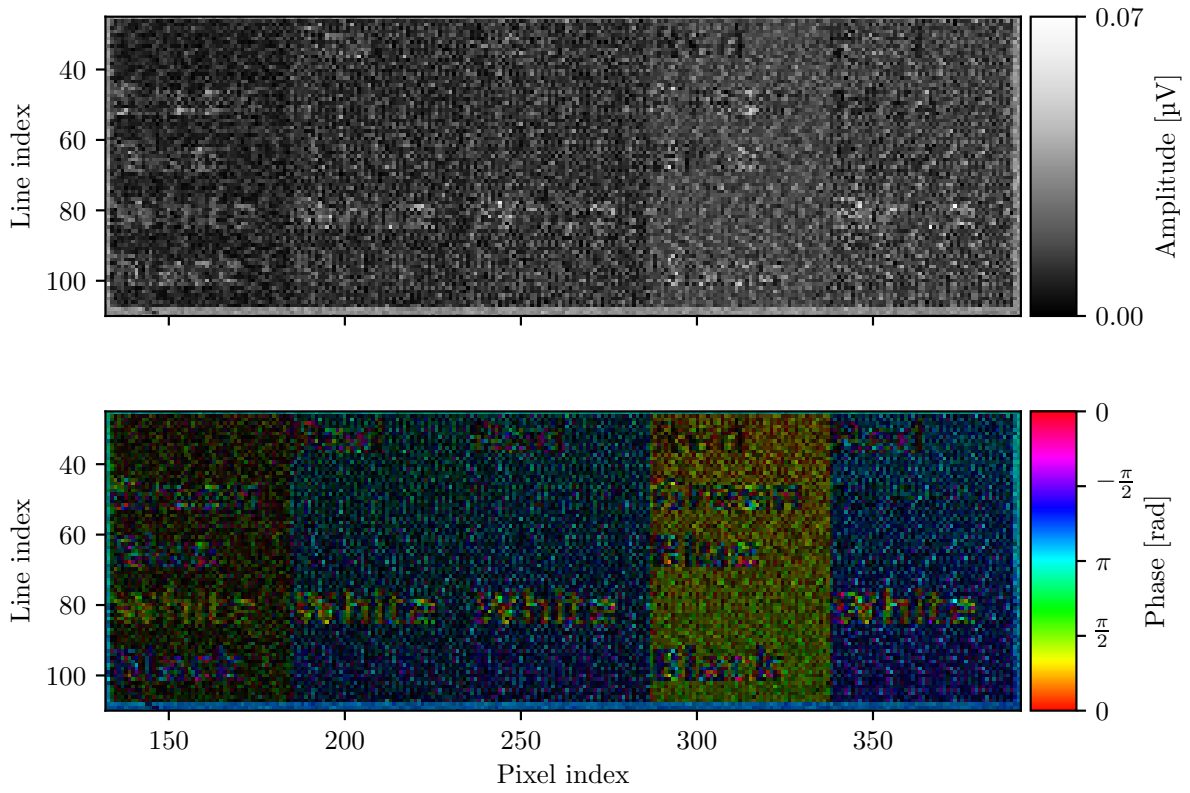


Figure 2.15. When we use hue to represent phase in the raster, having corrected the offset between f_c and f_p (§4.2), some features not visible in the amplitude-only raster become visible. Here we compare how the coloured text boxes change appearance when incorporating phase information as the hue. Text not previously readable can now be observed due to a large change in phase, improving the clarity of the feature.

parallel position. We represent the phase of $\boldsymbol{w}$ with the hue

$$C_H[n] = \angle w[n]\,. \tag{2.57}$$

Brightness uses the amplitude-demodulated value C_V from Equation 2.49. Saturation is left as a user-defined variable U_S according to preference. When $U_S = 0$, this method is equivalent to amplitude-only rasterisation. Together they create an HSV tuple for pixel (x, y), which we convert to RGB:

$$b[x, y] = \text{toRGB}(C_H[y \cdot w_T + x],\, U_S,\, C_V[y \cdot w_T + x])\,. \tag{2.58}$$

Incorporating phase information into the raster allows visual inspection of the phase changes as the video signal progresses. Figure 2.15 compares how the colour boxes appear when using HSV after correcting for the difference between f_c and f_p (§4.2). Like the magnitude, phase is uniform across the sections of constant colour. However features previously unobservable when $U_S = 0$ can now be seen more clearly as the phase change is more pronounced than that of the amplitude. Using a restricted colour palette to produce similar emission amplitudes has previously been proposed as a soft countermeasure [27,

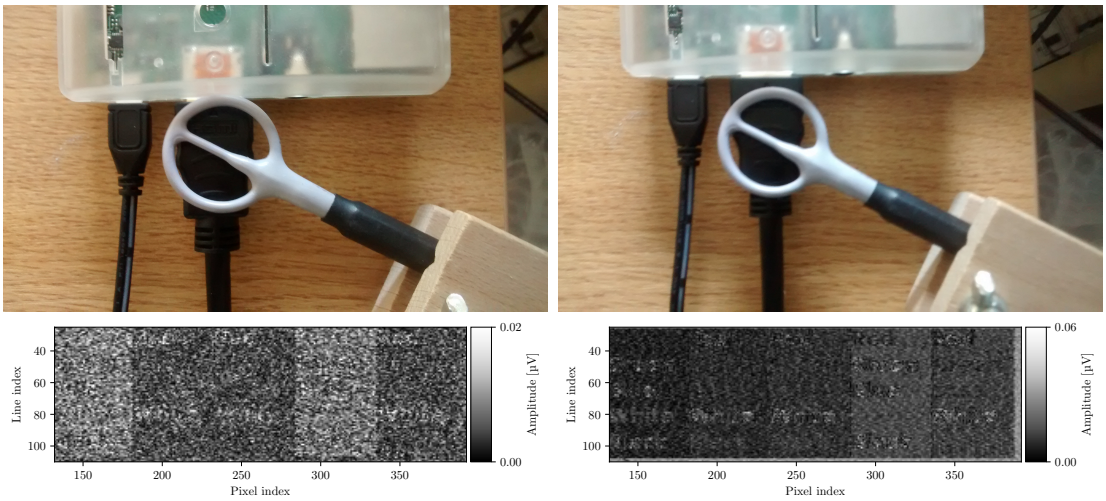


Figure 2.16. Comparing rasters before and after a small change in antenna position relative to the emanation source can produce a change in raster appearance. Here, the relative intensities between various features have changed, with some becoming brighter or darker relative to others, while others become easier or more difficult to differentiate between. The effect can be seen here where the red colour box (Figure B.1) is brighter than others in (a), but darker in (b). Attempts to use these rasters to subtract out the signal component were met with failure, and often produced a clear inverse image in the result.

83], yet we can see in Figure 2.15 incorporating phase information after phase unrotation can help subvert such measures. Without correcting for the difference, rainbow banding will appear (Figure 4.3). Using phase does not, however, help to discern gradients as their phase is constantly modulated by the changing characters.

2.7 Signal-to-noise ratio estimation

To obtain an objective measure of raster quality we need to estimate the SNR. I initially attempted to obtain a noise-minimal template signal using the near-field probe which could be subtracted out from measurements leaving only the noise component. I found the relative brightness between features in the raster was dependant on the position of the antenna. I demonstrate this effect in Figure 2.16. Adjusting the probe position slightly between measurements causes a change in the appearance of the raster. This supports the findings of Hayashi *et al.* [4] and Zhang *et al.* [23] who found the position of the probe relative to the cable connector influenced whether a video signal could be recovered or not. Being unable to guarantee the same relative brightness between recordings I was unable to subtract out the signal component. Such issues are not uncommon for EmSec measurements [84]. Let us instead assume $\boldsymbol{v}$ consists of a superposition of some noise-free

signal component

$$\boldsymbol{s} \sim \mathcal{N}(\mu,\, 0)\,, \tag{2.59}$$

and additive white Gaussian noise (AWGN) background noise

$$\boldsymbol{n} \sim \mathcal{N}(0,\, \sigma^2)\,, \tag{2.60}$$

Multiple observations of

$$\boldsymbol{v} = \boldsymbol{s} + \boldsymbol{n} \tag{2.61}$$

are independent and identically distributed (i.i.d.). Given two recordings of an emanating source $\boldsymbol{v_A}$ and $\boldsymbol{v_B}$ taken from the same position in the environment relative to the target device, when the signal components are aligned we can state that

$$\boldsymbol{v_A} + \boldsymbol{v_B} = \mathcal{N}\left(2\mu,\, 2\sigma^2\right)\,, \tag{2.62}$$

where the variances add due to the i.i.d. property. If we subtract them, the means subtract but the variances still add

$$\boldsymbol{v_A} - \boldsymbol{v_B} = \mathcal{N}\left(0, 2\sigma^2\right)\,. \tag{2.63}$$

We estimate the SNR ϱ as the ratio between the mean of the signal-plus-noise component, and the standard deviation of the noise-only component:

$$\varrho = \frac{\sqrt{2}}{2} \cdot \frac{|\mathbb{E}\left[\boldsymbol{v_A} + \boldsymbol{v_B}\right]|}{\sqrt{\mathbb{V}\left[\boldsymbol{v_A} - \boldsymbol{v_B}\right]}} \tag{2.64a}$$

$$= \frac{\sqrt{2}}{2} \cdot \frac{2\mu}{\sqrt{2}\sigma} \tag{2.64b}$$

$$= \frac{\mu}{\sigma}\,. \tag{2.64c}$$

Multiplying by a factor of $\frac{\sqrt{2}}{2}$ corrects the amplitude and variance scaling. The result can be expressed in decibels using:

$$\varrho_{\mathrm{dB}} = 20\ \mathrm{dB} \cdot \log_{10} \varrho\,, \tag{2.65}$$

where the logarithm term is multiplied by a factor of 20 dB as ϱ is a field ratio.

2.8 Simulating uniform wave propagation

The primary metric of interest for those wishing to protect against eavesdropping is the distance at which an attack can be successfully performed. Often this information is used to create a controlled zone. Countermeasure designers use limit curves, such as those in SDIP-27/1, to place a device carrying RED data a minimum distance from uncontrolled space, allowing emissions to dissipate to an acceptable level before they can be intercepted by would-be eavesdroppers. The main advantage of this approach is its low

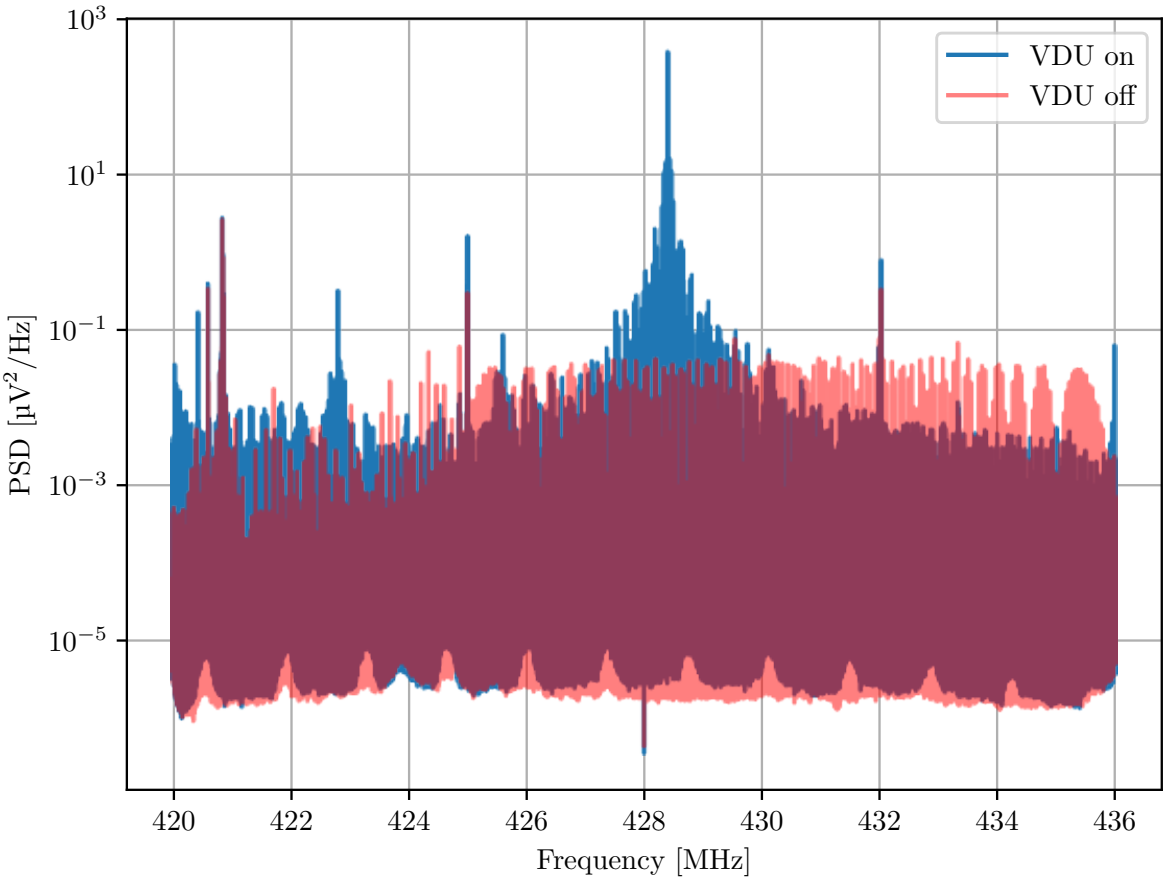


Figure 2.17. Plot of the PSD, produced using Welch's method, of $\boldsymbol{v_0}$ ($d_0 = 0.25$ m) containing an emanating 800⊗525@25.2 MHz video signal superimposed on the PSD of $\boldsymbol{v_n}$. The large peak resulting from the pixel clock has disappeared in $\boldsymbol{v_n}$. However various peaks resulting from background sources of noise are present in both. If we were to instead simulate noise with generated AWGN, the background sources would be attenuated along with the video data. Linearly combining the signals approximates different SNRs whilst still capturing real world effects at approximately their full strength. Not all peaks are present in both PSD plots as there are many intermittent and low-duty cycle background noise sources.

cost-to-attenuation ratio [85]. However zoning is a somewhat crude method, as distance does not necessarily equate to safety. Möller [52] recommended all claimed eavesdropping distances should be independently verified. Attempts have been made to estimate the maximum distance a VDU can be eavesdropped by Du and Zhang [86]. However environmental factors such as multipath interference and diffraction caused by different construction materials, room layout, background environment, etc. can all serve to conduct, absorb, and obscure emissions in varied and often unpredictable ways. This creates a field of non-uniform signal strength around an emanation source. When observing rasters of signals intercepted from different positions around the target device in an unshielded laboratory I found the SNR sometimes ran counter to intuition and much clearer rasters were recovered when the antenna was further away from the device, even when both

antenna positions had a direct line of sight to the target. This could relate to corridor layout, which itself can cause significant disruption to wave propagation [87]. This demonstrated how physical distance makes an unsuitable candidate for the independent variable in experiments. The real metric of interest is the SNR, but as this is only approximately correlated to eavesdropping distance it is difficult to control. Instead, we can simulate a field where the signal strength attenuates uniformly as distance increases.

The power of the emanated electric field falls off proportionally to the inverse square of the distance in the far field, meaning the electric or magnetic field strength will attenuate linearly [47]. The relationship in the near field is more complicated with field strength falling off proportionally to the inverse-cube [63] or inverse-square distance, depending on antenna type and polarisation. For frequency f, the transition point between the near and far field occurs at

$$r = \frac{\mathrm{c}}{2\pi f}\,, \tag{2.66}$$

where c is the speed of light. We make two recordings: first we intercept emissions from the target device $\boldsymbol{v_0}$ near to the device while ensuring $d_0 > r$ for $f = f_\mathrm{c} - 0.5 \cdot B$. We will make the assumption that this is the maximal SNR an eavesdropper could achieve in the far field. Immediately after making the recording the target device is switched off and a second recording $\boldsymbol{v_\mathrm{n}}$ is made with the same recording parameters in the same physical location. This is intended to capture only the background noise. We can now estimate how the signal would appear at distance d from the target device as

$$v_d[n] = v_0[n] \cdot \frac{d_0}{d} + v_\mathrm{n}[n] \cdot \left(1 - \frac{d_0}{d}\right)\,. \tag{2.67}$$

This can be used to evaluate the performance of algorithms over a range of SNRs quickly and reliably, while still including real background sources of interference. Examples of background sources are shown in Figure 2.17, where the PSD of an example $\boldsymbol{v_0}$ and $\boldsymbol{v_\mathrm{n}}$ are superimposed. While the large peak of the target device's pixel clock is conspicuously absent in $\boldsymbol{v_\mathrm{n}}$, peaks from some background sources manifest in both plots and hence will not be attenuated when combined. There are still some peaks which only appear in one PSD due to intermittent transmission, or low duty cycle.

Let us compare how the simulated distance data compares with actual distance. I eavesdropped an emanating 800⊗525@25.2 MHz VDU at harmonic $h = 17$, $\boldsymbol{v_\mathrm{real}}$, from a distance of $d = 9$ m, two rooms away from the target device. I then performed the same attack in simulation for $d_0 = 0.25$ m to obtain $\boldsymbol{v}_d$. I combine the rasters into a composite (Figure 2.18) to show that the simulated distance data compares favourably with actual distance data. There is some additional noise in $\boldsymbol{v_\mathrm{real}}$ due to effects that are not included in the simulation, possibly additional levels of signal absorption or refraction in the environment—$\boldsymbol{v_\mathrm{real}}$ was eavesdropped through two walls. However the SNRs of the rasters are comparable, allowing use of simulation in experimentation to test algorithm performance over a large range of values.

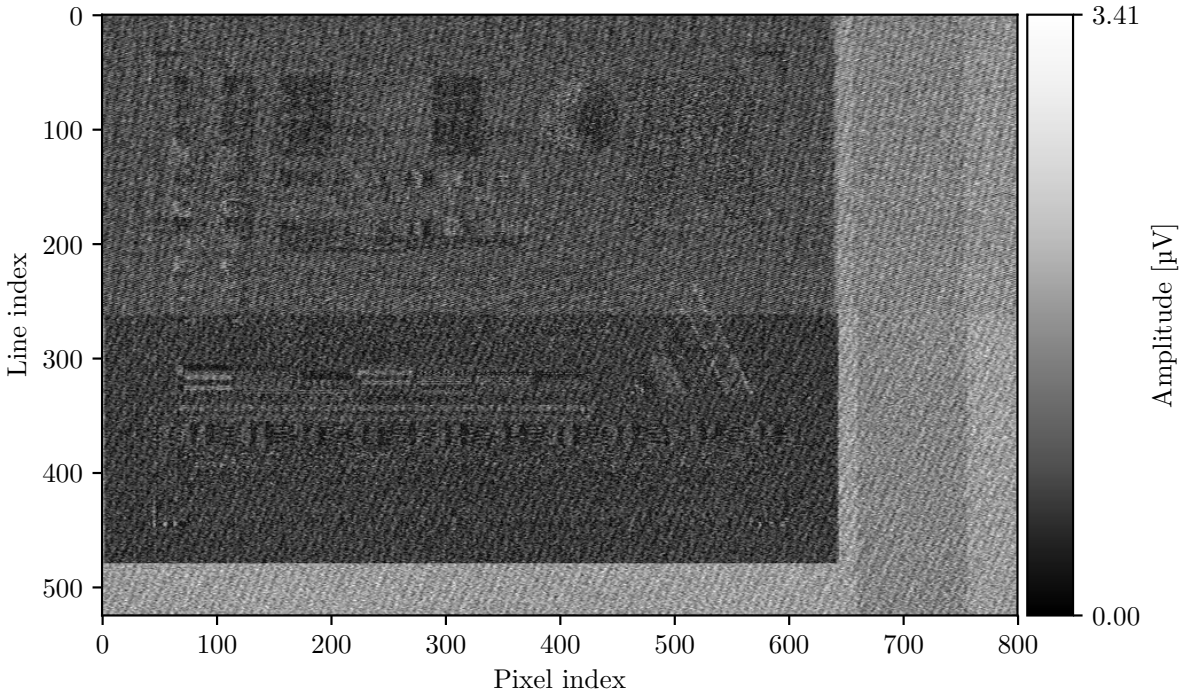


Figure 2.18. A composite raster of two frames to compare how an attack carried out at distance d compares with data simulating d from distance d_0: the top half (lines 0 to 262) is generated by an attack carried out at $d = 9$ m, while the bottom half (lines 263 to 525) is generated by simulating $d = 9$ m from $d_0 = 0.25$ m. The images have a comparable level of clarity. There is slightly more noise in the real attack compared to the simulated one, which can be attributed to additional environmental factors. In this example, two walls were between the antenna and the target device. The structure of the noise has been effectively captured, appearing in both rasters as a diagonal banding effect. The peak amplitudes of the signals were very similar, exhibiting only 0.31% difference. These rasters were averaged for 600 frames (Chapter 4).

2.9 Data sets

Throughout this work I make use of data sets of VDU emanations I built for testing and development. I collected data from a variety of contexts which I categorised for different use cases. Creating a shielded room to act as a TEMPEST laboratory is a complex procedure [1, 43, 52], and is primarily required to measure emission levels for standards compliance. This may not be as helpful when approaching emissions from the perspective of an attacker. I made recordings within normal working hours in an unshielded office environment[14]. Recordings are stored using the HDF5 file format. The raw data $\dot{\boldsymbol{v}}$ is stored as alternating I and Q values[15] such that

$$\Re\left(v[n]\right) = \dot{v}[2n]\,, \tag{2.68}$$

$$\Im\left(v[n]\right) = \dot{v}[2n+1]\,, \tag{2.69}$$

[14]William Gates Building, Cambridge, England

[15]The data format returned by the UHD.

where $\dot{v}[n] \in \{0, 1, \ldots, 2^{15} - 1\}$ is a short integer.

2.9.1 The VGA data set

I connected a Raspberry Pi 2 Model B+ revision 1.2 to a Samsung SyncMaster 710T monitor via a 3 m HDMI to DVI cable. The Raspberry Pi is a credit card-sized single-board ARM-based computer that cost 30 GBP. It is capable of outputting 1080p video data over two video outputs (HDMI and DSI). Its low cost and easy availability make it a suitable test device to allow anyone to recreate these experiments. The use of named devices seems to be a point of contention: some [10] deliberately conceal the identity of vulnerable devices whilst others [12, 26] name and shame. As the vulnerability is inherent to the video interface rather than device, I have no such qualms. It outputs displayed video mode 640×480@60 Hz (VGA) [88, p.12], with a full video mode of 800⊗525@25.2 MHz. VGA makes a suitable target video mode as it must be supported by all VGA compliant displays, and its low clock rate reduces the bandwidth of the video data, and hence the memory constraints on collecting and processing data for demonstration of various algorithms.

I captured emanations using the WA5VJB PCB log-periodic antenna in the far field at different distances where $d_0 > 0.25$ m[16], and applied 30 dB gain using a Langer PA 303 preamplifier. Recordings were made at $f_c = 428$ MHz, close to the $h = 17$ harmonic of the pixel clock ($h \cdot f_p = 428.4$ MHz). I selected this value as it is the lowest harmonic within the operational range of the WA5VJB antenna, reducing the spread of bandwidth. To down convert and digitise samples I used the USRP B200 SDR front-end for low bandwidth measurements, where the bandwidth was set to $B = 16$ MHz ($f_s = 16$ MHz), and the USRP X300 for high bandwidth measurements $f_s = B \geq 20$ MHz. I used the low-bandwidth measurements to perform fast automated testing, and the higher bandwidths to produce figures. I also made several noise recordings in the same environment with the target switched off to obtain background noise recordings for SNR simulation.

2.9.2 The Scan data set

To investigate the discovery of emanating VDUs and parameters (Chapter 3) I made 79 consecutive non-overlapping recordings at $B = 50$ MHz intervals ($f_s = 50$ MHz) for $f_c \in \{35, 85, \ldots, 3935\}$ MHz. I used the Schwarzbeck VUSLP 9111B antenna connected to the USRP X300 SDR front-end via the Langer PA 303 preamplifier. I was able to manually verify the existence and parameters of three devices in the environment whose emanations were detectable in the data set (Table 2.1).

[16]The lower frequency bound for the WA5VJB is 400 MHz meaning the maximum far field of the captured frequency range starts at approximately $r = 0.11$ m

Device	Full video mode	Distance
Samsung SyncMaster 710T	800⊗525@25.2 MHz	1 m
Iiyama Prolite B2403WS	2080⊗1235@154 MHz	2.7 m
Samsung SyncMaster 713T	1760⊗1235@130.25 MHz	6 m

Table 2.1. VDUs verified to be producing emanations in the Scan data set.

2.9.3 The Stitch data set

To investigate high-bandwidth estimates made from low-bandwidth recordings (Chapter 5), I made several recordings of a single 800⊗525@25.2 MHz device with overlap between bands. This data set contains two sets of recordings. The first is nine bands with very low bandwidth $B = 1$ MHz and a 500 kHz overlap to cover a total frequency band of 5 MHz centred over $h \cdot f_\mathrm{p} = 428.4$ MHz. The target device displayed the test card (Appendix B). The second set is three bands of $B = 160$ MHz ($f_\mathrm{s} = 200$ MHz), with an 80 MHz overlap centred over the $h = 3$ harmonic of the bit rate $h \cdot f_\mathrm{b} = 756$ MHz. The target device displayed an enlarged section of the colour boxes from the test card. Recordings were made using the USRP X300 SDR front-end with the Langer XF-R 400-1 near-field probe placed over the HDMI connector, and the Langer PA 303 preamplifier.

2.10 Summary

I outlined the practicalities of performing a basic video eavesdropping attack using SDR: utilising SDR allows for rapid prototyping and development, and opens up the possibility of using general purpose computing hardware to perform more interesting analysis of emanations, explored in the coming chapters. So far we have seen:

- An SDR front-end down converts and filters a band of the frequency spectrum permitting it to be sampled at lower rates. The IQ samples returned by the front-end are complex valued, creating a point in Cartesian space which encodes the instantaneous amplitude of the signal in its magnitude, in essence performing envelope detection. The phase is relative to the SDR front-end's LO. Positive and negative frequency offsets relative to the tuning frequency can be distinguished by the direction of phasor rotation.

- The emissions, caused by imbalances in the differential signalling wires, are bandpass filtered by the digital down-conversion stage. We can use the amplitude of IQ samples as a brightness value to approximate video signal content. This works well for areas of homogeneous brightness. However, gradient and colour information are lost. We can extend the estimation to include phase information as a hue param-

eter to help (with unrotation) better discern features with little relative amplitude difference, but with significant phase-angle difference.

- Rasterisation introduces additional quantisation noise as the SDR front-end returns a signal with a higher dynamic range than the 8 bit output raster. A transmitted video signal contains two disjoint sections: the content area, which carries the video data payload; and the non-addressable blanking interval, which has a generally higher magnitude. We can compress the dynamic range of the video signal by clipping amplitudes outside of the content area, minimising loss of visually useful information in the output raster.

- To convert a one dimensional signal into a two dimensional raster, we must know its full video mode: the width w_{T} and height h_{T} of the frame, and the pixel rate f_{p}. This allows us to resample to the pixel rate and reshape the vector into a raster. Incorrect values result in distortion of the raster proportional to the size of the error.

- Because emanated waves suffer multipath interference indoors, distance of attack can be a misleading metric. We can simulate a field of free-space propagation by linearly combining a high SNR recording with a recording of background noise, allowing us to perform testing over a wide range of simulated distances in an easily repeatable manner, independent of environmental variation.

- In addition to non-free-space wave propagation, antenna position relative to the target device can complicate comparison between rasters. The relative brightness between eavesdropped features can change dramatically, preventing one from subtracting out a template of known video content to estimate the noise level. The SNR can be estimated by using two recordings of the emanating device to compute the ratio between the signal with noise and the noise alone.

- I have created data sets representative of some typical video eavesdropping scenarios. Each has been designed for a specific context, referenced throughout this thesis.

Chapter 3

Using cepstral features for viable target device detection

In a real eavesdropping attack an attacker is unlikely to know the operational parameters of vulnerable VDUs in advance; they must be extracted from the captured emanations. Preferred values for the timing parameters used in commercially available VDU interfaces are defined by various standards organisations, most notably VESA and Consumer Electronics Association (CEA), to ensure compatibility between manufacturers. We shall define our search space as a database of full video modes $\boldsymbol{Q}$ compiled from the standards available. In this chapter I used the VESA and CEA-861-C standards, which gave a search space of $|\boldsymbol{Q}| = 138$, to search for devices in the Scan data set. The database should be recompiled regularly to incorporate the latest video modes to maximise the effectiveness of searches.

3.1 Candidate harmonic identification

We will assume no prior knowledge about which $Q \in \boldsymbol{Q}$ may be emanating in a given location. Martin and Ward [89] describe a method where emanating LOs are used to detect television-sets receiving a signal for the purpose of tax collection. Kuhn [21] states that as newer digital TV tuners often use zero-IF (LOs operating at the carrier frequency without down converting via an IF) it has become more complicated to distinguish LOs from the transmitting carrier. While not an exact cognate, VDUs lack a tuner with an LO and do not display a known signal, there is a parallel in attempting to find a VDU in a noisy environment where there may be multiple strong sources broadcasting at similar frequencies. We scan the available radio spectrum: we measure a frequency range using M consecutive non-overlapping bands with bandwidth B and sampling rate f_s ($M = 79$ for the Scan data set). The set of centre frequencies is defined with respect to the lowest

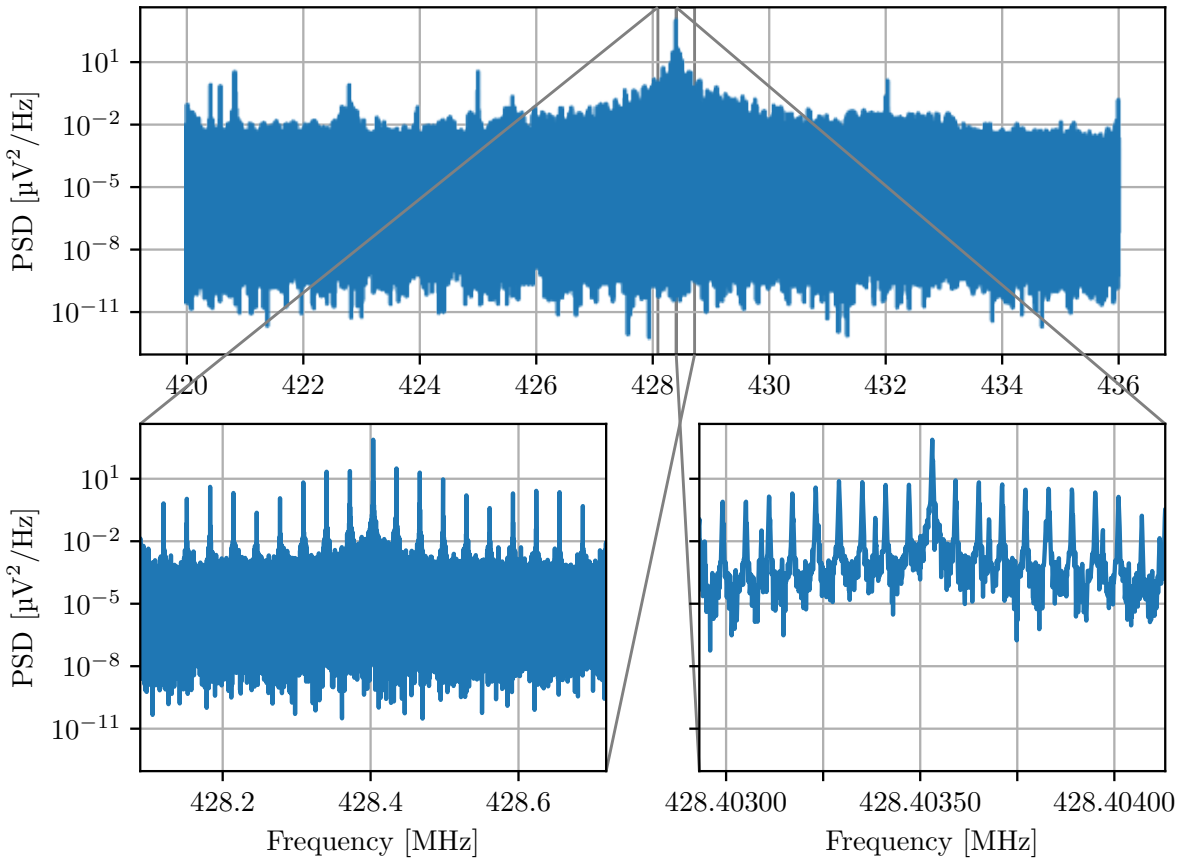


Figure 3.1. The estimate of the PSD of a spectrum containing an emanated 800⊗525@25.2 MHz video signal (top). The $h = 17$ harmonic forms a peak around which other features of the video signal can be seen: peaks spaced at the line rate $f_\mathrm{h} = 31.5$ kHz (bottom left), and smaller peaks at the frame rate $f_\mathrm{v} = 60.0$ Hz (bottom right). We can also see peaks from a background source spaced approximately every 150 Hz, uncorrelated with the video signal. A discrepancy between the expected peak location $f_\mathrm{p} \cdot h = 428.4$ MHz, where we would expect the peak to appear from the specifications [88, p.12], and the actual peak location of 428.40353 MHz, can be seen in the lower right plot due to a $\epsilon = 8$ ppm difference between the transmitting and receiving LOs, as well as clock drift.

centre frequency ${f_\mathrm{c}}^\mathrm{L}$

$$\boldsymbol{f_\mathbf{c}} = \{{f_\mathrm{c}}^\mathrm{L} + k \cdot B : k \in \mathbb{Z}_M\} \tag{3.1}$$

to cover as much of the frequency spectrum as permitted by the hardware. In doing so we maximise the chance of capturing multiple harmonics of available pixel clocks. Being periodic, the clock signal creates the strongest emanating component. We identify each recording $\boldsymbol{v}_k \in \mathbb{C}^\mathbb{Z}$ with the index k used to define its centre frequency. We compute the PSD $\left|\overline{\boldsymbol{V}_k}\right|^2$ for each $f_\mathrm{c} \in \boldsymbol{f_\mathbf{c}}$. We estimate the PSD instead of using a single Fourier transform as the signal will contain non-stationary components. I used Welch's method [90] to estimate the PSD by averaging the Fourier transforms of consecutive windows of length $N \geq \frac{f_\mathrm{s}}{1\ \mathrm{Hz}}$. Welch's method is an improvement on Bartlett's method, which was

used by Peterson *et al.* [91] to reveal hidden spectral peaks. It applies a Hanning window to attenuate spectral leakage and introduces an overlap between windows, in our case 50%, to increase the number of Fourier transforms used in the average, reducing the variance of the noise (§4.5). Preferably, use $\log_2 N \in \mathbb{N}$ for efficient computation of the fast Fourier transform (FFT). Figure 3.1 plots the PSD of a single band containing a strong video signal. We can see a number of peaks which are clearly distinct from the background noise as a result of various signal sources. The largest central peak is harmonic $h = 17$ from the clock signal of a nearby 800⊗525@25.2 MHz video signal. Its magnitude is larger than other sources due to the device's proximity to the antenna ($d_0 = 0.25$ m). Looking more closely around the clock peak, other characteristics of the video signal emerge. Initially, smaller peaks can be observed spaced at regular intervals of f_h; and between those smaller peaks spaced at intervals of f_v. This corresponds with what is expected from Figure 2.1.

For each $\left|\overline{\boldsymbol{V}_k}\right|^2$ we search for the peaks with the strongest carrier-to-noise ratio (CNR) values to generate a set of candidate harmonic peaks $\hat{\boldsymbol{c}}_k$. I define the CNR as the ratio in decibels between the magnitude at frequency f relative to the background noise level:

$$\varsigma(f) = 10\ \mathrm{dB} \cdot \log_{10}\left(\frac{\left|\overline{\boldsymbol{V}_k}\right|^2\left[\left\lfloor\left(\frac{(f-f_\mathrm{c})N}{f_\mathrm{s}} \bmod f_\mathrm{s}\right)+0.5\right\rfloor\right]}{\mathrm{median}\left|\overline{\boldsymbol{V}_k}\right|^2}\right). \tag{3.2}$$

We approximate the background noise level using the median magnitude as large outliers (peaks) would increase the mean, making it a non-representative estimate of the noise level. Peaks are identified as frequencies which satisfy the constraint of a higher CNR than the surrounding f_min frequencies:

$$\hat{\boldsymbol{c}}_k = \left\{ f : f = \underset{j\in[f-0.5\cdot f_\mathrm{min},\, f+0.5\cdot f_\mathrm{min})}{\arg\max} \varsigma(j) \right\}. \tag{3.3}$$

I empirically found $f_\mathrm{min} = 8$ MHz to be an appropriate search window. The discovered peaks from all the bands are combined into a single comprehensive set of candidate pixel-clock peaks

$$\hat{\boldsymbol{c}} = \bigcup_{k\in\mathbb{Z}_M} \hat{\boldsymbol{c}}_k\,. \tag{3.4}$$

This yielded $|\hat{\boldsymbol{c}}| = 407$ candidate peaks for the Scan data set.

3.2 Manual search

I will first demonstrate the complexity of searching for video signals manually within $\hat{\boldsymbol{c}}$, where a trained human operator appraises the results. Similar approaches have been presented by Watanabe *et al.* [60], and also by Vuagnoux and Pasini [3] who used the short-time Fourier transform to produce a spectrogram to visually inspect keyboard emanations.

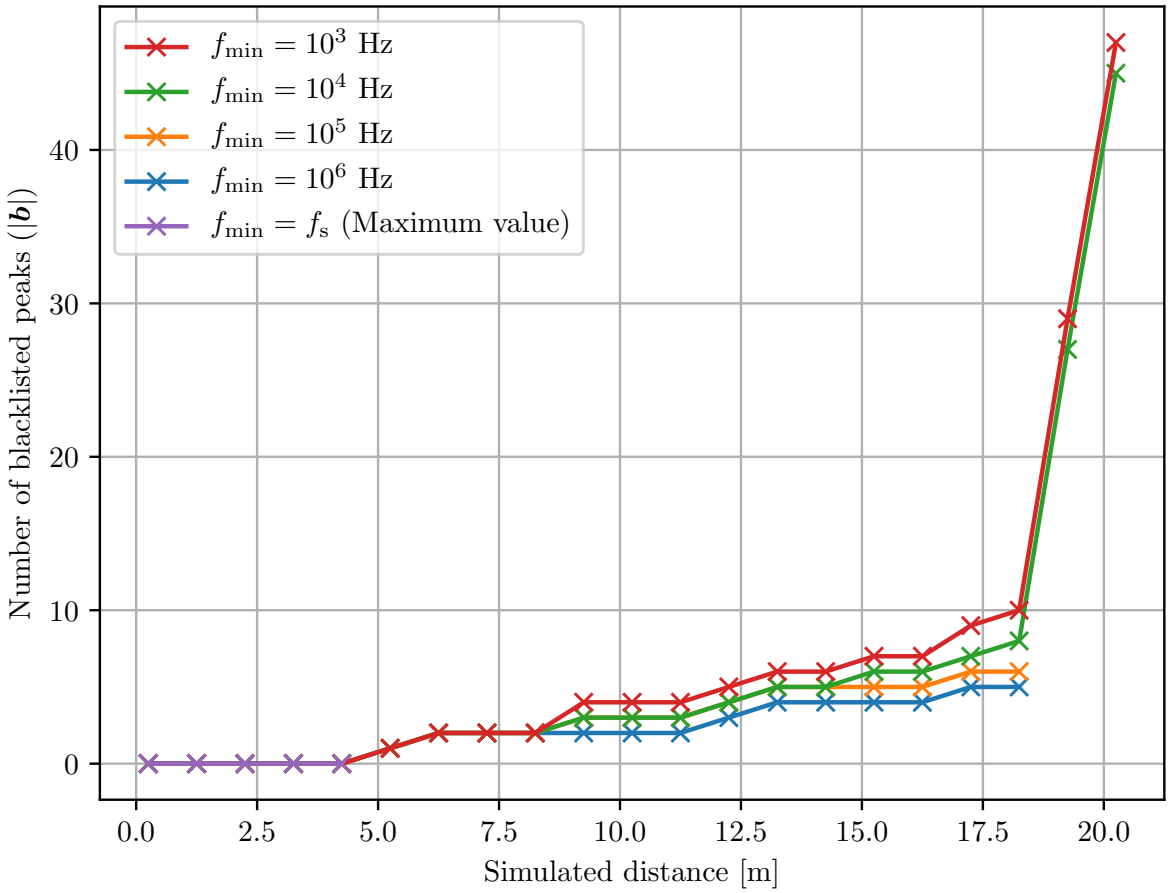


Figure 3.2. When the CNR is high, selecting the largest value in the PSD is sufficient to locate the pixel clock peak. However as noise increases, more peaks must be blacklisted and the search window constraints must be relaxed to return more candidates. This creates the roughly linear increase in the number of peaks which must be searched and blacklisted at lower SNRs. This makes the search process prohibitively time consuming. This measure does not take into account whether the operator can see any video features in the raster. This is more difficult to measure as other factors, such as operator aptitude and fatigue, could cause low SNR rasters to be missed, but will add an additional a layer of complexity to the search.

An eavesdropper would need to iterate each $\hat{c} \in \hat{\boldsymbol{c}}$ using some heuristic such as peak CNR

$$\hat{c}'_n = \underset{\hat{c} \in \hat{\boldsymbol{c}} \setminus \boldsymbol{b}_n}{\arg\max}\, \varsigma(\hat{c})\,, \tag{3.5}$$

and attempt to raster a video image for all $Q \in \boldsymbol{Q}$. If no video signal can be observed for any $\boldsymbol{Q}$ the peak is added to a set of blacklisted peaks

$$\boldsymbol{b}_{n+1} = \boldsymbol{b}_n \cup \{\hat{c}'\}\,, \tag{3.6}$$

where $\boldsymbol{b}_0 = \{\}$, and the process repeated until a valid peak is found. The worst case search complexity is $O(|\hat{\boldsymbol{c}}| \cdot |\boldsymbol{Q}|)$. Figure 3.2 plots the number of peaks one would need to blacklist when searching by CNR strength over a range of simulated distances within a single band.

The target is $h = 17$ of 800⊗525@25.2 MHz within a bandwidth of $B = 16$ MHz (VGA data set). To automate the test, peaks within 15 Hz of the known value of f_p were accepted. The pixel clock has the highest CNR in the PSD until $d = 4.25$ m, at which point background noise is incorrectly identified instead. I varied f_min to adjust the size of $\hat{\boldsymbol{c}}$; narrowing the peak search window returns more peaks, increasing the chance of the correct peak being placed in $\hat{\boldsymbol{c}}$, but increases the magnitude of the search space for the eavesdropper. The number of peaks to be blacklisted increases linearly at around 15 m, quickly becoming intractable for a human operator to render and inspect each peak individually, especially for a high $|\boldsymbol{Q}|$. In addition, this process will only work when a raster has a sufficiently high SNR to produce a visible frame. While additional processing to increase the SNR can be performed (Chapter 4), this adds another layer of complexity demonstrating the labour-intensive nature of a manual peak search, which is often unreliable. Due to these difficulties, it is clear that automation of the search process quickly becomes necessary.

3.3 Harmonic discriminant analysis

Having scanned a wide range of the frequency spectrum we will almost certainly have observed multiple harmonics of the present pixel clocks. Together they can give some indication of which Q are transmitting in the spectrum by fitting them to our model of harmonic decomposition. This is a similar approach to *nonlinear junction detectors*, which perform general electronic device discovery by illuminating an area of a building with RF and listening for returned odd harmonics, providing a range of a few metres [14, p.528]. Unlike nonlinear junction detectors however, our approach passively listens for emanated harmonics allowing us to use our knowledge of existing full video modes to restrict our results to VDU. For some f_p we would expect to observe a peak every $(2\mathbb{N} - 1) \cdot f_\mathrm{p}$. In practice additional even harmonics may be available if the clock signal waveform exhibits asymmetry between the rising and falling edges (§4.1). However the even harmonics tend to be of significantly lower amplitude than those at odd multiples of the base frequency, so we will restrict our model to odd integers only.

In a scenario without error and perfect measurement, harmonics of the same pixel clock could be identified by finding the greatest common divisor (GCD), which would be some integer multiple of f_p. In reality we cannot estimate peak location without some error: $\frac{Nf_\mathrm{p}}{f_\mathrm{s}} \notin \mathbb{N}$ or thermal changes in the LOs occurring in the interval between recordings may all either preclude computation of the GCD, or break the relationship entirely. Instead, we estimate the presence of some Q in the spectrum by comparing $\hat{\boldsymbol{c}}$ with our model of the Fourier expansion of the clock signal in Q. For each $Q \in \boldsymbol{Q}$, where $Q = w_\mathrm{T} \otimes h_\mathrm{T} @ f_\mathrm{p}$, we select the subset

$$\boldsymbol{c}_Q = \left\{ \arg\min_{c \in \hat{\boldsymbol{c}}} |h \cdot f_\mathrm{p} - c| \ : \ h \in \{1, 3, \ldots, H_Q\} \right\}, \tag{3.7}$$

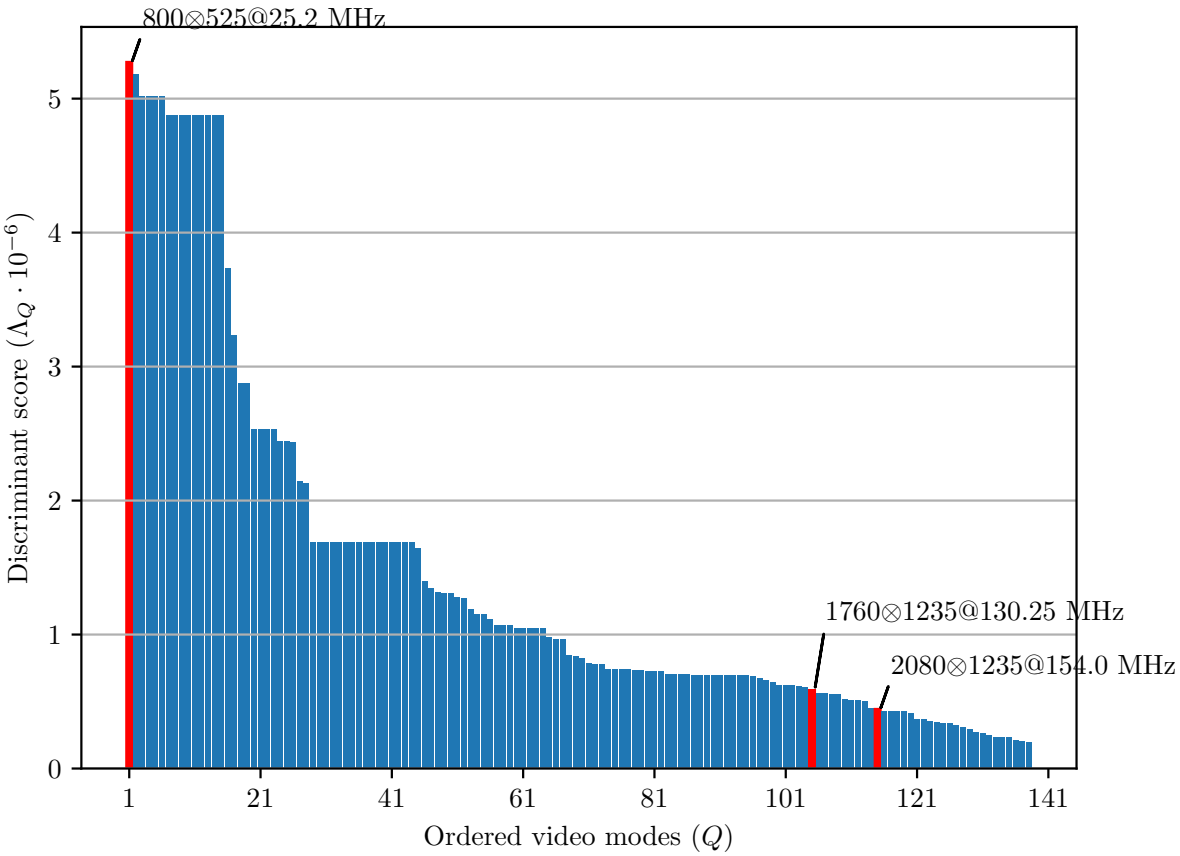


Figure 3.3. The discriminant scores assigned to each $Q \in \boldsymbol{Q}$, ordered by magnitude, in the Scan data set. The VDUs I was able to physically verify the existence of are highlighted in red. The VDU closest to the antenna, 800⊗525@25.2 MHz, is the winner and as such will be given precedence when assigning peaks. The other VDUs I verified have rather low scores at this stage. This can be due to other VDU emanating in the unshielded laboratory and background sources coinciding with expected peak locations for another Q. This motivates a second stage to eliminate incorrectly assigned peaks, allowing them to be reconsidered in later iterations.

which best matches the expected harmonics of f_p of Q, where

$$H_Q = \left\lceil 0.5 \cdot \left\lfloor \frac{\max \boldsymbol{f_c} + 0.5B}{f_\mathrm{p}} \right\rfloor \right\rceil \tag{3.8}$$

is the maximum number of odd harmonics f_p could have across the measured spectrum (assuming $f_\mathrm{p} \geq \min \boldsymbol{f_c} - 0.5B$). As f_p, and by extension f_v and f_h, are defined within some tolerance $|\epsilon| \leq 0.5\%$ [92, p.19], we assume the actual frequency of the pixel-clock harmonic to be normally distributed

$$c \sim \mathcal{N}\left(h_c \cdot f_\mathrm{p},\, (\epsilon \cdot h_c \cdot f_\mathrm{p})^2\right) , \tag{3.9}$$

where $\epsilon = 0.005$, and

$$h_c = 2 \cdot \left\lfloor \frac{c}{2 \cdot f_\mathrm{p}} \right\rfloor + 1 \tag{3.10}$$

is the odd harmonic index for c. We can then model the probability that $c \in \boldsymbol{c}_Q$ is harmonic h_c of f_p based on its distance from the expected location $h_c \cdot f_\mathrm{p}$ using the probability density function (pdf) of the normal distribution

$$p(c\,;\, f_\mathrm{p}, h_c) = \frac{1}{\sqrt{2\pi\mathbb{V}\left[c\right]}} \mathrm{e}^{-\frac{(\mathbb{E}[c]-c)^2}{2\mathbb{V}[c]}}\,. \tag{3.11}$$

Averaging over all $c \in \boldsymbol{c}_Q$, we obtain the discriminant score for Q:

$$\Lambda_Q = \frac{1}{|\boldsymbol{c}_Q|} \cdot \sum_{c \in \boldsymbol{c}_Q} p(c\,;\, f_\mathrm{p}, h_c)\,. \tag{3.12}$$

Plotting Λ_Q for all $Q \in \boldsymbol{Q}$ in Figure 3.3, we see Q = 800⊗525@25.2 MHz, which was emanating nearest to the antenna (Table 2.1), demonstrates the highest discriminant score. Other devices we verified to be in the vicinity have much lower scores due to peaks coinciding with the models of other Q, be this due to other unknown devices or background noise. Thus, peak proximity to the expected harmonic locations alone is not a reliable heuristic to distinguish harmonics from misidentified background noise; rather it identifies probable peak sets which can be further inspected for video signal features. We select the Q with the highest discriminant score

$$\hat{Q} = \operatorname*{arg\,max}_{Q \in \boldsymbol{Q}} \Lambda_Q \tag{3.13}$$

as the most likely device to be emanating in the spectrum, and discard other candidate harmonic sets as they are not guaranteed to be disjoint from $\boldsymbol{c}_{\hat{Q}}$. Considering $\hat{Q}$ first gives priority to it, reducing the chance harmonics will be misattributed.

3.4 Identifying periodicity features

We saw in Figure 3.1 how a pixel clock harmonic in the PSD is characterised by surrounding peaks spaced at intervals of f_v and f_h. We can guarantee the presence of these frequencies due to the blanking interval. During the blanking interval, the increased bit transitions result in a distinct amplitude and phase in the eavesdropped samples. Although the blanking interval is not shown on the target device, it presents a constant signal feature that can help discern a video signal from background sources for an eavesdropper. I demonstrate this property by plotting a typical recording of an emanated TMDS signal in Figure 3.4. The blanking intervals are clearly visible as a temporary increase in amplitude whilst the content displays consistently lower amplitude. The phase has a demonstrably lower variance (after unrotation, see §4.2) during blanking as changes between control characters are less common. We use these periodicities as features to differentiate actual video signals from noise. Let us consider some harmonic and the recording $\boldsymbol{v}_k$ where it was observed.

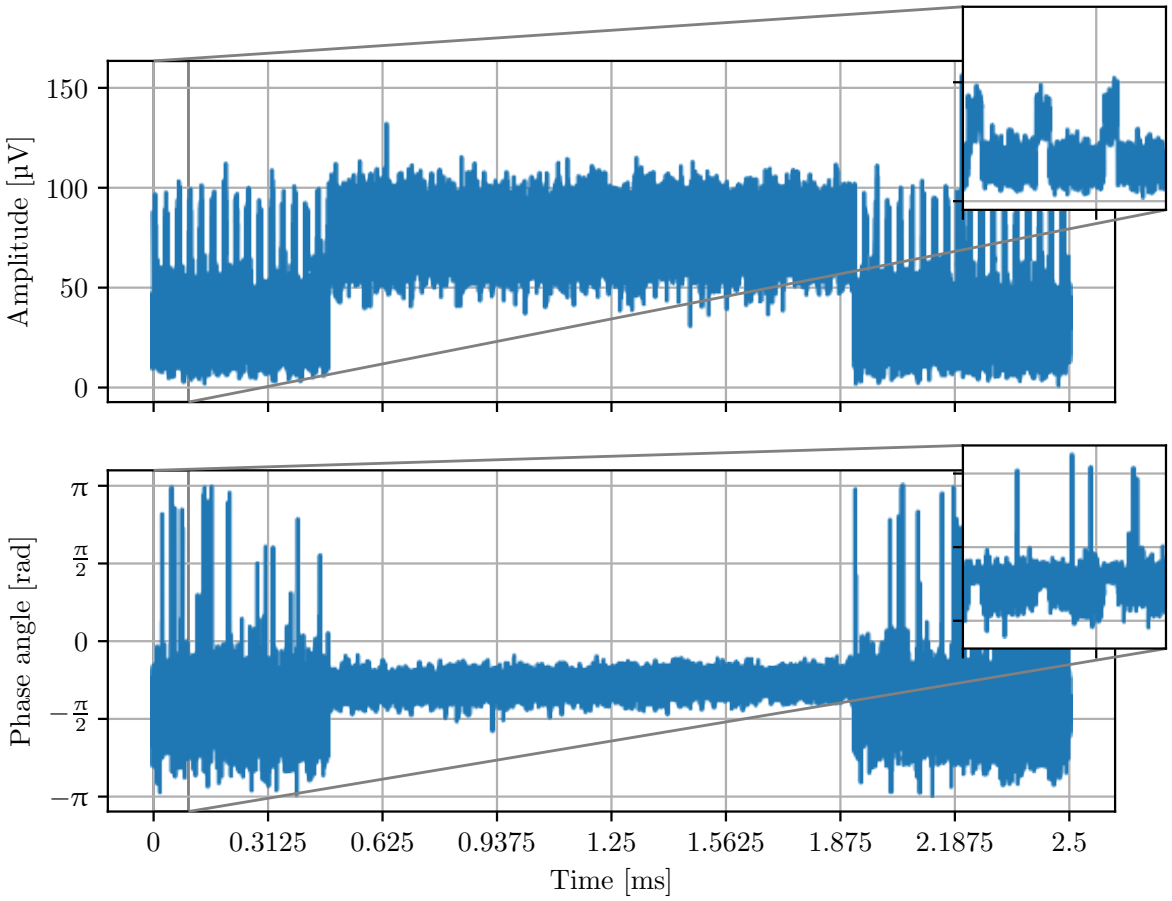


Figure 3.4. The amplitude of an eavesdropped TMDS video signal in this example increases during the vertical and horizontal (inset) blanking intervals. This is due to the higher number of bit transitions that TMDS uses in the blanking interval for synchronisation, in contrast to its objective of minimising transitions when transmitting pixel content. In the phase plot, there is a stabilisation in the (unrotated) phase angle during the blanking interval, as there are fewer character changes.

3.4.1 Autocorrelation

Perhaps the most common method of periodicity detection is the autocorrelation of a signal; its use in extracting video eavesdropping parameters for a known device is well established [64, 65, 74, 93, 94]. It is defined as the sliding dot product—the cross-correlation—of a signal with itself delayed by τ:

$$R\{\boldsymbol{v}_k\}[\tau] = \frac{1}{N-\tau} \sum_{n=0}^{N-1} v_k[n] \cdot v_k[n+\tau]^* \,. \tag{3.14}$$

Dividing by the level of overlap corrects magnitude bias favouring lower τ introduced by the fewer samples overlapping at higher delays. The resulting dot product indicates the level of self-similarity $\boldsymbol{v}_k$ has after delay τ. Thus, a peak in the autocorrelation at τ

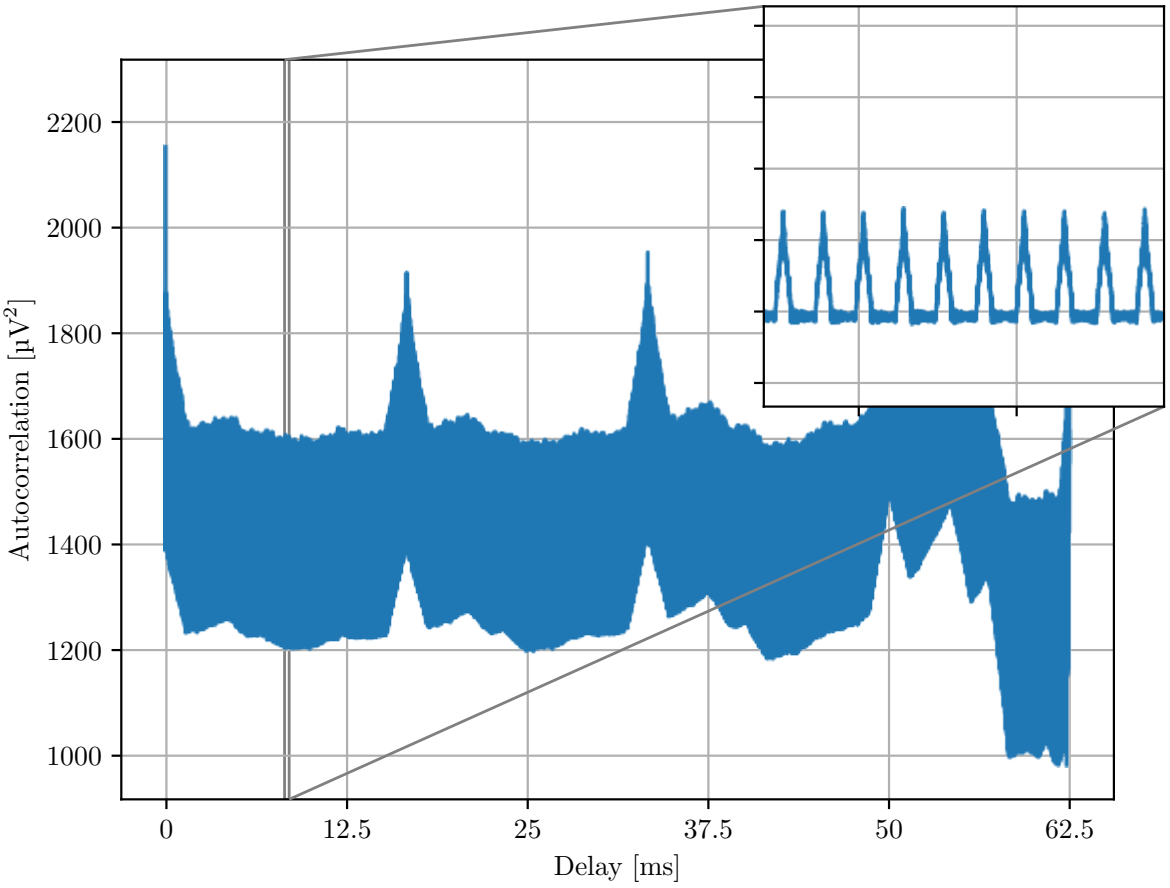


Figure 3.5. Autocorrelation plot of an emanated 800⊗525@25.2 MHz video signal at $h = 17$. The periodic frames create peaks every $f_\mathrm{s} \cdot {f_\mathrm{v}}^{-1}$ samples. The inset shows a similar effect every $f_\mathrm{s} \cdot {f_\mathrm{h}}^{-1}$ samples as a result of the line rate. The autocorrelation shows the highest level of self-similarity at $\tau = 0$ as it is correlating the signal with an exact copy of itself (resulting in the power) and so can be ignored. The autocorrelation is normalised by dividing by the level of overlap, otherwise the peaks would become difficult to discern at higher delays due to fewer overlapping values contributing to the dot product.

indicates the presence of a periodic component with a period that divides τ, and frequency

$$f_\tau = \frac{f_\mathrm{s}}{\tau}\,. \tag{3.15}$$

For every period present in $\boldsymbol{v}_k$ a peak in $R\{\boldsymbol{v}_k\}$ will occur every τ. Figure 3.5 plots the autocorrelation of an emanated video signal. Regular peaks result from the frame rate and line rate. We can ignore the first (and largest) peak at $\tau = 0$, which simply tells us the signal appears identical to itself. A peak search of the autocorrelation plot of a video signal was used by Elibol *et al.* [64] and Lee *et al.* [22] to estimate the line and frame periodicities; Sun *et al.* [95] used the Fourier transform of the autocorrelation—the periodogram. However, the rasters produced still demonstrated a slanting effect from imprecise parameters. Some more esoteric approaches used image processing to correct this, such as Hayashi *et al.* [4] who used the Hough transform to identify the characteristic

slant and apply a correcting factor. However this requires a raster with a high SNR, and only provides a rough approximation of the error.

Computation of the autocorrelation can be simplified using the Wiener-Khinchin theorem, which allows us to express the autocorrelation in terms of the periodogram [96]. Convolution is equivalent to multiplication in the frequency domain. The dot product of two signals—the correlation—is closely related to convolution, but without the time reversal of one of the inputs. Taking the conjugate of one of the frequency domain signals is the equivalent of reversing the time domain input—undoing the implicit reversal of the input when performing convolution in the frequency domain, resulting in correlation. In the frequency domain this is equivalent to computing the periodogram, which is a less sophisticated alternative to Welch's method for estimating the PSD:

$$\left|\overline{\boldsymbol{V}_k}\right|^2 \approx \mathcal{F}\{\boldsymbol{v}_k\} \cdot \mathcal{F}\{\boldsymbol{v}_k\}^* \,. \tag{3.16}$$

Because the discrete Fourier operations treat the inputs as infinitely long periodic functions, the resulting correlation is circular and will have wrap-around effects at the boundary. If we first zero pad $\boldsymbol{v}_k$ (of length N) to at least twice its length:

$$w_k[n] = \begin{cases} v_k[n] & n < N \\ 0 & N \leq n < 2N \,, \end{cases} \tag{3.17}$$

the correlation becomes non-circular. Now, taking the inverse Fourier transform of the periodogram results in the autocorrelation

$$R\{\boldsymbol{w}_k\}[\tau] = \frac{1}{N-\tau} \cdot \mathcal{F}^{-1}\{\mathcal{F}\{\boldsymbol{w}_k\} \cdot \mathcal{F}\{\boldsymbol{w}_k\}^*\}[\tau] \,. \tag{3.18}$$

We can now see more clearly the inverse relationship of the distance between the peaks in the PSD in Figure 3.1, and of the peaks in the autocorrelation plot. Again, this provides a speed-up in computation from $O(N^2)$ up to $O(N \log N)$ when using the FFT and $\log_2 N \in \mathbb{N}$.

3.4.2 Cepstral analysis

The Wiener-Khinchin formulation of the autocorrelation is closely related to cepstrum[1] analysis. The latter was originally developed to detect echoes in seismic signals resulting from nuclear detonation [98], and has found further use in vocal analysis and mechanical fault detection [99]. An echo is a time-delayed, amplitude-scaled version of a feature. In

[1]The word *cepstrum* is an anagram of *spectrum*. Bogert *et al.* [97] defined a host of words for log domain features and operations which all played on the theme of reversing the first syllable of the non-log cognate. While most of these words have faded from common use, cepstrum and quefrency (an anagram of *frequency*) remain.

our case we can see an 'echo' as a periodic feature, i.e. frames and lines, without any gain scaling. By way of example, let us define a periodic stationary signal as a convolution

$$\boldsymbol{v} = \boldsymbol{x} * \text{Ш}_\tau \tag{3.19}$$

of a single period of some feature $\boldsymbol{x}$ of duration τ with a Dirac comb, which creates repetitions of the feature in the time domain:

$$\begin{aligned}\boldsymbol{v} &= (\boldsymbol{x} * \boldsymbol{\delta}) + (\boldsymbol{x} * \boldsymbol{\delta}_\tau) + (\boldsymbol{x} * \boldsymbol{\delta}_{2\tau}) + \dots \\ &= \boldsymbol{x}_0 + \boldsymbol{x}_1 + \boldsymbol{x}_2 + \dots .\end{aligned} \tag{3.20}$$

When delayed in the time domain by $i \in \mathbb{N}$ periods, the Fourier component f of $\boldsymbol{x}_i$ is equivalent to component f of $\boldsymbol{x}$ phase shifted by

$$S_i[f] = \mathrm{e}^{-2\pi \mathrm{j} f i \tau}\,, \tag{3.21}$$

allowing us to express $\boldsymbol{X}_i$ in terms of the non-delayed component

$$\mathcal{F}\{\boldsymbol{x}_i\}[f] = \mathcal{F}\{\boldsymbol{x}\}[f] \cdot S_i[f]\,. \tag{3.22}$$

Thus the frequency content of $\boldsymbol{v}$ can be expressed as

$$\begin{aligned}\boldsymbol{V} &= \boldsymbol{X} + \boldsymbol{X} \cdot \boldsymbol{S}_1 + \boldsymbol{X} \cdot \boldsymbol{S}_2 + \dots \\ &= \boldsymbol{X} \cdot \left(1 + \sum_i \boldsymbol{S}_i\right)\,.\end{aligned} \tag{3.23}$$

Like the autocorrelation, we obtain $\boldsymbol{V}$ by computing the periodogram of $\boldsymbol{v}$. However we then apply a logarithm operation to convert to the cepstral domain. This has important implications for how we interact with the data: by virtue of logarithmic laws, multiplications become additions, and divisions become subtractions:

$$\log \boldsymbol{V} = \log \boldsymbol{X} + \log\left(1 + \sum_i \boldsymbol{S}_i\right)\,. \tag{3.24}$$

Now, rather than acting to apply a phase rotation, the phasors of the delay components add a log-sum component to the cepstral domain. Applying the inverse Fourier transform[2] returns a time-domain signal of the log-scaled periodogram, known as the *quefrency* domain. We take the absolute square to arrive at the definition of the power cepstrum:

$$\mathcal{C}\{\boldsymbol{w}_k\} = |\mathcal{F}^{-1}\{\log(|\mathcal{F}\{\boldsymbol{w}_k\}|^2)\}|^2\,. \tag{3.25}$$

The delay components cause ripples in the quefrency domain which indicate the points of self-similarity[3]. To discuss the magnitude of the ripples in terms of a real-world unit I

[2]Some definitions apply the forward Fourier transform instead [98] which results in an amplitude scaled result.

[3]Dubbed *rahmonics* in cepstrum jargon.

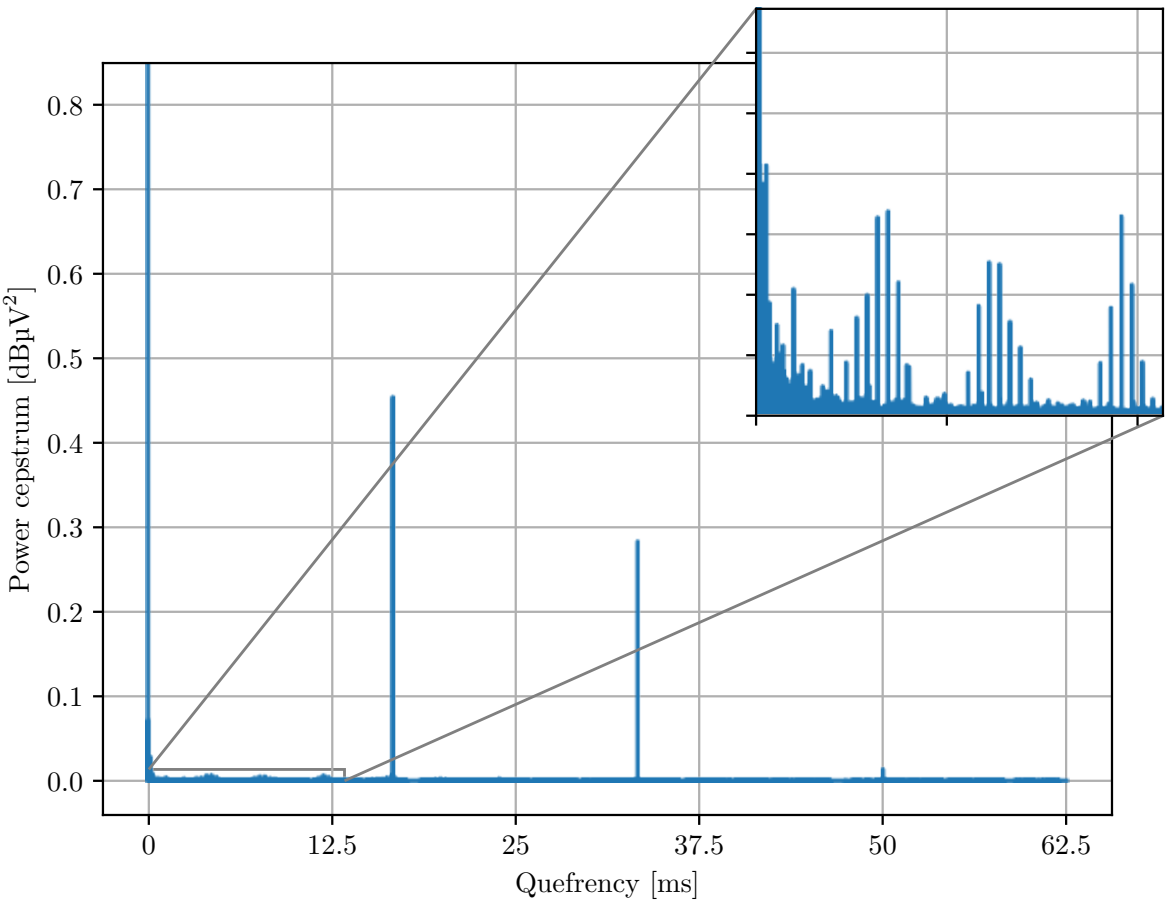


Figure 3.6. Cepstrum plot of an emanated 800⊗525@25.2 MHz video signal at $h = 17$. The peaks caused by the frame rate are clearly defined relative to other periodicities and background noise, much more so than in the autocorrelation plot. The value at $\tau = 0$ is so large compared to the other values it has been clipped in this plot to make other features observable.

make a slight adjustment to Equation 3.25. A logarithm takes a dimensionless input, thus we divide out the unit with the desired reference value—in the examples I use microvolts. By using base 10 for the logarithm and scaling the result by 10 dB the unit of the logarithm operation becomes dBμV. As we are using a field ratio we can simplify the equation by multiplying by 20 dB and removing the square operations:

$$\begin{aligned} \mathcal{C}\{\boldsymbol{w}_k\} &= \left| \mathcal{F}^{-1} \left\{ 10 \text{ dB} \cdot \log_{10} \left(\frac{|\mathcal{F}\{\boldsymbol{w}_k\}|^2}{|1 \text{ μV}|^2} \right) \right\} \right|^2 \\ &= \left| \mathcal{F}^{-1} \left\{ 20 \text{ dB} \cdot \log_{10} \left(\frac{|\mathcal{F}\{\boldsymbol{w}_k\}|}{|1 \text{ μV}|} \right) \right\} \right|^2 . \end{aligned} \tag{3.26}$$

The final square operation after converting back to the quefrency domain means the units become dBμV^2. The result is a scaled version of the original power cepstrum definition and is used from now on.

In Figure 3.6 I plot the cepstrum of the same section of signal used to produce the autocorrelation in Figure 3.5. The peaks caused by f_v are clearly observable, as are smaller

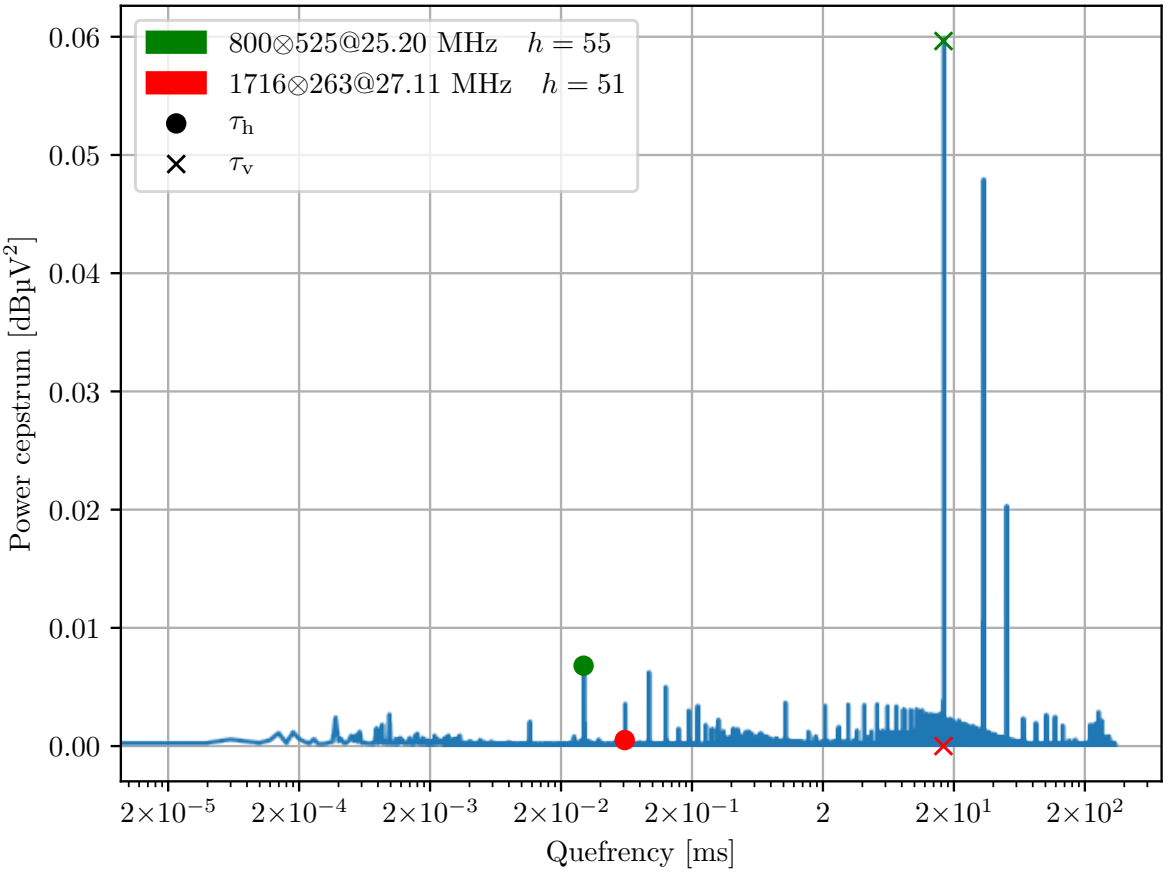


Figure 3.7. Peaks appearing at the corresponding τ_v and τ_h locations in the quefrency domain strongly indicate the presence of a video signal with parameters Q. In this cepstrum plot the two larger peaks coincide with the computed locations of τ_v and τ_h for $h_c = 55$ of 800⊗525@25.2 MHz. The video signal 1716⊗263@27.1 MHz at $h_c = 51$, which also had a candidate clock harmonic located in the band, produces quefrency locations near to, but not exactly at, the peak locations resulting in much lower values. The high contrast between ripples and the background level significantly reduces the power at the quefrencies of the incorrect peaks. Consequently $h_c = 51$ can be rejected as an invalid harmonic of 1716⊗263@27.1 MHz.

ripples caused by f_h. Compared to the autocorrelation plot the peaks are much sharper and more clearly defined against the background noise level. Like the autocorrelation, the power at some quefrency τ relates to the similarity of the signal to itself when delayed by τ. We can, therefore, use the cepstrum as a drop-in replacement for autocorrelation. (Other types of cepstrum exist, such as the complex cepstrum which does not discard phase information and instead uses a complex logarithm. However, these are intended to be used if the operation needs to be reversible, such as when filtering (liftering) in the cepstral domain to remove potential divisions by small numbers. In practice the complex cepstrum has been found to be generally difficult to use [99].)

3.4.3 Periodicity covariance

For each $c \in \boldsymbol{c}_{\hat{Q}}$ we compute the corresponding f_v and f_h using Equations 2.39 and 2.40 where

$$f_\mathrm{p}{}' = \frac{c}{h_c}\,. \tag{3.27}$$

The corresponding $f_\mathrm{v}{}'$ and $f_\mathrm{h}{}'$ components will appear at quefrency samples

$$\tau_\mathrm{h} = \frac{f_\mathrm{s}}{f_\mathrm{h}{}'}\,, \tag{3.28}$$

$$\tau_\mathrm{v} = \frac{f_\mathrm{s}}{f_\mathrm{v}{}'}\,, \tag{3.29}$$

which we round to the nearest integer to locate the bin containing the most information for the respective periodic components for c:

$$L_{\mathrm{h},c} = \mathcal{C}\{\boldsymbol{w}_k\}\left[\lfloor \tau_\mathrm{h} + 0.5 \rfloor\right]\,, \tag{3.30}$$

$$L_{\mathrm{v},c} = \mathcal{C}\{\boldsymbol{w}_k\}\left[\lfloor \tau_\mathrm{v} + 0.5 \rfloor\right]\,. \tag{3.31}$$

where

$$k = \frac{\arg\min\left\{|f_\mathrm{c} - c| - f_\mathrm{c}{}^\mathrm{L} \,:\, f_\mathrm{c} \in \boldsymbol{f_c}\right\}}{B}\,. \tag{3.32}$$

Strong components at both these quefrency locations indicate c is indeed harmonic h_c of $\hat{Q}$. Figure 3.7 plots the cepstra of $\boldsymbol{w}_{27}$ from the Scan data set ($f_\mathrm{c} = 1385$ MHz) and marks the location of τ_v and τ_h for $Q = 800{\otimes}525@25.2$ MHz at $h_c = 55$, and $Q = 1716{\otimes}263@27.1$ MHz at $h_c = 51$, both of which had candidate harmonics in $\boldsymbol{w}_{27}$, the latter incorrectly so. The computed τ_v and τ_h for the harmonic of $1716{\otimes}263@27.1$ MHz are close to, but ultimately miss, the peaks resulting in a low $L_{\mathrm{h},c}$ and $L_{\mathrm{v},c}$. For $800{\otimes}525@25.2$ MHz they exactly align resulting in high values.

We compute the quefrency strengths for all $c \in \boldsymbol{c}_{\hat{Q}}$ to obtain the sets

$$\boldsymbol{L}_\mathbf{v} = \left\{L_{\mathrm{v},c} \,:\, c \in \boldsymbol{c}_{\hat{Q}}\right\} \tag{3.33}$$

$$\boldsymbol{L}_\mathbf{h} = \left\{L_{\mathrm{h},c} \,:\, c \in \boldsymbol{c}_{\hat{Q}}\right\}\,. \tag{3.34}$$

Figure 3.8 plots $L_{\mathrm{v},c} \in \boldsymbol{L}_\mathbf{v}$ against $L_{\mathrm{h},c} \in \boldsymbol{L}_\mathbf{h}$ for each selected $\hat{Q}$; a trend emerges for VDUs known to be operating in the vicinity. The data points of misidentified peaks form a mass of uncorrelated values with a low magnitude. Conversely, candidate peaks produce a large value for both $L_{\mathrm{v},c}$ and $L_{\mathrm{h},c}$ when correctly associated with $\hat{Q}$, forming a positive correlation. We can measure this relationship with the covariance:

$$\mathrm{covar}(\boldsymbol{c}) = \frac{1}{|\boldsymbol{c}|}\sum_{c\in\boldsymbol{c}}(L_{\mathrm{v},c} - \overline{\boldsymbol{L}_\mathbf{v}})(L_{\mathrm{h},c} - \overline{\boldsymbol{L}_\mathbf{h}})\,, \tag{3.35}$$

where $\overline{\boldsymbol{L}_\mathbf{v}}$ and $\overline{\boldsymbol{L}_\mathbf{h}}$ are the quefrency set's means. The covariance gives a positive value when $\boldsymbol{L}_\mathbf{v}$ and $\boldsymbol{L}_\mathbf{h}$ are positively correlated, indicating $\boldsymbol{c}_{\hat{Q}}$ is a valid harmonic set for $\hat{Q}$.

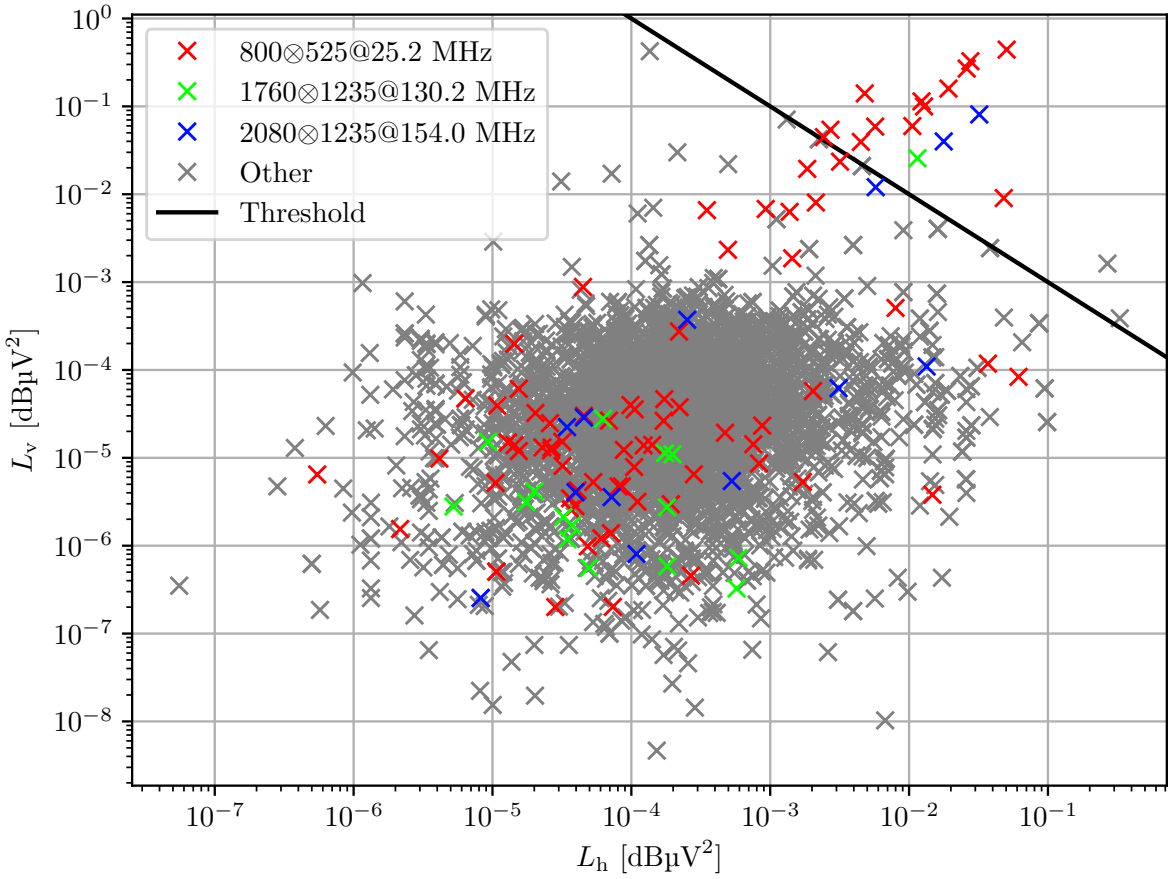


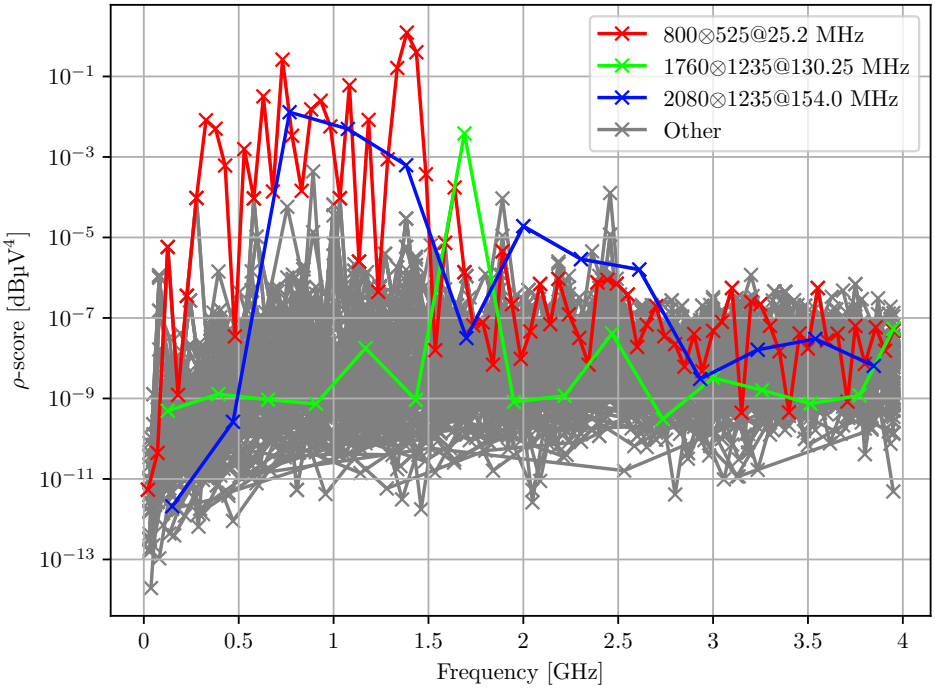
Figure 3.8. When plotting each $L_{\mathrm{v},c} \in \boldsymbol{L}_{\mathbf{v}}$ and $L_{\mathrm{h},c} \in \boldsymbol{L}_{\mathbf{h}}$ for every $\boldsymbol{c}_{\hat{Q}}$ considered by the algorithm, we see most appear uncorrelated with a low magnitude, requiring log-log axes to see. A few harmonics form a positive trend however, which indicates strong frequency components for both f_{v} and f_{h} for the associated $\hat{Q}$. A threshold separates the positively correlated values, allowing us to discard misidentified harmonics. The candidate harmonics for the Q I manually verified to exist in the spectrum are highlighted in a different colour for each Q; a significant proportion are rejected by the threshold indicating a high number of peaks are the misidentified result of noise.

To select some $\boldsymbol{d} \subseteq \boldsymbol{c}_{\hat{Q}}$, rejecting those which do not fit the profile of $\hat{Q}$, we measure the contribution of each $c \in \boldsymbol{c}_{\hat{Q}}$ to the covariance:

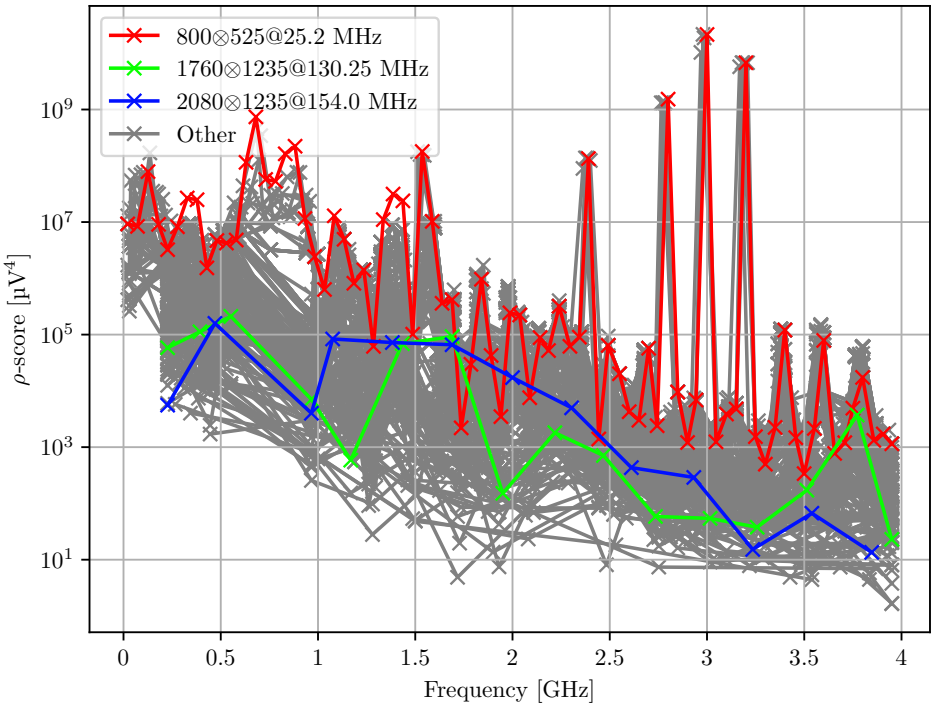
$$\rho_c = (L_{\mathrm{v},c} \cdot L_{\mathrm{h},c}) h_c^2 \,, \tag{3.36}$$

obtaining a measure for the strength of the relationship between f_{v} and f_{h}, normalised by the expected harmonic amplitude scaling[4] to make ρ-scores of different harmonics comparable. A low ρ_c indicates c is not a harmonic of $\hat{Q}$. Figure 3.9 compares the ρ-scores of each $c \in \boldsymbol{c}_Q$ for each $Q \in \boldsymbol{Q}$. The ρ-scores generated using the cepstrum (Figure 3.9a) clearly distinguish the Q present in the spectrum from the background noise. The ρ-scores reduce in magnitude at higher harmonics as the signal power is attenuated making

[4]The Fourier expansion of a square wave shows the amplitude of harmonic h is scaled by $A(h) = \frac{4}{h\pi}$.



(a) Cepstrum



(b) Autocorrelation

Figure 3.9. Comparison between the ρ-scores assigned to the candidate harmonic sets $\boldsymbol{c}_Q$ for each Q. The cepstrum produces a clear distinction between valid harmonics and noise, whereas the autocorrelation provides little distinction between the two with a significant level of noise. Trends in the autocorrelation tend to affect all peaks in the vicinity, demonstrating its susceptibility to background noise.

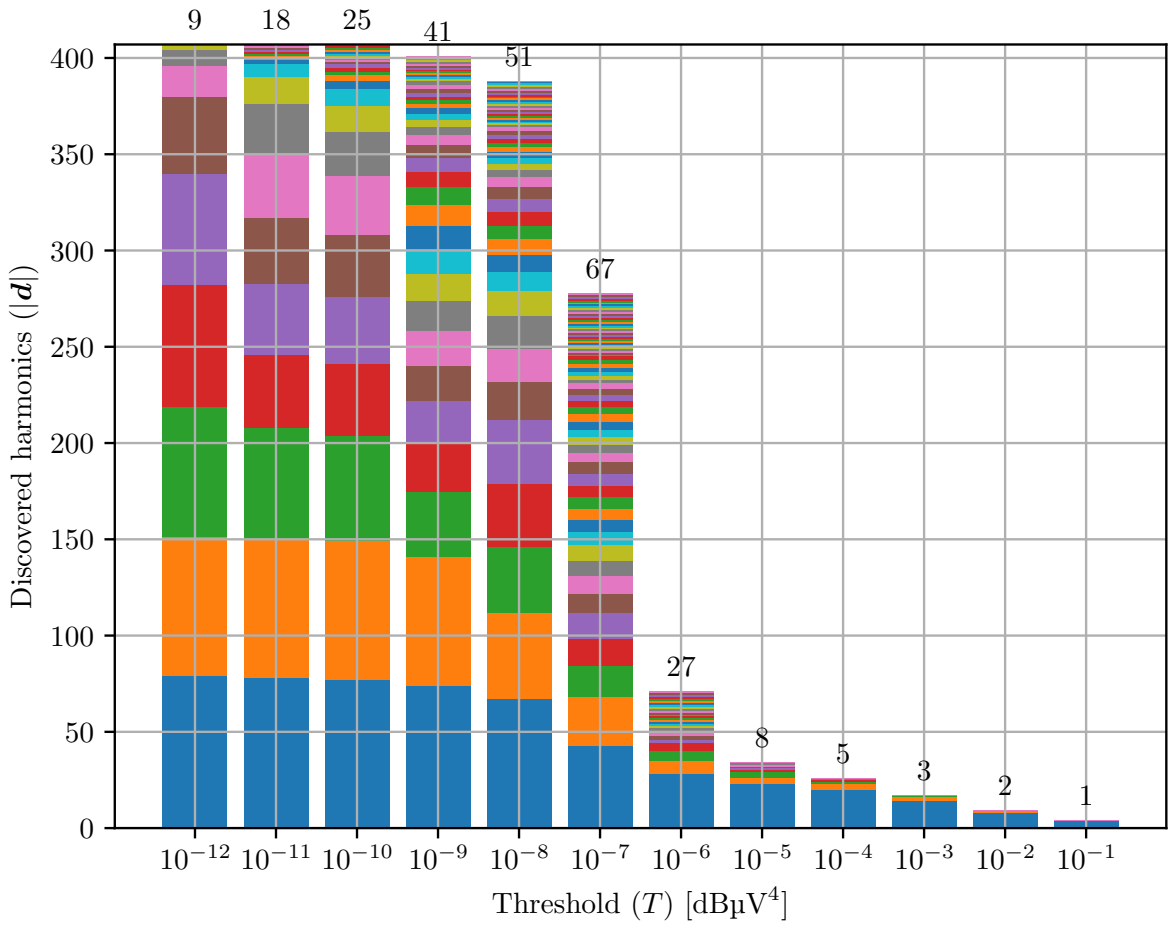


Figure 3.10. Lowering the threshold to be more accepting of candidate harmonics slowly increases the number of discovered devices, displayed above each bar, as does the number of harmonics recovered per Q ($|\boldsymbol{d}|$). However, there is a sudden increase in the number of devices discovered once the threshold becomes so low it hits the noise floor and many incorrect devices are identified. The number of devices discovered reduces as the threshold continues to be lowered. This is because a less discerning threshold results in many harmonics being assigned to a Q early in the algorithm, leading to premature termination of the procedure.

them indistinguishable from noise. This results in stronger background sources being incorrectly identified as harmonics which must then be rejected. Conversely, the ρ-scores generated using the autocorrelation (Figure 3.9b) give a much noisier result, with little to differentiate between video and non-video signals. This is due to the much noisier autocorrelation plot, demonstrating its unsuitability for this approach.

3.4.4 False positive rejection

We reject erroneous $c \in \boldsymbol{c}_{\hat{Q}}$ by a process of thresholding, generating the set of viable harmonics for $\hat{Q}$:

$$\boldsymbol{d}_{\hat{Q}} = \{c \in \boldsymbol{c}_{\hat{Q}} \,:\, \rho_c > T\}\,, \tag{3.37}$$

Algorithm 1 Detection of potential emanating clock harmonics

procedure DETECT($\boldsymbol{Q}$, $\hat{\boldsymbol{c}}$, $\boldsymbol{w}$, T)
  $\boldsymbol{F} \leftarrow \{\}$ ▷ Empty set of discovered devices
  while $\boldsymbol{Q} \neq \{\}$ and $\hat{\boldsymbol{c}} \neq \{\}$ **do** ▷ While modes and peaks remain
    $\hat{Q}, \boldsymbol{c}_{\hat{Q}} \leftarrow \arg\max_{Q \in \boldsymbol{Q}} \Lambda(Q, \hat{\boldsymbol{c}})$ ▷ Select mode and harmonics (Eq. 3.7 – 3.13)
    for $c \in \boldsymbol{c}_{\hat{Q}}$ **do** ▷ For each candidate harmonic
      $k \leftarrow \text{index}(c, \boldsymbol{f}_{\mathbf{c}})$ ▷ Select index of source recording (Eq. 3.32)
      $\rho_c \leftarrow \text{score}(c, \mathcal{C}\{\boldsymbol{w}_k\})$ ▷ Compute harmonic score (Eq. 3.27 – 3.36)
      if $\rho_c < T$ **then** ▷ If the score is below threshold
        $\boldsymbol{c}_{\hat{Q}} \leftarrow \boldsymbol{c}_{\hat{Q}} \setminus c$ ▷ Discard
      end if
    end for
    if $\boldsymbol{c}_{\hat{Q}} = \{\}$ **then** ▷ If no harmonics remain
      $\boldsymbol{Q} \leftarrow \boldsymbol{Q} \setminus \hat{Q}$ ▷ Discard mode
    else
      $\hat{\boldsymbol{c}} \leftarrow \hat{\boldsymbol{c}} \setminus \boldsymbol{c}_{\hat{Q}}$ ▷ Remove selected harmonics from pool
      $\boldsymbol{F} \leftarrow \boldsymbol{F} \cup \left\{\left(\hat{Q}, \boldsymbol{c}_{\hat{Q}}\right)\right\}$ ▷ Add to discovered set
    end if
  end while
  return $\boldsymbol{F}$ ▷ Return discovered devices
end procedure

where T is a threshold to the ρ-score that can be adjusted to be more stringent or relaxed, as decided by the eavesdropper. We consider $\hat{Q}$ discovered if $\boldsymbol{d}_{\hat{Q}} \neq \{\}$. Having been assigned to $\hat{Q}$, we remove $\boldsymbol{d}_{\hat{Q}}$ from $\hat{\boldsymbol{c}}$. Figure 3.10 plots the $|\boldsymbol{d}_Q|$ identified for each discovered $Q \in \boldsymbol{Q}$ as the threshold is adjusted. If no harmonics have been assigned to $\hat{Q}$, then it is removed from $\boldsymbol{Q}$. The algorithm is iterated from the computation of the discriminant scores until either $\hat{\boldsymbol{c}} = \{\}$ or $\boldsymbol{Q} = \{\}$. Algorithm 1 summarises the entire procedure.

I set the threshold above the noise floor to $T = 10^{-4}$ dBµV^4 and list the discovered Q in Table 3.1. Of these I was able to manually verify the existence of three VDUs (Table 2.1).

3.5 Harmonic selection

While the Fourier expansion of a square wave shows higher harmonics will emanate less power, they may result in a higher SNR as lower harmonics are more easily absorbed [15, 53] or affected by lower antenna factors. Also, background radiation is attenuated at higher frequencies [100], potentially resulting in stronger SNR at higher harmonics.

Video Mode	Number of Harmonics
800⊗525@25.2 MHz	20
2080⊗1235@154.0 MHz	3
858⊗525@27.0 MHz	1
864⊗625@27.0 MHz	1
1760⊗1235@130.2 MHz	1

Table 3.1. Discovered VDUs in the Scan data set when $T = 10^{-4}$ dBμV^4.

Nearby interference, such as radio stations or other VDUs, can interfere with some harmonics, but not others. Therefore selection of the harmonic for the best SNR is not immediately obvious. We select the $d \in \boldsymbol{d}_{\hat{Q}}$ with the strongest ρ-score

$$\hat{d} = \underset{d \in \boldsymbol{d}_{\hat{Q}}}{\arg\max}\, \rho_d \tag{3.38}$$

to eavesdrop, as this estimates the strongest video features. We use $\hat{d}$ to raster the verified Q in Figure 3.11. Detail is observable for 800⊗525@25.2 MHz as it is both nearer the antenna, providing a higher SNR, and has a lower f_p, so the video data is contained within a smaller bandwidth. However the 2080⊗1235@154 MHz raster still produces clearly defined blanking interval borders. For 1760⊗1235@129.6 MHz, although no video data is immediately visible in the raster, I was able to verify the existence of the VDU in a neighbouring office at a distance of approximately 6 m from the antenna through a single drywall. Further processing will be needed to lift the video signal out of the noise. This shows that whilst the signal is not visible to an operator, a requirement for manual searching, it could still be detected using this approach. As such, TEMPEST limits should take into account that having the video signal lost in noise is insufficient to hide its existence from an eavesdropper.

3.6 Summary

I have presented an approach to identify VDUs potentially vulnerable to video eavesdropping attacks by analysing the periodicity features characteristic to a video signal. The main contributions are as follows:

- Although manually searching for video signals is conceptually easy to understand and implement, it is a labour-intensive and unreliable process. One must attempt to raster a video signal for each peak in the PSD and blacklist erroneous solutions. After a threshold distance, the number of peaks one must search and discard increased approximately linearly as the distance increased. While emanating pixel

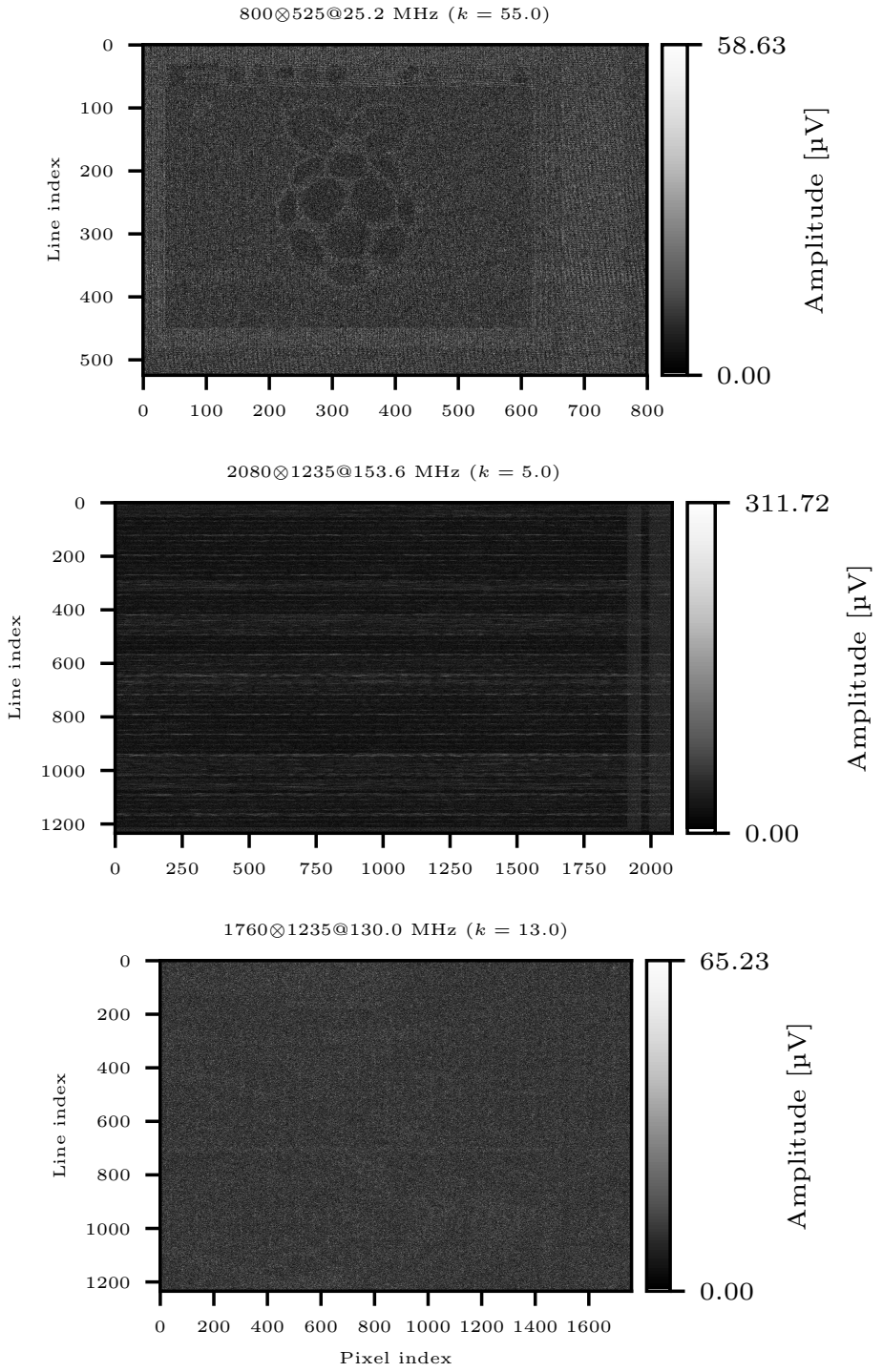


Figure 3.11. Single frame rasters of the harmonics with the strongest ρ-score ($\hat{d}$) for the detected video signals in Table 3.1. The 800⊗525@25.2 MHz raster provides the best picture quality as it has a higher SNR, and its content is contained in a narrower bandwidth. This is in contrast to 2080⊗1235@153.6 MHz where a faint outline of the blanking interval can be observed. No picture data can be seen for 1760⊗1235@129.6 MHz, located in a neighbouring room, showing that even if a VDU is not visible to an operator it can still be identified. Further processing will need to be performed to make its contents visible.

clocks will eventually be identified, given the sheer number of full video modes and frequency location combinations, and often low CNR, it is unrealistic to carry out a manual search in practice.

- I compiled the search space of full video modes from standards documentation used in commercially available VDUs. A model of square wave harmonic decomposition was initially used to find likely groupings of harmonics from the same device. Incorrect harmonics and background noise were then eliminated by identifying periodicity features characteristic of a specific full video mode. This contradicts Möller [52], who states the frame and line rates are not compromising because they do not contain video data. As the line and frame rates carry information which can help locate emanating video data, I recommend they should be considered information carrying when designing eavesdropping countermeasures.

- The blanking interval ensures the presence of the frame and line rate periodicities used for video signal identification. The quasiperiodic nature of the pixel content can also increase periodicity magnitude, but this cannot be guaranteed. The strength of the combined f_{h} and f_{v} periodicities were used as a heuristic to score potential displayed video mode harmonics, and eliminate candidates using a threshold.

- I proposed cepstrum analysis as a method of periodicity detection and compared it against the previously proposed autocorrelation. Cepstrum was found to provide better distinction between peaks and background noise. This allowed the rejection of false positive harmonics using a threshold; the autocorrelation results were too noisy in this environment to be useful, resulting in misidentified peaks.

- An eavesdropper will often have a choice of harmonics to attack, with some producing rasters with better SNRs than others. The optimal harmonic will depend on factors such as noise levels and available equipment. Not every detected harmonic will result in visible content in the raster, noted as a drawback of manual peak searching. In these instances, additional processing is required to attenuate noise and make the pixel content visible.

Chapter 4

PLL-based pixel-clock recovery for long-term complex averaging

Having recovered the operational parameters of a target VDU in Chapter 3, we can attempt to raster the eavesdropped data. We saw in Figure 3.11 that not all detected VDUs will produce visible data because of a low SNR. As the video data is quasiperiodic, and a cyclostationary process in the short-term, we can apply periodic averaging to increase the SNR. However, averaging for any appreciable period will not provide an increase (Figure 4.14a). Time-varying differences between how the emanating VDU and receiving front-end express frequency cause small changes to the pixel rate which manifest as horizontal movement of the video data (Figure 4.11a). Thus the long-term averaging necessary to increase very low SNRs will destroy the video content. The frequency difference can be measured in the phase of the IQ samples. Traditional eavesdropping attacks discard phase information by performing amplitude demodulation (Equation 2.49); in this chapter I discuss the design of a control mechanism based around a phase-locked loop (PLL) to actively track the pixel-clock frequency of an emanating VDU in the VGA data set. This will allow the use of non-uniform resampling to stabilise raster drift, permitting long-term averaging; and the ability to unrotate the IQ samples, enabling coherent periodic averaging to be performed in the complex domain. Complex averaging can increase the SNR by attenuating uncorrelated background sources which would otherwise obfuscate the video data, increasing the range at which an attack can be performed.

4.1 Clock signal waveform

The target VDU generates a pixel clock operating at nominal frequency f_p. The receiver, in our case the SDR front-end, uses an LO operating at $f_{\mathrm{c,RX}}$ to down convert harmonic h of f_p. In practice the clocks are not perfect devices, so their frequencies will differ by

$$f_\mathrm{d} = f_\mathrm{p} - \frac{f_{\mathrm{c,RX}}}{h}\,. \tag{4.1}$$

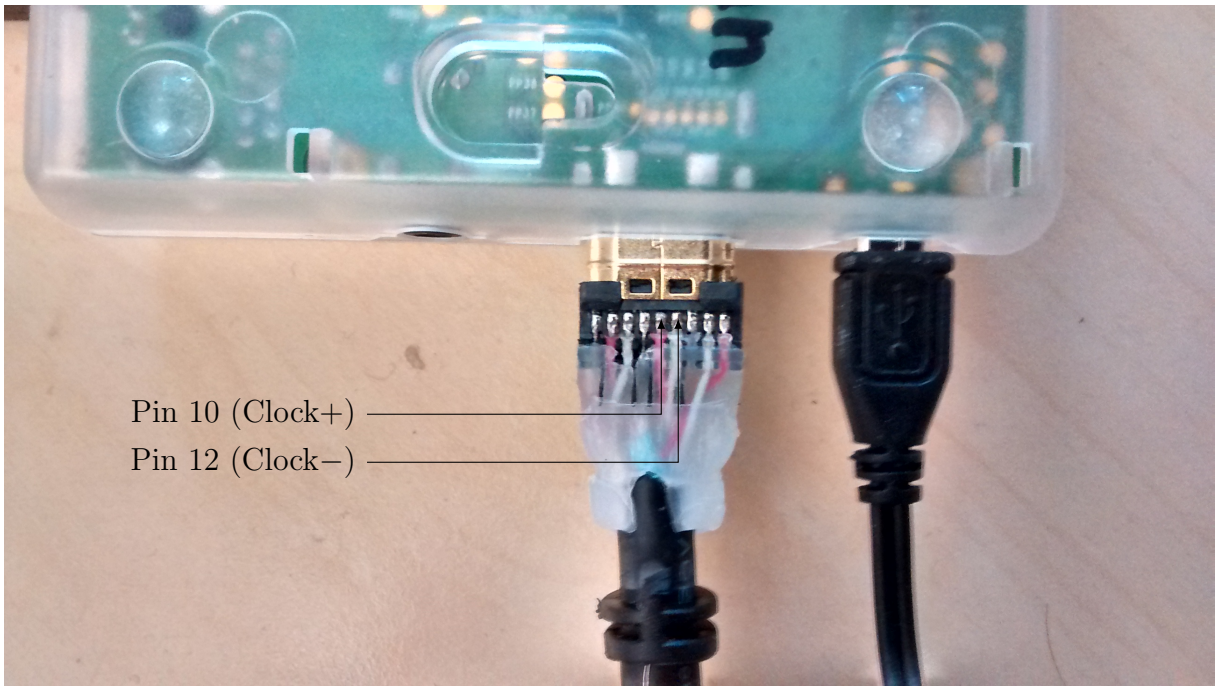


Figure 4.1. A standard HDMI connector with the casing removed to expose the pins. The clock signal is transmitted across pins 10 and 12 as the difference in current between the lines. I measured the derivative of the current using an oscilloscope with a differential probe to extract the waveform of the clock signal and produce Figure 4.2.

Furthermore, factors such as thermal changes will manifest as low frequency drift (Figure 4.4), creating an additional time-varying error component $f_{\mathrm{t}}(t)$, assumed to be normally distributed with zero mean. While drift affects both the transmitting and receiving clocks, we model the receiver as a theoretically perfect device with the transmitter expressing the sum of all time-varying frequency errors $f_{\mathrm{t}}(t)$ in the system. This choice is arbitrary but convenient for defining a single error term:

$$f_{\mathrm{e}}(t) = f_{\mathrm{d}} + f_{\mathrm{t}}(t)\,. \tag{4.2}$$

Thus after down conversion, frequency information will appear offset from its expected location by $h \cdot f_{\mathrm{e}}(t)$.

We will use the pixel clock component to estimate the frequency error. Let us begin by analysing the target signal. I removed the casing of the HDMI connector on an HDMI to DVI cable to expose the pins. I used a Tektronix P7330 differential probe connected to a Tektronix TDS7254B oscilloscope to measure the clock signal transmitted over pins 10 and 12 (Figure 4.1) and plot the waveform in Figure 4.2. It is a square wave with a 50% duty cycle, transmitting *on* for the duration of five TMDS encoded bits, followed by five *off*. We can see the uneven rising and falling edges which produce the emissions we eavesdrop. The stronger falling edge is often caused by the construction of the circuit as noted by Kuhn [93, p.35] and has been observed in other environments such as VGA cables. The waveform edges exhibit some ramping up rather than an instantaneous increase [19]. The

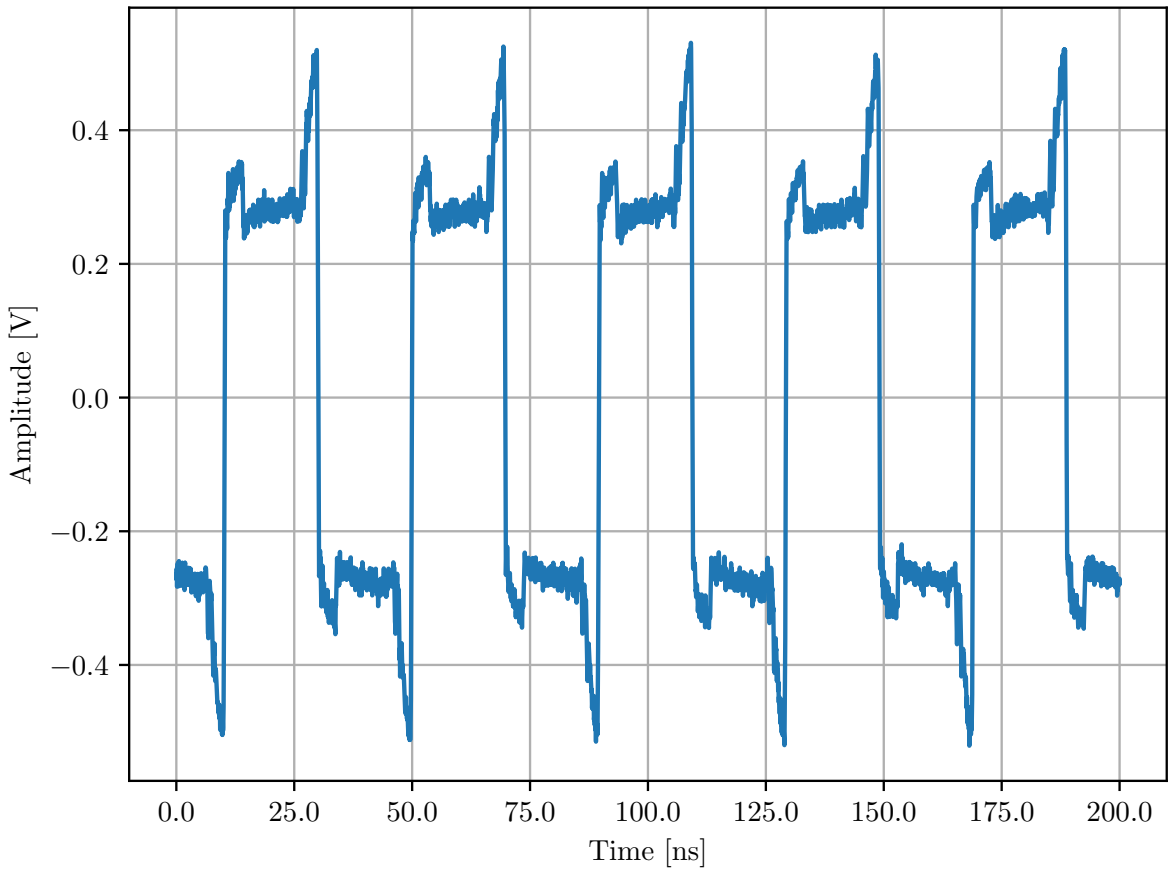


Figure 4.2. A trace of a pixel clock waveform used in TMDS to transmit a 800⊗525@25.2 MHz video signal. The measurement was made across pins 10 and 12 of the HDMI connector inserted into the Raspberry Pi using a Tektronix TDS7254B oscilloscope with a differential probe. The signal is a square wave with a 50% duty cycle. The imbalance in impedance between the clock lines manifests as uneven edges with a level of regularity. Here, the trailing edge of each pulse demonstrates a significantly larger magnitude than the leading edge. It is this effect which causes information to be emanated into the RF field. The low transition time means the harmonics will be emanated across a wide frequency band. While the standards documentation specifies $f_\mathrm{p} = 25.2$ MHz [88, p.12], the oscilloscope reported the nominal frequency to be approximately $f_\mathrm{p} = 27.17$ MHz, a deviation of approximately 8% from what we would expect.

faster the transition time of the pulse edges, the wider the bandwidth the harmonics of the clock signal will appear across [19]. As they decrease in sharpness the higher harmonics will lose power, and lower frequency harmonics will be noisier [10]. If $f_\mathrm{e}(t) = 0$ the phasor produced by a harmonic of the pixel clock down converted to DC would exhibit some stable angle, at the phase difference between the transmitting and receiving LOs. For $f_\mathrm{e}(t) \neq 0$ the phasor will rotate with an angular velocity $\omega_\mathrm{e}(t) = 2\pi h f_\mathrm{e}(t)$. We can now state our problem as estimating the vector of frequency errors $\boldsymbol{f_\mathrm{e}}$ by measuring the phase of the pixel-clock component, which can be corrected with additional stages of down conversion to keep the pixel-clock frequency component centred at 0 Hz and the complex

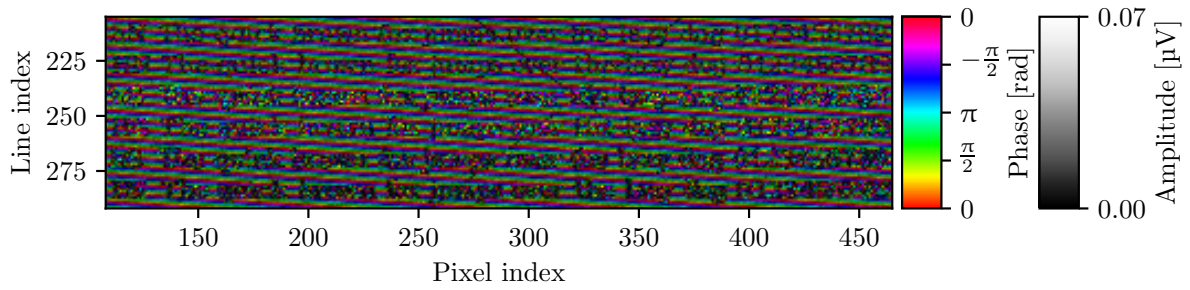


Figure 4.3. HSV raster of emanated text as it appears when output by the SDR front-end without further unrotation. The phasor makes multiple complete rotations per line due to the difference between $h \cdot f_\mathrm{p}$ and f_c. It is unsuitable for averaging in the complex domain because the different phase between pixels in consecutive frames will not have the same argument, causing the video information to be averaged out along with the noise.

phasor stable.

4.2 Pixel-clock acquisition

The first stage to correct for $\boldsymbol{f_\mathbf{e}}$ is to remove the constant f_d, which we call clock acquisition. The front-end hardware has a limit on the resolution for selecting centre and sampling frequencies. I found the resolution offered by the USRP front-ends for selecting $f_\mathrm{c,RX}$ to be approximately 1000 ppm. This is insufficient to allow the device to be accurately tuned to correct for temperature drift, which can be less than 10 ppm [21]. This motivated the use of additional stages of down conversion in software. In §3.4.3 we estimated the pixel clock frequency $f_\mathrm{p}{}'$ for some VDU with $Q = w_\mathrm{T} \otimes h_\mathrm{T} @ f_\mathrm{p}$ at a suitable harmonic h. We tune the SDR front-end to down convert from

$$f_\mathrm{c} \approx f_\mathrm{p}{}' \cdot h \,, \tag{4.3}$$

in the vicinity of the selected harmonic. The harmonic should be well within the reception band to prevent a significantly unbalanced frequency band as we further frequency convert the signal, but not too near the operating frequency of the receiving analogue LO (for direct down-converting front-ends) to prevent distortion:

$$0 < |f_\mathrm{p}{}' \cdot h - f_\mathrm{c,RX}| \ll 0.5B \,. \tag{4.4}$$

We receive the signal

$$\begin{aligned} v[n] &= \mathrm{D}\{\boldsymbol{v_\mathbf{R}}\}[n] \\ &= A[n] \cdot \mathrm{e}^{2\pi \mathrm{j} \sum_{s=0}^{n} ((f_\mathrm{p} + f_\mathrm{e}[s])h - f_\mathrm{c} + f_\mathrm{n}[s])\tau} \,, \end{aligned} \tag{4.5}$$

where $\boldsymbol{f_\mathbf{n}}$ is all frequency information unrelated to the clock signal, including both the background noise and the video data. The phase of the video component as output from

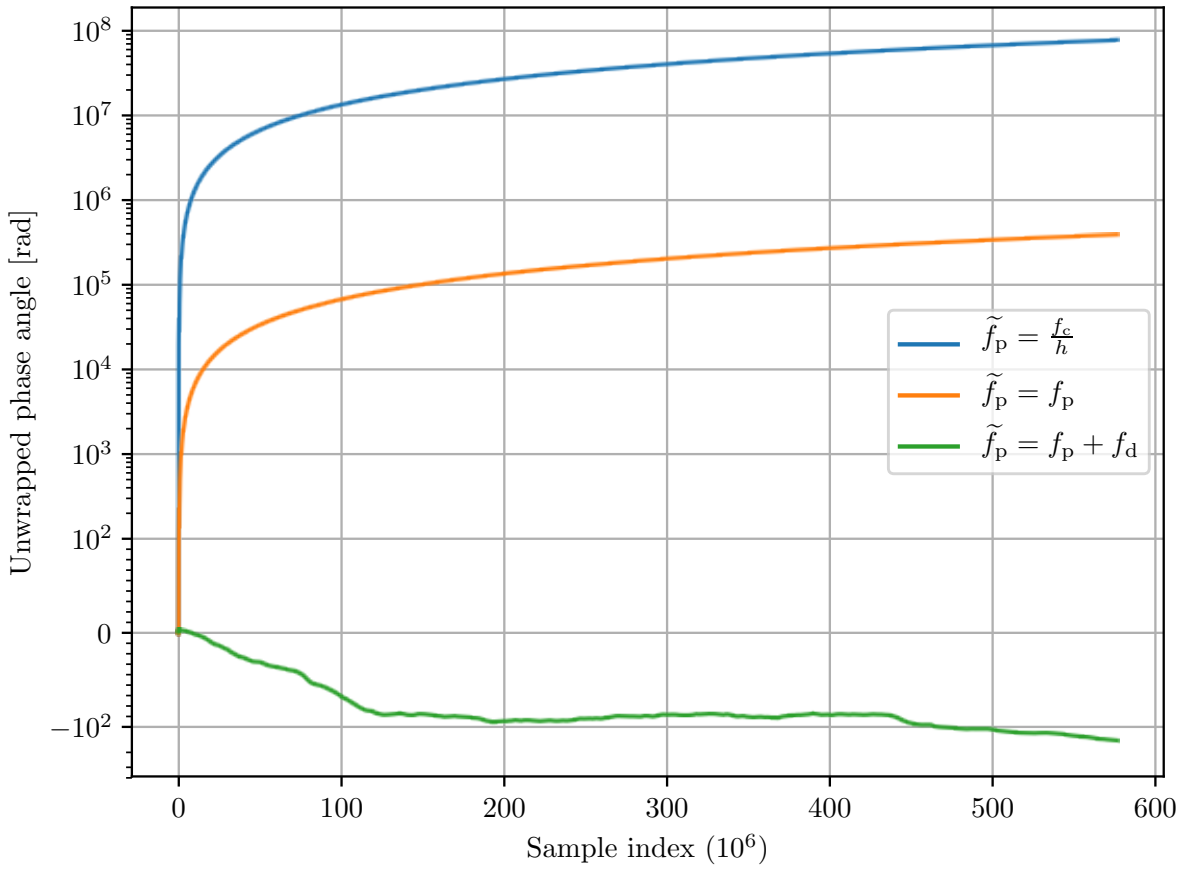


Figure 4.4. Without any processing, the phasor of $\boldsymbol{v}$ rotates rapidly over time as the frequency of the pixel clock is offset from f_c. Down converting such that the nominal f_p is at baseband reduces the rate of rotation but does not eliminate it. The offset is removed after factoring f_d into the unrotation. We see how drift remains, but no longer displays monotonic behaviour, demonstrating that unrotation by a constant factor is insufficient to keep the pixel clock at 0 Hz in the long term.

the SDR front-end will rotate at a velocity proportional to the difference between f_c and $h \cdot (f_\mathrm{p} + f_\mathrm{e}[s])$. When the CNR is strong enough, its effect on $\angle v[n]$ can be clearly seen: while a coherent raster can be produced (Figure 4.3), the complex phasor performs multiple complete rotations during a single line, demonstrated by the rapidly changing hue. Figure 4.4 shows the discrepancy produces a steep monotonically increasing curve when plotting the unwrapped phase angle. Attempting to down convert to 0 Hz by naïvely estimating the pixel-clock frequency $\widetilde{f}_\mathrm{p} = f_\mathrm{p}$ reduces, but does not eliminate, the phasor rotation. We estimate the frequency offset to be

$$f_\mathrm{d} = f_\mathrm{p}{}' - f_\mathrm{p}\,. \tag{4.6}$$

We generate a complex phasor with unit amplitude:

$$u_\mathrm{d}[n] = \mathrm{e}^{2\pi\mathrm{j}((f_\mathrm{p}+f_\mathrm{d})h - f_\mathrm{c})n/f_\mathrm{s}}\,, \tag{4.7}$$

using a numerically controlled oscillator (NCO) (§4.2.1) and multiply its conjugate, which negates the frequency, with $\boldsymbol{v}$:

$$\begin{aligned} w[n] &= v[n] \cdot u_{\mathrm{d}}[n]^* \\ &= A[n] \cdot \mathrm{e}^{2\pi \mathrm{j} \sum_{s=0}^{n} (h \cdot f_{\mathrm{t}}[s] + f_{\mathrm{n}}[s])\tau} \,. \end{aligned} \tag{4.8}$$

This frequency shift removes both the offset we introduced when setting f_{c} and f_{d}. The pixel clock is now approximately at 0 Hz and the angle no longer monotonically increases nor decreases. Rather, it continues to circle the origin due to latent drift $\boldsymbol{f_{\mathrm{t}}}$. Now when observing the raster (Figure 4.9a), we see a more uniform hue and it becomes necessary to raster multiple consecutive frames to observe phasor rotation. I recommend using a segment length of at least

$$N = \left\lceil \frac{f_{\mathrm{s}}}{1 \text{ Hz}} \right\rceil \tag{4.9}$$

when computing the PSD to obtain sub-hertz accuracy for $h \cdot f_{\mathrm{p}}$. We can say it has been acquired to within one frequency bin of resolution for the PSD used to estimate the clock location in §3.1.

4.2.1 Direct digital synthesis

We implement the NCO as a direct digital synthesiser (DDS) using fixed-point arithmetic. The algorithm is based upon a method used by the Network Time Protocol to achieve accurate timestamps [101, pp. 510–514]. We precompute a lookup table of 2^α phasor positions:

$$\Phi[i] = \mathrm{e}^{2\pi \mathrm{j} i / 2^\alpha} \quad \forall i \in \mathbb{Z}_{2^\alpha} \,, \tag{4.10}$$

alleviating the expense of repeated calls to sine and cosine or exponential functions, making the DDS efficient to run, particularly for real-time applications. We represent the current phasor position using a β bit $\geq \alpha$ bit unsigned integer $\phi \in \mathbb{Z}_{2^\beta}$. Phase is advanced by frequency $\boldsymbol{f}$:

$$\phi[n] = \left\lfloor 2^\beta \cdot \sum_{s=0}^{n} \frac{f[s]}{f_{\mathrm{s}}} \right\rfloor \bmod 2^\beta \,. \tag{4.11}$$

The modulo operation is performed implicitly when ϕ exceeds 2^β and the integer wraps around. The α most significant bits of ϕ address the bin in Φ containing the appropriate phasor value

$$u[n] = \Phi \left[\left\lfloor \frac{\phi[n]}{2^{\beta-\alpha}} \right\rfloor \right] \,, \tag{4.12a}$$

$$\approx \mathrm{e}^{2^{1-\alpha} \pi \mathrm{j} \left\lfloor 2^\alpha \left(\left(\sum_{s=0}^{n} f[s]\tau \right) \bmod 1 \right) \right\rfloor} \,. \tag{4.12b}$$

The phase is floored to the nearest index in $\boldsymbol{\Phi}$ (Figure 4.5). Increasing α provides higher phase resolution at the expense of memory. Our ability to precisely represent frequency is increased compared to using IEEE 754 double precision floating-point numbers. On a

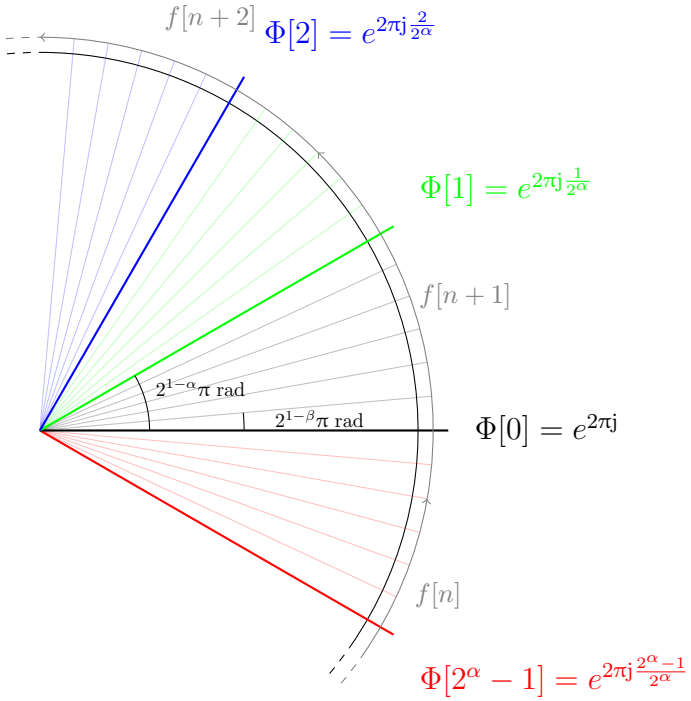


Figure 4.5. A visualisation of a segment of the DDS structure which predefines a look-up table $\mathbf{\Phi}$ of complex phasor values. Each computed phasor position is subdivided into fractional parts that quantify the phase with 2^β bits of resolution, higher than would be available using floating-point numbers. The phase is advanced by frequency $f[n]$. When the value of the phasor is read out, the most significant α bits of the phase register address the appropriate bin in $\mathbf{\Phi}$. Using $\alpha \ll \beta$ bits when precomputing the phasor positions greatly reduces the memory overhead from storing 2^β values, whilst still providing very high frequency resolution.

64 bit platform we can use $\beta = 64$ bits of frequency resolution with $\alpha = 10$ bits of phase resolution. This gives a 64 bit mantissa compared to the 53 bits provided by floating-point arithmetic. The complete procedure is detailed in Algorithm 2.

4.3 Pixel-clock tracking

To keep the pixel-clock signal at 0 Hz we use a PLL to estimate $\boldsymbol{f_t}$ and then apply a correcting factor. We must first attenuate the $\boldsymbol{f_n}$ term in $\boldsymbol{w}$ to prevent it from hindering the tracking mechanism. We design an

$$M = \left\lfloor \frac{f_\mathrm{s}}{f_\mathrm{PLL}} + 0.5 \right\rfloor \tag{4.13}$$

tap FIR low-pass filter, where $f_\mathrm{PLL} = f_\mathrm{v}$, with coefficients

$$b[i] = \frac{1}{M} \qquad \forall i \in \mathbb{Z}_M \,. \tag{4.14}$$

Algorithm 2 Direct digital synthesis of a frequency modulated complex sinusoid

procedure DDS(α,β,$\boldsymbol{f}$,f_s)
  for $i \in \{0, \ldots, 2^\alpha - 1\}$ **do**
    $\Phi[i] \coloneqq e^{2\pi \mathrm{j} i 2^{-\alpha}}$ ▷ Generate lookup table
  end for
  $\phi \leftarrow 0$ ▷ Initialise phase register
  for $n \in (0, 1, 2, \ldots)$ **do** ▷ For each output sample
    $\phi \leftarrow \phi + \frac{f[n]}{f_\mathrm{s}} \cdot 2^\beta$ ▷ Advance phase register
    $u[n] \coloneqq \Phi\left[\phi \cdot 2^{\alpha-\beta}\right]$ ▷ Retrieve complex phasor value
  end for
  return $\boldsymbol{u}$ ▷ Return array of complex values
end procedure

This filter has a zero near integer multiples of f_v and therefore is insensitive to video content other than the frame average. It has a linear phase response with a cut-off frequency $\overline{f} \approx 26.5$ Hz, removing the majority of the background frequency information but leaving some additional bandwidth to prevent the transition band from affecting the clock component:

$$\overline{w}[n] = \sum_{i=0}^{M-1} b[i] \cdot w[n-i] \tag{4.15a}$$

$$\approx \mathrm{e}^{2\pi \mathrm{j} h \sum_{s=0}^{n} f_\mathrm{t}[s]\tau} \,. \tag{4.15b}$$

We then downsample by factor M to produce the PLL's input signal

$$\breve{w}[n] = \overline{w}\left[nM\right] \,, \tag{4.16}$$

where each sample is the average value of a single video frame duration. Filtering and downsampling can improve stability because it reduces the PLL response to small phase perturbations, such as the transition from the blanking interval to the content area. The PLL outputs a complex phasor

$$u_\mathrm{t}[n] = \mathrm{e}^{2\pi \mathrm{j} h \sum_{s=0}^{n} \widetilde{f}_\mathrm{t}[s]\tau} \tag{4.17}$$

using an NCO, which estimates the phase of $\boldsymbol{f_\mathrm{t}}$ through feedback control of $\widetilde{\boldsymbol{f}}_\mathbf{t}$. We compute a complex-valued error signal

$$e[n] = \breve{w}[n] \cdot u_\mathrm{t}[n]^* \,. \tag{4.18}$$

The argument of $e[n]$ is the phase difference between $\breve{w}[n]$ and $u_\mathrm{t}[n]$, i.e. the amount the PLL needs to adjust its output phase to match the input phase. The set-point $\angle e[n] = 0$ rad will be reached when the phasors have matching arguments. Using conjugate multiplication to compute the error signal avoids phase wrapping effects at crossings

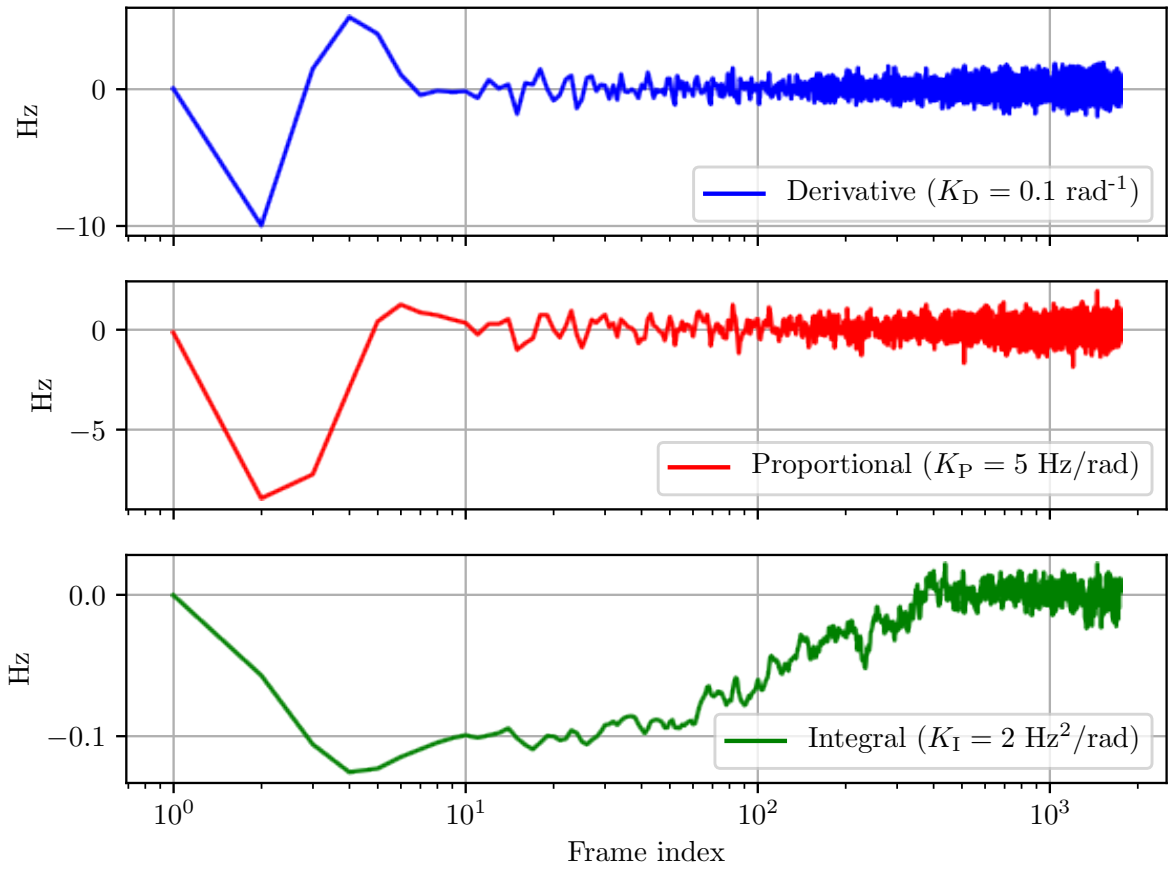


Figure 4.6. The derivative, proportional, and integral actions of the PID controller when tracking clock drift. The derivative action converges first, achieving frequency lock, with the proportional action converging shortly after as phase lock is achieved. Convergence happens quickly requiring a logarithmic scale on the abscissa to see it occurring. This indicates the drift was easy to locate and lock onto by the PLL as a consequence of the acquisition stage. Good phase lock, as indicated by the integral action, occurs much more slowly.

of the negative real axis. We feed $\boldsymbol{e}$ into a proportional-integral-derivative (PID) controller

$$h \cdot \widetilde{f}_{\mathrm{t}}[n] = \sum_{s=0}^{n} \left(K_{\mathrm{P}} \cdot \angle e[s] + K_{\mathrm{I}} \cdot \sum_{r=0}^{s} \angle e[r] \cdot \tau_{\mathrm{PLL}} + K_{\mathrm{D}} \cdot \frac{\angle (e[s] \cdot e[s-1]^*)}{\tau_{\mathrm{PLL}}} \right), \tag{4.19}$$

where $K_{\mathrm{P}} \in \mathbb{R}$, $K_{\mathrm{I}} \cdot \tau_{\mathrm{PLL}} \in \mathbb{R}$, and $K_{\mathrm{D}}/\tau_{\mathrm{PLL}} \in \mathbb{R}$ are the proportional, integral, and derivative parameters respectively. This adjusts $\widetilde{\boldsymbol{f}}_{\mathbf{t}}$ to converge upon the set-point in a smooth, controlled manner. Initially, the most important term of the PID is the derivative term, which frequency locks $\boldsymbol{u}_{\mathbf{t}}$ to $\breve{\boldsymbol{w}}$. Once the derivative action is near zero we can state the PLL has achieved frequency lock. Its strength lies in its ability to pre-empt the error. However, it can be highly susceptible to noise [102, pp. 6–9]. Thus a value lower than K_{P} or K_{I} is recommended. The proportional term removes phase drift, achieving phase lock; however, because the strength of this action is reduced as the error approaches zero, it can not completely remove all the error on its own. The remainder is removed by the integral term, which uses the long-term error to remove the remaining offset. Once

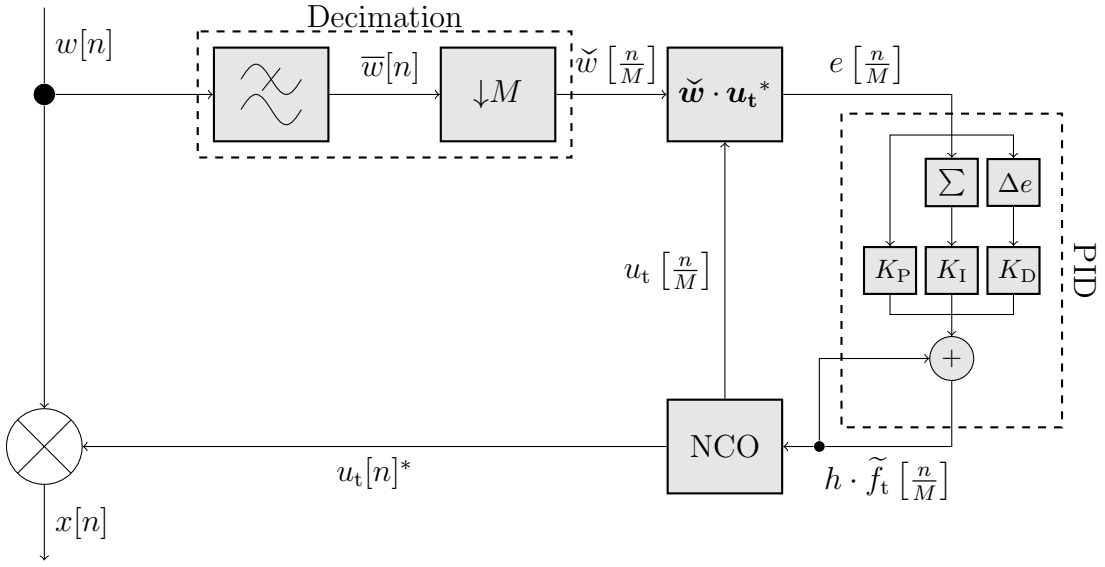


Figure 4.7. Block diagram of the PLL controller architecture used to track and unrotate the time-varying frequency error $\boldsymbol{f_t}$ of the clock signal. $\boldsymbol{w}$ is low-pass filtered to attenuate background noise, and downsampled to the PLL rate. Its phase is compared with that of the NCO to generate the complex-valued error signal $\boldsymbol{e}$. The PID controller adjusts the frequency of the NCO to converge upon the set-point $\angle e[n] = 0$ rad. The phasor generated by the NCO is used to frequency shift $\boldsymbol{w}$, unrotating the remaining frequency error of the clock signal by its estimate $h \cdot \widetilde{\boldsymbol{f}}_\mathbf{t}$.

all three actions have converged we state the PLL has *good* phase lock. I tuned the PID controller parameters to $K_\mathrm{P} = 5$ Hz/rad, $K_\mathrm{I} = 2$ Hz^2/rad, and $K_\mathrm{D} = 0.1$ rad^{-1}. Empirically these were effective in tracking the oscillator drift. The response curves of the individual PID actions when tracking a video signal are shown in Figure 4.6. The derivative action achieves lock first, with the proportional action quickly after. However, it takes longer for the integral action to converge and leave the PLL in a locked state.

We use $\boldsymbol{u_t}$ to keep the clock component at 0 Hz and 0 rad:

$$x[n] = w[n] \cdot u_\mathrm{t}[n]^* \tag{4.20a}$$

$$= A[n] \cdot \mathrm{e}^{2\pi\mathrm{j}\sum_{s=0}^{n}\left(\left(f_\mathrm{t}[s]-\widetilde{f}_\mathrm{t}[s]\right)h+f_\mathrm{n}[s]\right)\tau} \tag{4.20b}$$

$$\approx A[n] \cdot \mathrm{e}^{2\pi\mathrm{j}\sum_{s=0}^{n} f_\mathrm{n}[s]\tau}\,, \tag{4.20c}$$

removing the remainder of $\boldsymbol{f_e}$. Provided we have a good estimate $f_\mathrm{t}[n] - \widetilde{f}_\mathrm{t}[n] \approx 0$ the rotation of the clock component will be negligible and we can say the clock signal has been unrotated—the argument of the clock component phasor should remain constant. I summarise the tracking stage in Figure 4.7. When plotting the effect tracking has on the angle (Figure 4.8), we can see an almost immediate effect on the phase showing that good phase lock is not a strict prerequisite for a stable phase. This effect is further demonstrated in Figure 4.9b, where the hue displays a primarily reddish colour ($C_\mathrm{H} \approx 0$ rad). Any hue

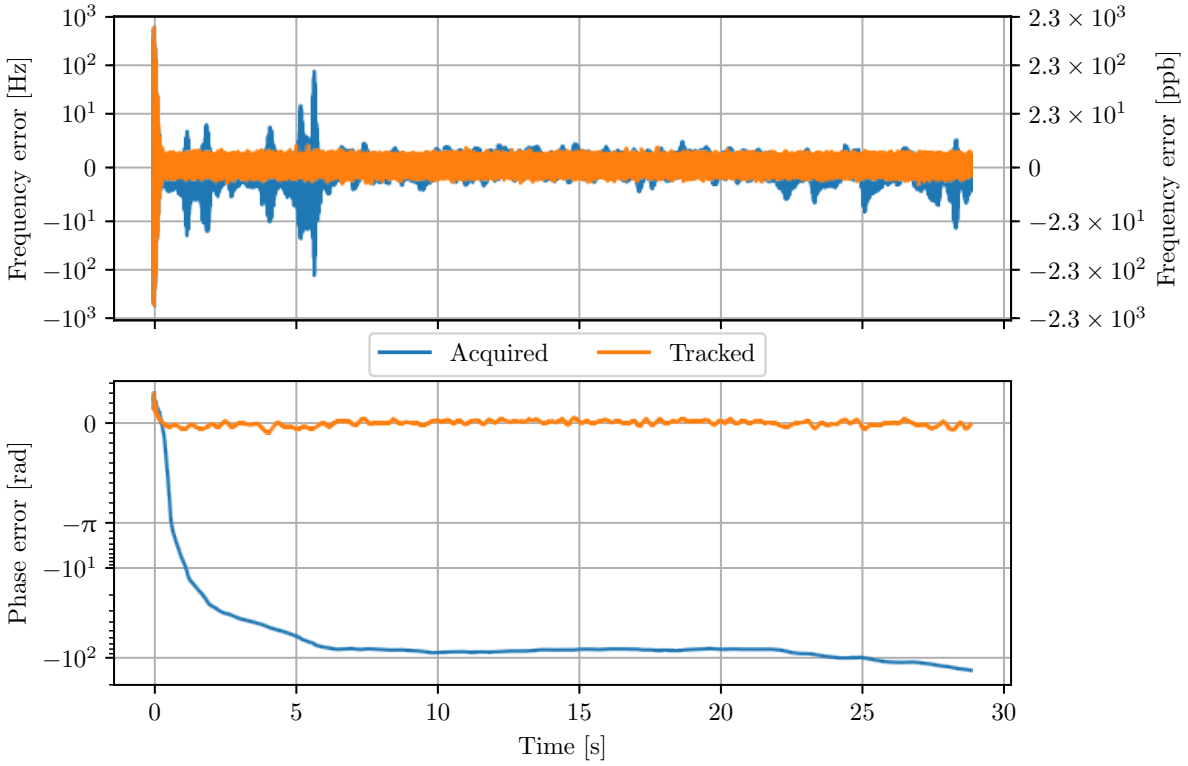
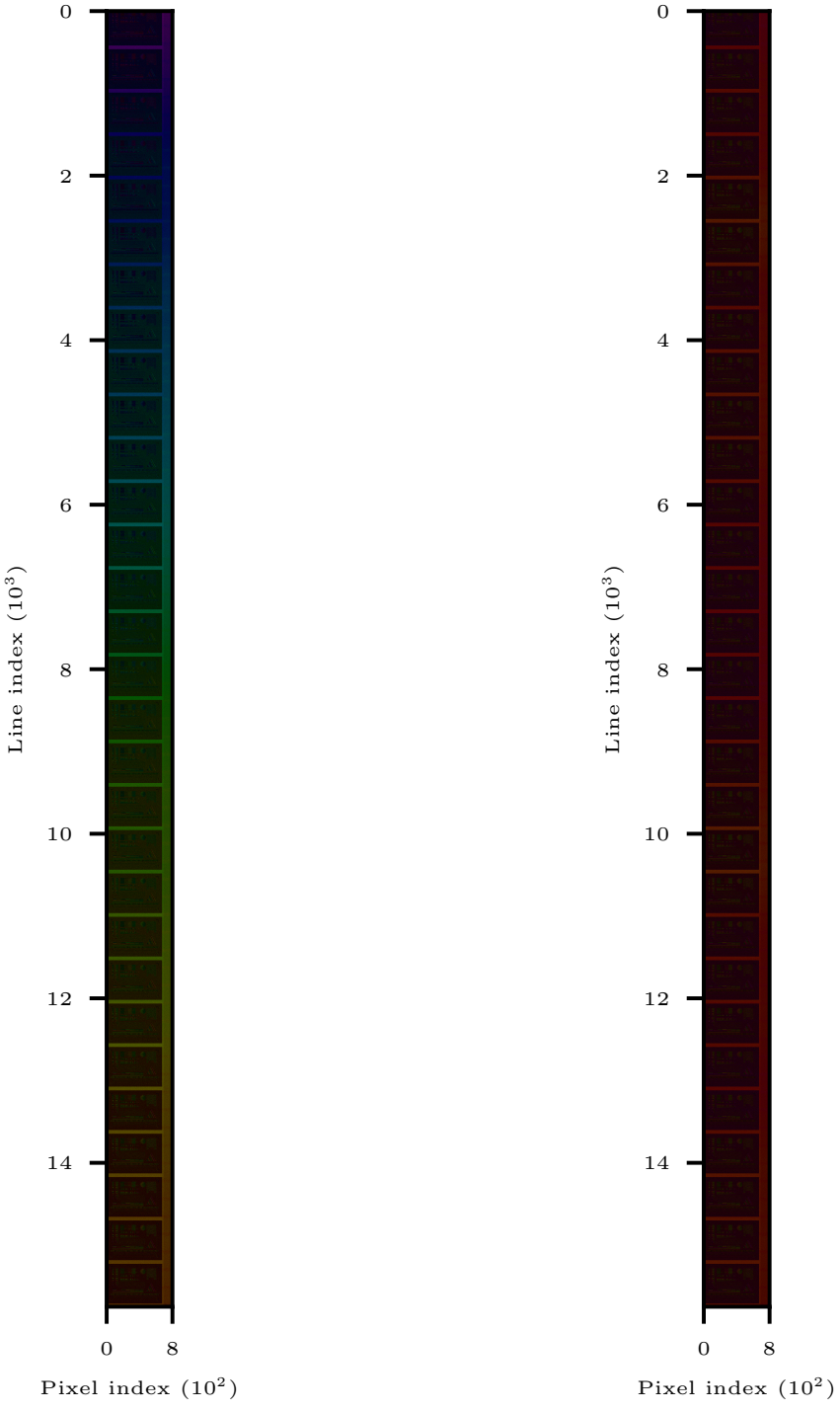


Figure 4.8. A small frequency error less than 10 ppb (parts-per notation used for harmonic invariance)—three orders of magnitude less than the value stated by Kuhn [21]—has a large effect on the phase, which varies dramatically across the duration of the signal. After tracking the signal, both the phase and frequency errors exhibit more stable behaviour. The phase remains close to 0 rad. This is reflected in the frequency error, which remains close to 0 ppb.

changes as the clock phase begins to drift are swiftly corrected as the clock signal is returned to 0 rad by the PLL. The raster hue can be used to help further tune the PLL. For example, a uniform non-red tint will occur if the PLL is frequency locked, stabilising the phasor, but is unable to return it to the real axis; therefore K_P and K_I will require adjusting. Rainbow-coloured banding emerges in the raster unless $|f_\mathrm{t}[n] - \widetilde{f}_\mathrm{t}[n]| \ll f_\mathrm{v}$, i.e. if $\widetilde{f}_\mathrm{p}$ has a small but not insignificant error—a state which may be temporary whilst the PLL acquires lock, or indicative that we should retry signal acquisition. If the bands are repeatedly changing size then $\boldsymbol{x}$ is oscillating around the real axis and K_P should be reduced to limit overshoot and K_D increased to dampen the high-frequency oscillations.

4.3.1 Oscillator stability

To measure the effect tracking has on the frequency stability of the pixel clock relative to the LO in the SDR front-end, we can use the Allan deviation (ADEV) [103]. The overlapping ADEV computes the standard deviation between the phases of overlapped



(a) Acquired **(b)** Acquired and tracked

Figure 4.9. Rasters of 30 consecutive video frames before and after tracking and unrotation. The oscillator drift can be seen manifesting as a change in hue in (a), but the effect is so gradual multiple frames need to be rastered to see the effect clearly. This contrasts with the same frames after tracking, which maintain a reddish hue indicating the angle remains at 0 rad. Some small deviations in hue are still observable, but these are quickly corrected by the PLL. Phase changes between the content area and blanking interval remain, but do not affect the operation of the PLL run at $f_{\mathrm{PLL}} = f_{\mathrm{v}}$ as it averages entire frame periods.

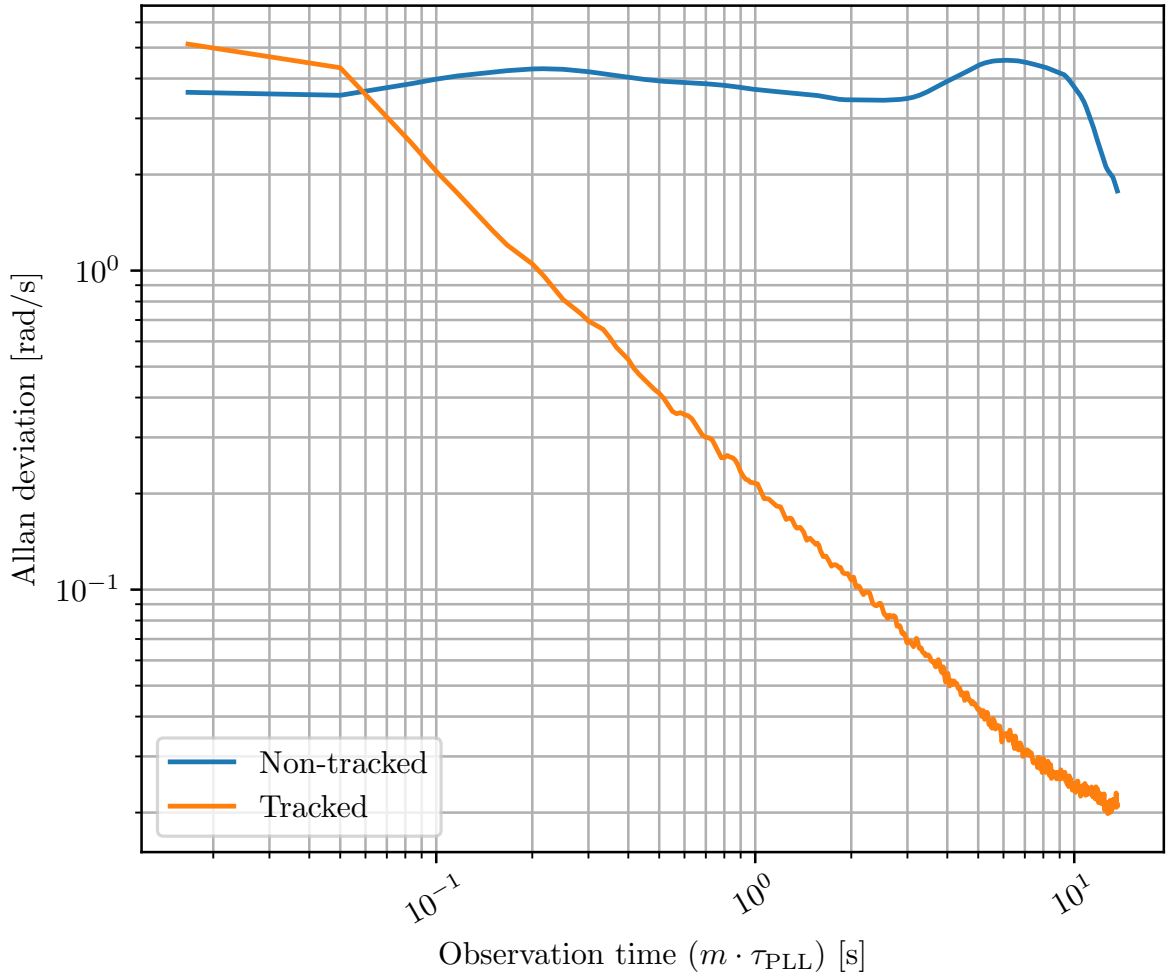


Figure 4.10. The frequency of the eavesdropped pixel clock is not stable relative to the LO in the front-end, evidenced by the meandering ADEV curve. As the observation time increases, the ADEV exhibits an approximately zero gradient indicating the presence of frequency flicker noise—low frequency noise we saw in Figure 4.8. Tracking has a clear effect on the observed stability, which appears as a decreasing curve with a gradient of negative one. The gradient indicates the precence of white phase noise, which is averaged out over longer observation periods. The minimum point in the ADEV indicates the maximum duration for averaging before the clocks are no longer stable enough relative to one another to be comparable. The PLL should prevent such an occurrence, so the presence of a local minimum could indicate a point of failure for the PLL. Here, the PLL keeps the LOs stable for the duration of the recording.

periods of m samples. It is most easily defined in terms of the unwrapped phase angle:

$$\sigma_{\breve{\boldsymbol{w}}}(m) = \sqrt{\frac{\sum_{j=0}^{N-2m-1} \left(\measuredangle\breve{w}[j] - 2 \cdot \measuredangle\breve{w}[j+m] + \measuredangle\breve{w}[j+2m]\right)^2}{2(N-2m)(\tau_{\mathrm{PLL}} \cdot m)^2}}, \tag{4.21}$$

where N is the length of the input signal. A lower ADEV as m increases indicates a long-term stable oscillator. The gradient of the curve on a log-log plot helps identify the types of noise present in the system. Figure 4.10 compares the frequency stability of the pixel

clock before and after tracking. Without tracking the curve indicates an unstable clock signal: it meanders as the observation period increases. It has a roughly zero gradient, indicating the presence of frequency flicker noise, low-frequency noise present in many electrical systems. Averaging the video signal in this state for even a short period would produce a meaningless result. However when we compare the frequency stability after unrotation we see a very different picture. The ADEV curve $\sigma_{\breve{\boldsymbol{x}}}$ where

$$\breve{x}[n] = \sum_{i=0}^{M-1} b[i] \cdot x[nM - i] \tag{4.22}$$

decreases with a gradient of negative one, indicating white phase noise. Averaging this signal will improve the SNR. If the ADEV were to plateau or begin to rise, which could occur if the PLL were to lose lock, it would indicate the point at which continued averaging would begin to reduce the SNR. Thus, we can use the ADEV to identify the maximum practical averaging duration

$$\tau_{\mathrm{a}} = \tau_{\mathrm{PLL}} \cdot \underset{m \in \mathbb{Z}_{\lfloor 0.5(N-1) \rfloor}}{\arg\min} \; \sigma_{\breve{\boldsymbol{x}}}(m) \,. \tag{4.23}$$

4.4 Variable-rate Lanczos resampling

In §2.5, a fixed ratio was used to resample from f_{s} to f_{p}. This was sufficient to raster single frames without averaging. However to average many successive frames, we require a higher level of precision to correct for the measured instability between the clocks. Naïvely assuming $f_{\mathrm{p}} = w_{\mathrm{T}} \cdot h_{\mathrm{T}} \cdot f_{\mathrm{v}}$ ($f_{\mathrm{e}} = 0$ Hz) produces a drifting raster as the video data is resampled into an incorrect number of pixels. We demonstrate the drifting effect in Figure 4.11a, where the frames drift at a rate of approximately 200 pixel/s. Any attempt to average the signal in this state would result in the video data being destroyed, even if the samples were unrotated, as we would no longer be averaging successive measurements of the same pixel. For example Lee *et al.* [104] could only use 200 ms of signal in their average. To permit long-term averaging of the signal we use $\widetilde{\boldsymbol{f}}_{\mathbf{t}}$ to drive the resampler and stabilise long-term drift by estimating the pixel clock as a time-varying function of the nominal frequency, the observed frequency offset, and the (initially zero) time-variable offset:

$$\widetilde{f}_{\mathrm{p}}[n] = f_{\mathrm{p}} + f_{\mathrm{d}} + \widetilde{f}_{\mathrm{t}}[n] \,. \tag{4.24}$$

Resampling from f_{s} to $\widetilde{f}_{\mathrm{p}}$ is similar to resampling a signal with a non-uniform sampling rate to a fixed rate. Each $x[n]$ has a corresponding resampling ratio

$$R[n] = \frac{f_{\mathrm{s}}}{\widetilde{f}_{\mathrm{p}}[n]} \,. \tag{4.25}$$

For a given index $m \in \mathbb{N}$ in the resampled output signal $\boldsymbol{y}$, we retrieve the input position $\nu \in \mathbb{R}$ in $\boldsymbol{x}$ with:

$$\nu[m] = \sum_{s=0}^{m-1} R\left[\lfloor \nu[s] \rfloor\right] \,, \tag{4.26}$$

where $\nu[0] = 0$. We interpolate the samples in $\boldsymbol{x}$ to estimate the value at $\nu[m]$ to produce the resampled output

$$y[m] = \sum_{k=-K+1}^{K} x[\lfloor \nu[m] \rfloor + k] \cdot \Phi\left((\nu[m] \bmod 1) - k\right) , \tag{4.27}$$

where Φ is some interpolation kernel of radius K. As a simple example we can linearly interpolate using the $K = 1$ kernel

$$\Phi(i) = \max(0, 1 - |i|) , \tag{4.28}$$

which reduces (4.27) to

$$y[m] = x[\lfloor \nu[m] \rfloor] \cdot (1 - (\nu[m] \bmod 1)) + x[\lfloor \nu[m] \rfloor + 1] \cdot (\nu[m] \bmod 1) . \tag{4.29}$$

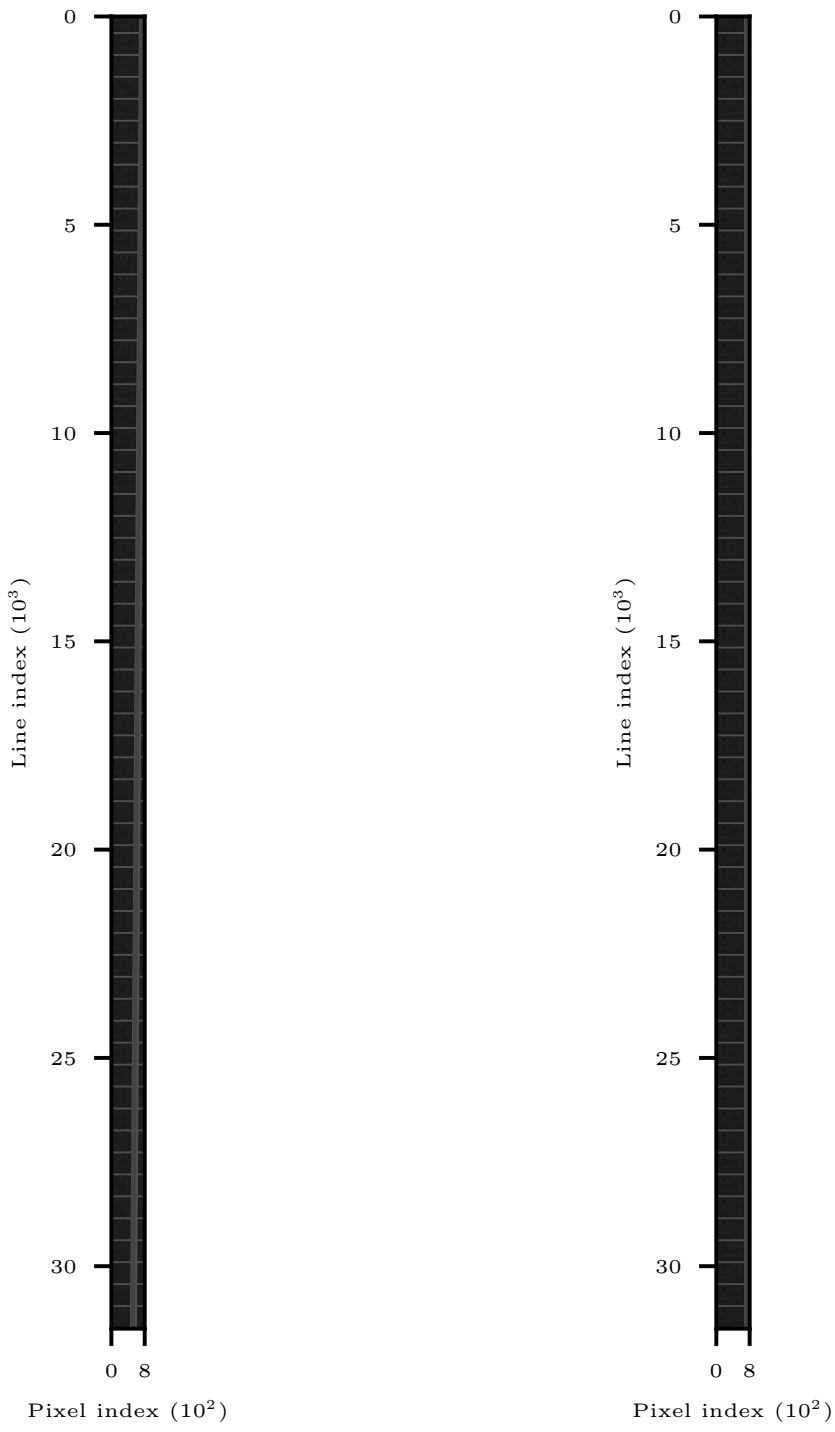
This is suited to upsampling, or downsampling when $R[n] < 2$. For higher levels of downsampling an appropriate antialiasing filter will need to be applied first. A more sophisticated approach is to use Lanczos interpolation [105, pp. 26–27] by applying a Lanczos kernel:

$$\Phi(i) = \begin{cases} 1 & i = 0 \\ K \cdot \frac{\sin(i\pi/K)\sin(i\pi)}{(i\pi)^2} & \text{otherwise} \end{cases} . \tag{4.30}$$

A normalised sinc pulse $\sin(i\pi)/(i\pi)$ is the optimal signal reconstruction filter when infinite time is available as its Fourier transform is a rectangle function. The Lanczos kernel approximates the unrealisable sinc reconstruction filter by applying two additional windowing functions: to make it finite the sinc is truncated with a time-domain rectangular window, which is applied implicitly by only generating the window over the domain $i \in [-K, K]$. Truncation introduces ripples into the frequency domain representation, so we smooth it by multiplying with the main lobe of a second scaled sinc function $\sin(i\pi/K)/(i\pi/K)$. In the Fourier domain this is equivalent to convolution with a rectangular function scaled down by K to the width of the ripples, smoothing the response. I used $K = 2$. The resampler with the Lanczos kernel is shown graphically in Figure 4.12. After resampling, successive frames remain aligned (Figure 4.11b) as the video data is resampled into the correct number of pixels. Without tracking, the accumulation of $\boldsymbol{f_t}$ over long periods will cause the raster to exhibit some lateral movement and prevent averaging over longer periods. I found an acquired but untracked signal would usually remain stable to the eye for a few seconds before exhibiting noticeable movement.

4.4.1 Resampler stability

Using a floating-point representation for $\nu[m]$ will result in a loss of precision as the exponent necessary to represent $\nu[m]$ increases. While using modulo 1 arithmetic would prevent a floating-point representation from becoming unstable over long periods, it would



(a) Naïve resampling

(b) Variable resampling

Figure 4.11. Demonstration of the movement undergone by a raster of an eavesdropped video signal when $|\widetilde{f}_\mathrm{p} - f_\mathrm{p}| \gg 0$ over the course of 60 frames. Without correction the raster drifts approximately 200 pixels to the left. After correcting for $\boldsymbol{f}_\mathbf{e}$, the video frame no longer drifts.

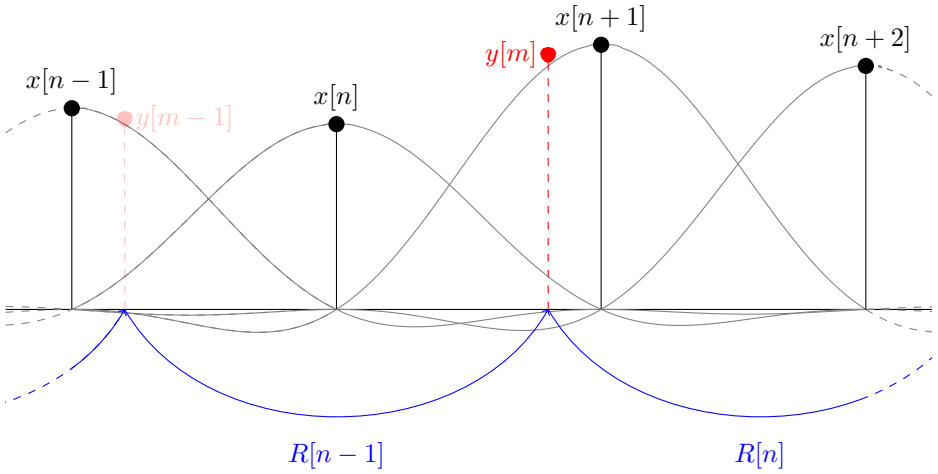


Figure 4.12. To resample from f_s to $\widetilde{f}_\mathrm{p}$ when $\widetilde{f}_\mathrm{p}$ is a time-varying function we sum the resampling ratios to locate the corresponding real-valued location in the input signal. The position determines the proportions of other samples we must combine to create the resampled output. Here, I have sketched the weighting curves for a $K = 2$ Lanczos kernel (grey). This method can be used with other interpolation techniques, such as linear interpolation, by plugging in the appropriate kernel. In practice fixed-point arithmetic is used to prevent floating-point errors accumulating, thus improving accuracy when resampling long duration signals, and increase precision.

make the exponent redundant, thus discarding bits that could otherwise be used to keep track of the resampler's position. To maintain long-term accuracy whilst maximising precision, I implemented the resampler with modular fixed-point arithmetic using an algorithm derived from the DDS (§4.2.1), summarised in Algorithm 3. The position between input sample $n \in \mathbb{Z}_{N-1}$ and $n+1$ is represented with β bits of fixed-point resolution using an unsigned integer $\phi \in \mathbb{Z}_{2^\beta}$. n is incremented by one each time ϕ exceeds $2^\beta - 1$ and wraps around.

Let us compute the resolution β required by the resampler to ensure the raster remains stable for the total duration of the averaging period τ_a. Any latent frequency error will cause the raster to drift by $|f_\mathrm{e}| \cdot 1$ pixel. Even if $|f_\mathrm{e}| < \tau_\mathrm{a}^{-1}$, information from one pixel can still begin to bleed into neighbouring pixels. We specify a limit $\varepsilon = \tau_\mathrm{a}^{-1} \cdot 0.1$ pixel on the acceptable sub-pixel drift over the entire averaging period τ_a. This equates to a maximum acceptable drift per pixel of

$$\varepsilon_\mathrm{p} = \frac{\varepsilon}{f_\mathrm{p}} \tag{4.31}$$

over τ_a. The number of bits required to represent this error is

$$\beta > -\log_2 \varepsilon_\mathrm{p} \,. \tag{4.32}$$

Algorithm 3 Variable-rate resampling using integer arithmetic

procedure RESAMPLE($\boldsymbol{x}$,$\boldsymbol{R}$,kernel,K)
  for $i \in \{0, \ldots, K \cdot 2^{\alpha}\}$ **do**
    $\Phi[i] \coloneqq \text{kernel}(i)$ ▷ Single-sided kernel lookup table
  end for
  $n \leftarrow 0$ ▷ Initialise position in $\boldsymbol{x}$
  $\phi \leftarrow 0$ ▷ Initialise intersample phase
  $\phi_{\text{old}} \leftarrow 0$ ▷ Variable to store ϕ at time $m-1$
  for $m \in (0, 1, 2, \ldots)$ **do** ▷ For each output sample
    $y[m] \leftarrow 0$ ▷ Initialise output sample
    for $k \in \{-K+1, \ldots, K\}$ **do** ▷ For each kernel element
      $y[m] \leftarrow y[m] + x[n+k] \cdot \Phi\left[\left|-k \cdot 2^{\alpha} + \left\lfloor \phi \cdot 2^{\alpha-\beta} \right\rfloor\right|\right]$ ▷ Interpolate
    end for
    $n \leftarrow n + \lfloor R[n] \rfloor$ ▷ Increment whole samples
    $\phi \leftarrow \phi + (R[n] \bmod 1) \cdot 2^{\beta}$ ▷ Increment fractional sample
    if $\phi < \phi_{\text{old}}$ **then** ▷ If ϕ has wrapped around
      $n \leftarrow n + 1$ ▷ Increment an additional sample
    end if
    $\phi_{\text{old}} \leftarrow \phi$ ▷ Cache ϕ for next iteration
  end for
  return $\boldsymbol{y}$ ▷ Return resampled signal
end procedure

To average a VDU outputting 800⊗525@25.2 MHz, which is modest by modern standards, for 20 seconds $\beta > 32.23$ bits of resolution would be required. Thus, 32 bit integers offered in many environments will be insufficient in the majority of cases. In contrast, on 64 bit platforms we can set the resolution to $\beta = 64$ bits using an unsigned long integer, which will allow 800⊗525@25.2 MHz to be resampled without drifting more than ε for a little over 2 321 years, usually sufficient.

4.5 Complex-domain averaging

The value of a (stationary) pixel at index $k \in \mathbb{Z}_{w_{\text{T}} h_{\text{T}}}$ over a series of frames can be considered a random variable $\boldsymbol{y}_k$ of which we have made multiple measurements:

$$y_k[n] = y[k + n \cdot w_{\text{T}} \cdot h_{\text{T}}]\,. \tag{4.33}$$

Each complex sample is a linear sum of phasors: the pixel phasor we wish to estimate, of amplitude μ_k and (having unrotated the signal with respect to the pixel clock) some fixed argument ϕ; a background noise component of mean μ_{n} and phase angle $\phi_{\text{n}}[n]$,

uncorrelated with the video signal and thus not aligned by the unrotation procedure; and some random noise we assume to be AWGN. This defines the complex Gaussian distribution:

$$y_k[n] \sim \mathcal{N}\left(\mu_k \cos(\phi) + \mu_\mathrm{n} \cos(\phi_\mathrm{n}[n]),\, \sigma^2\right) + \mathrm{j} \cdot \mathcal{N}\left(\mu_k \sin(\phi) + \mu_\mathrm{n} \sin(\phi_\mathrm{n}[n]),\, \sigma^2\right) . \quad (4.34)$$

An estimator ψ for the magnitude of the video signal component is unbiased if

$$\mathbb{E}\left[\psi(\boldsymbol{y}_k)\right] = \mu_k \,, \quad (4.35)$$

and when $\mathbb{V}\left[\psi(\boldsymbol{y}_k)\right]$ is the minimum variance achievable by any estimator which exists it is called the minimum variance unbiased (MVU) estimator. The Cramer-Rao lower bound (CRLB) gives the minimum variance with which a random variable can be estimated. To compute it, we take the likelihood function of $\boldsymbol{y}_k$ (assuming $\mu_\mathrm{n} = 0$) given F observations:

$$\mathrm{P}(\boldsymbol{y}_k \,|\, \mu_k, \sigma^2, F) = \prod_{i=0}^{F-1} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(y_k[i]-\mu_k)^2}{2\sigma^2}} \,, \quad (4.36)$$

apply the logarithm to obtain the log-likelihood

$$\log \mathrm{P}(\boldsymbol{y}_k \,|\, \mu_k, \sigma^2, F) = 0.5 \left(N \log(2\pi\sigma^2) - \sigma^{-2} \sum_{i=0}^{F-1} (y_k[i] - \mu_k)^2 \right) , \quad (4.37)$$

and evaluate for the second partial derivative with respect to the mean:

$$\frac{\partial}{\partial \mu_k} \log \mathrm{P}(\boldsymbol{y}_k \,|\, \mu_k, \sigma^2, F) = \frac{F}{\sigma^2} (\overline{y}_k - \mu_k) \,, \quad (4.38a)$$

$$\frac{\partial^2}{\partial \mu_k^2} \log \mathrm{P}(\boldsymbol{y}_k \,|\, \mu_k, \sigma^2, F) = \frac{F}{\sigma^2} \,, \quad (4.38b)$$

where $\overline{y}_k$ is the sample mean of the observations. Now we compute the Fisher information:

$$\mathcal{I}(\mu_k) = -\mathbb{E}\left[\frac{\partial^2}{\partial \mu_k^2} \log \mathrm{P}(\boldsymbol{y}_k \,|\, \mu_k, \sigma^2, F) \right] \quad (4.39a)$$

$$= \frac{F}{\sigma^2} \,, \quad (4.39b)$$

and invert to obtain the CRLB:

$$\mathcal{I}(\mu_k)^{-1} = \frac{\sigma^2}{F} \,. \quad (4.40)$$

If the variance of some MVU ψ matches the CRLB, it is said to be *efficient*, i.e. it is the optimal estimator.

The magnitude of $\boldsymbol{y}_k$ is a Rice distributed random variable:

$$|\boldsymbol{y}_k| \sim \mathcal{R}\left(\left| \mu_k \mathrm{e}^{\mathrm{j}\phi} + \mu_\mathrm{n} \mathrm{e}^{\mathrm{j}\phi_\mathrm{n}[i]} \right| , \, \sigma \right) . \quad (4.41)$$

A Rice distribution $\mathcal{R}(\nu, \sigma)$ is a generalisation of the Rayleigh distribution with a noncentrality parameter ν, and scale parameter σ. The expected value and variance of a Rice distributed variable are non-trivial:

$$\mathbb{E}\left[|\boldsymbol{y}_k|\right] = \sigma\sqrt{\frac{\pi}{2}} L_{0.5}\left(-\frac{\nu^2}{2\sigma^2}\right) \tag{4.42}$$

$$\mathbb{V}\left[|\boldsymbol{y}_k|\right] = 2\sigma^2 + \nu^2 - \frac{\pi\sigma^2}{2} L_{0.5}\left(-\frac{\nu^2}{2\sigma^2}\right)^2 \tag{4.43}$$

where $L_m(\cdot)$ is an m degree Laguerre polynomial. As the SNR increases, the Rice distribution tends towards approximating a Gaussian distribution, and as it reduces it tends towards a Rayleigh distribution (when $\nu = 0$ they are identical). The CRLB of a Rice distribution is an integral with no closed form [106], however from the noncentralily parameter in (4.41) we can see that by discarding the phase information and then averaging the magnitude, the noise component will contribute to the expected value. Thus the expected value will be a function of the noise and produce a biased estimate ($\mathbb{E}\left[\psi(|\boldsymbol{y}_k|)\right] \neq \mu_k$) of the video signal. The magnitude of the bias will be inversely proportional to the SNR. As it does not meet the unbiased criteria, real-valued averaging cannot be considered efficient in the presence of non-AWGN noise.

Now let us consider complex averaging F measurements of $\boldsymbol{y}_k$. By averaging prior to amplitude demodulation in the complex domain, uncorrelated background components will be attenuated. We first sum together the measurements: the means of the distributions add linearly and, as the samples are i.i.d., so do their variances (for brevity I omit the imaginary component):

$$\sum_{i=0}^{F-1} \Re\left(y_k[i]\right) \sim \mathcal{N}\left(F \cdot \left(\mu_k \cos(\phi) + \mu_\mathrm{n} \sum_{i=0}^{F-1} \cos(\phi_\mathrm{n}[i])\right),\ F\sigma^2\right) . \tag{4.44}$$

Scaling the sum by F^{-1} to estimate the average

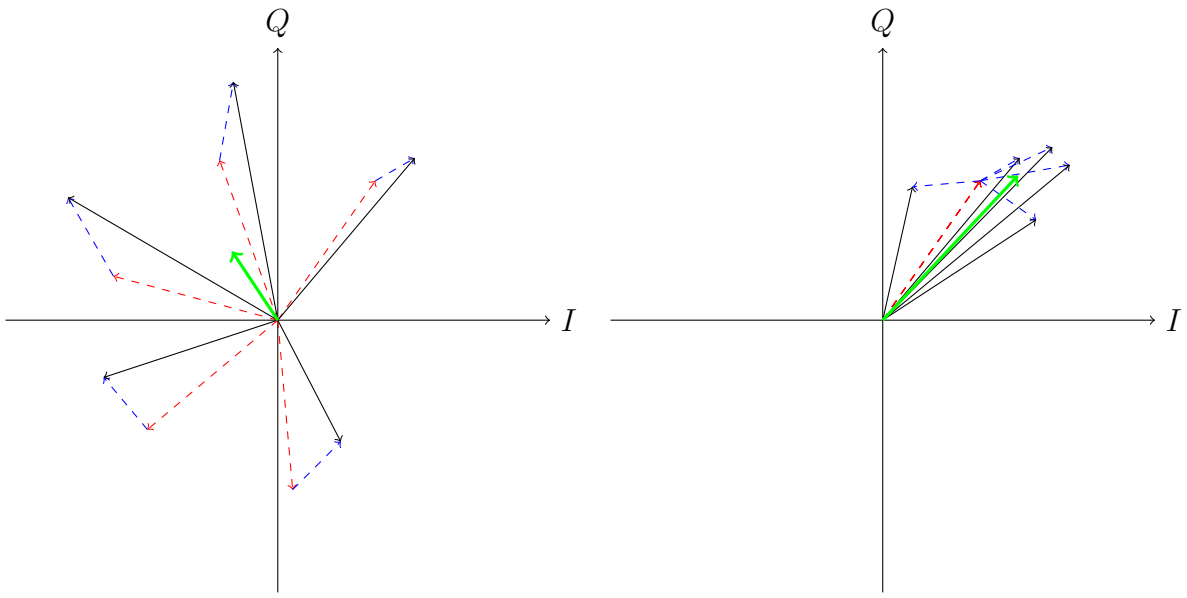
$$\psi_F\left(\boldsymbol{y}_k\right) = \frac{1}{F} \sum_{i=0}^{F-1} y_k[i] \tag{4.45}$$

is linear for the mean such that

$$\begin{aligned}\mathbb{E}\left[\Re\left(\psi_F\left(y_k[i]\right)\right)\right] &= \frac{F\left(\mu_k \cos(\phi) + \mu_\mathrm{n} \sum_{i=0}^{F-1} \cos(\phi_\mathrm{n}[i])\right)}{F} \\ &= \mu_k \cos(\phi) + \mu_\mathrm{n} \sum_{i=0}^{F-1} \cos(\phi_\mathrm{n}[i]) .\end{aligned} \tag{4.46}$$

When the background noise component is orthogonal to the desired signal component (their cross-correlation (§5.2.2) is zero) the phase terms will be uniformly distributed:

$$\boldsymbol{\phi}_\mathbf{m} \sim \mathcal{U}\left(-\pi, \pi\right) , \tag{4.47}$$



(a) Rotated **(b)** Unrotated

Figure 4.13. Each complex phasor (black) decomposes into a signal (red) and noise (blue) component. When the components have not been aligned, complex averaging results in a poor estimate (a, green) as the signal phasors cancel out, distorting the average. Averaging after amplitude demodulation in the real domain does not fair much better as the noise and signal components both contribute to the average, reducing the noise attenuation. Tracking and unrotating the signal component aligns the signal components in the complex domain (b, red). The same unrotation process is applied to the noise components, but as they are uncorrelated with the signal component they remain unaligned. Now, when averaging in the complex domain the noise components still cancel out, but the aligned signal components produce a more representative average (b, green).

with an expected value of zero, removing the background noise term:

$$\mathbb{E}\left[\Re\left(\psi_F\left(y_k[i]\right)\right)\right] = \mu_k \cos(\phi)\,. \tag{4.48}$$

The variance is scaled by the square of the scaling factor:

$$\begin{aligned}\mathbb{V}\left[\Re\left(\psi_F\left(\boldsymbol{y}_k\right)\right)\right] &= \frac{F\sigma^2}{F^2} \\ &= \frac{\sigma^2}{F} \\ &= \mathcal{I}(\mu_k)^{-1}\,.\end{aligned} \tag{4.49}$$

Thus, the mean of F samples of $\boldsymbol{y}_k$ is defined over the distribution:

$$\psi_F(\boldsymbol{y}_k) \sim \mathcal{N}\left(\mu_k \cos(\phi),\, \frac{\sigma^2}{F}\right) + \mathrm{j}\cdot\mathcal{N}\left(\mu_k \sin(\phi),\, \frac{\sigma^2}{F}\right)\,. \tag{4.50}$$

Now, $\left|\mathbb{E}\left[\psi_F(\boldsymbol{y}_k)\right]\right| = \mu_k$ and $\mathbb{V}\left[\psi_F(\boldsymbol{y}_k)\right] = \mathcal{I}(\mu_k)^{-1}$. Therefore, for a signal where the noise is AWGN or uncorrelated, complex averaging is both the MVU estimator and efficient when the target component has been unrotated.

Provided the PLL maintains lock, the angle of the video component will remain constant indefinitely, allowing complex averaging to be applied for as many frames as the pixel signal remains stationary (Figure 4.13), and the resolution of the resampler permits. For each additional frame used in the average, we can expect averaging to increase the SNR proportionally to the root of the number of frames:

$$\varrho_{\psi_F(\boldsymbol{y}_k)} = \frac{|\mathbb{E}\left[\psi_F(\boldsymbol{y}_k)\right]|}{\sqrt{\mathbb{V}\left[\psi_F(\boldsymbol{y}_k)\right]}} \tag{4.51a}$$

$$= \frac{\mu_k}{\sqrt{\frac{\sigma^2}{F}}} \tag{4.51b}$$

$$= \sqrt{F} \cdot \frac{\mu_k}{\sigma}\,, \tag{4.51c}$$

$$\therefore \varrho \;\propto\; \sqrt{F}\,. \tag{4.51d}$$

To apply periodic averaging to $\boldsymbol{y}$ for F consecutive frames we use an $N = w_\mathrm{T} \cdot h_\mathrm{T} \cdot (F-1)$ order FIR filter:

$$z[m] = \sum_{i=0}^{N} b[i] \cdot y[m-i]\,, \tag{4.52}$$

with filter coefficients

$$b[i] = \begin{cases} \frac{1}{F} & i \bmod w_\mathrm{T} \cdot h_\mathrm{T} = 0 \\ 0 & \text{otherwise} \end{cases}. \tag{4.53}$$

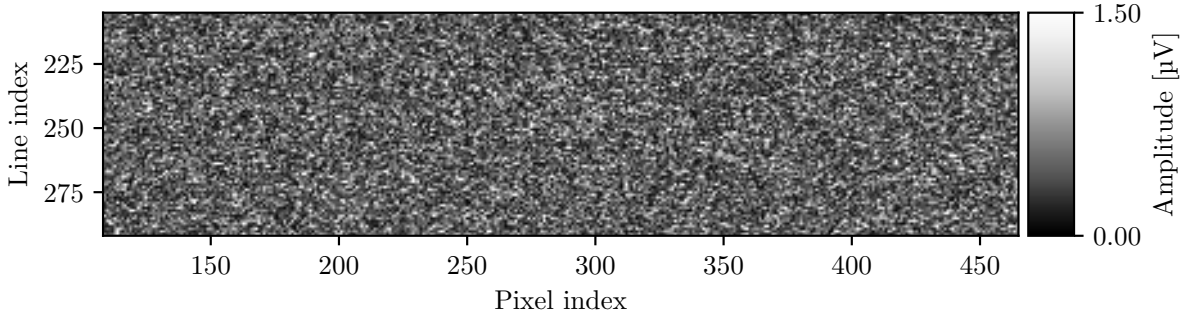
While conceptually simple to understand and implement, a large F can require an impractical amount of memory: with 16 GB of memory just 26 seconds of resampled emanations from a 1080p VDU (2200⊗1125@154 MHz [92, p.88]) could be averaged when storing complex samples as two 2 byte short integers—somewhat short of the theoretical limit of 379 years of stability permitted by the resampler ($\varepsilon = \tau_\mathrm{a}^{-1} \cdot 0.1$ pixel). Alternatively, we can use an *integrate and dump* filter:

$$z[m] = \frac{1}{F} \sum_{i=0}^{F-1} y[m - (i \cdot w_\mathrm{T} \cdot h_\mathrm{T})]\,. \tag{4.54}$$

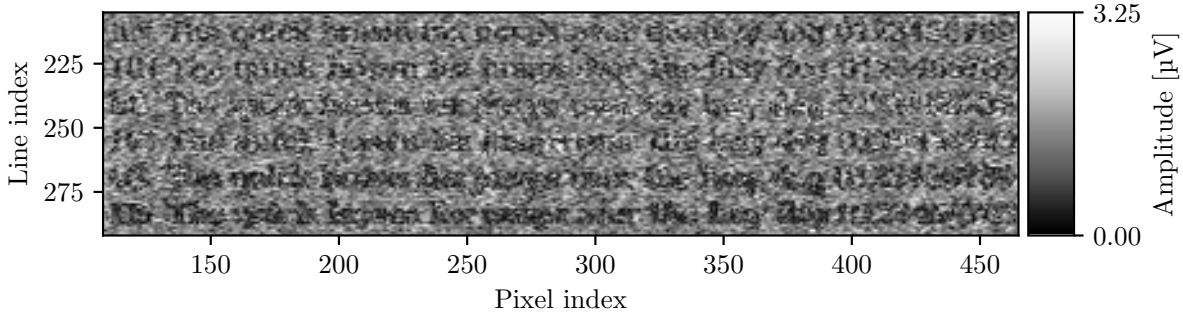
Samples are loaded from secondary storage, added to the result, and removed from memory immediately, reducing the space complexity to a single array of $w_\mathrm{T} \cdot h_\mathrm{T} \cdot 2 \cdot 2$ bytes. This estimator is still efficient: for offline processing it is equivalent to Equation 4.52. However if processing is to be performed online, only one averaged frame can be output at the end of the integration period after which we must discard the result, reducing the raster frame rate to $f_\mathrm{v} \cdot F^{-1}$. An alternative estimator is an infinite impulse response (IIR) exponential decay filter:

$$z[m] = \gamma \cdot \sum_{i=0}^{\infty} (1-\gamma)^i \cdot y[m - i \cdot w_\mathrm{T} \cdot h_\mathrm{T}] \tag{4.55a}$$

$$= \gamma \cdot y[m] + (1-\gamma) \cdot z[m - w_\mathrm{T} \cdot h_\mathrm{T}]\,, \tag{4.55b}$$



(a) $\widetilde{f}_\mathrm{p} = w_\mathrm{T} \cdot h_\mathrm{T} \cdot f_\mathrm{v}$.



(b) $\widetilde{f}_\mathrm{p} = w_\mathrm{T} \cdot h_\mathrm{T} \cdot f_\mathrm{v} + f_\mathrm{d}$ ($f_\mathrm{d} = 120.5$ Hz).

Figure 4.14. Complex-domain averaging $\boldsymbol{v}$ for 1728 frames has no visible effect when assuming the pixel clock frequency to be the nominal value (a) which, like rasters of individual frames, exhibits only static. Additonally unrotating by constant frequency $h \cdot f_\mathrm{d}$ prior to averaging (b) increases the SNR to a level where faint traces of features can be discerned. However the text is still unreadable. If we were to increase the averaging period, the features would be averaged out as drift accumulates.

where $\gamma \in (0, 1]$ is a smoothing factor which controls the weighting between newer and older frames. In the most extreme case, when $\gamma = 1$, the filter has no effect on the input. As γ decreases, older values are given a higher weighting and the filter better approximates the long-term average and increases the SNR, provided pixel content remains stationary. Pixels presented to this filter will have a (theoretically) permanent effect on the output. The resulting distribution (assuming $\mu_\mathrm{n} = 0$) for this estimator is

$$\psi_\gamma(\boldsymbol{z}_k) \sim \mathcal{N}\left(\mu_k \cos(\phi),\, \frac{\gamma\sigma^2}{2-\gamma}\right) + \mathrm{j} \cdot \mathcal{N}\left(\mu_k \sin(\phi),\, \frac{\gamma\sigma^2}{2-\gamma}\right). \tag{4.56}$$

As $\mathbb{V}\left[\psi_\gamma(\boldsymbol{z}_k)\right] > \mathcal{I}(\mu_k)^{-1}$ it cannot be considered efficient. Despite these drawbacks, exponential smoothing offers a good compromise for online processing, having the same space complexity as the integrate and dump filter, and outputting a frame every ${f_\mathrm{v}}^{-1}$.

After complex averaging has been applied, $\boldsymbol{z}$ can be demodulated and rastered as appropriate (§2.6). Let us now attempt to average a video signal with a poor SNR recorded at $d = 9$ m for 1728 frames using an integrate and dump filter. At this distance, no video data is observable when rastering individual frames. This does not change when

we down convert so the nominal f_p is at DC (Figure 4.14a). When averaging an acquired but not tracked signal (unrotated by $(f_\mathrm{p} + f_\mathrm{d})h - f_\mathrm{c}$), some of the stronger features begin to appear (Figure 4.14b). However they remain largely obscured by the noise. This is a consequence of the accumulation of the drifting we observed in Figure 4.4, and over a longer period would destroy the video information. Once we track and unrotate $h \cdot \boldsymbol{f_\mathbf{t}}$ using the PLL we see a dramatic improvement as a raster with clearly defined features is lifted out from what was unintelligible noise (Figure 4.15b). Without the tracking component, eavesdropping at this SNR would not have been viable. At each stage of unrotation we see a corresponding rise in the amplitude of the average. Unrotating by $h \cdot f_\mathrm{d}$ results in a relatively minor increase of 1.75 µV, and tracking increases the amplitude more than threefold. This is because more of the video signal component remains after averaging at each stage of unrotation, instead of being averaged out along with the noise. Now let us compare the average of the tracked signal using real-domain averaging, after amplitude-demodulation (Figure 4.15a). The image quality has significantly degraded, with the features obfuscated by the noise. The noise has some structure, as evidenced by a banding effect, indicating it is a product of some background source rather than simple AWGN. As the source of the noise is operating at a different frequency to the eavesdropped harmonic, it is not aligned by unrotation and attenuated by the complex averaging in Figure 4.15b. There is a large increase in both the maximum and minimum amplitudes compared to the complex average as the noise now contributes significantly to the amplitude. Thus, we can see how coherent complex averaging provides a significant increase in SNR compared to post-demodulation averaging.

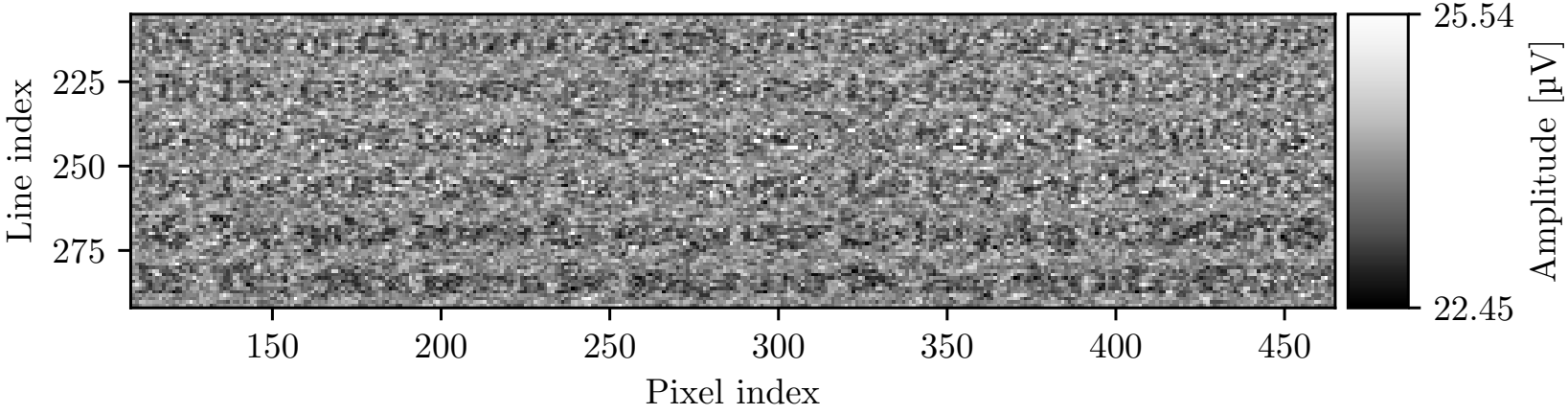
4.6 Viable eavesdropping distance

Figure 4.16 plots the estimated SNR (§2.8) of rasters produced from averaging F frames at different simulated distances. When plotted against the logarithm of $\sqrt{F}$, which gives the expected decibel increase in SNR, the curves are linear, supporting the assertion made in §4.5 that $\varrho \propto \sqrt{F}$, meaning AWGN is a suitable model for the background noise here. This is an important relationship to consider when designing eavesdropping countermeasures. Shielding efficacy of a shielding strategy[1] is measured by the attenuation level of the emanated field strength from E to E_s [107]:

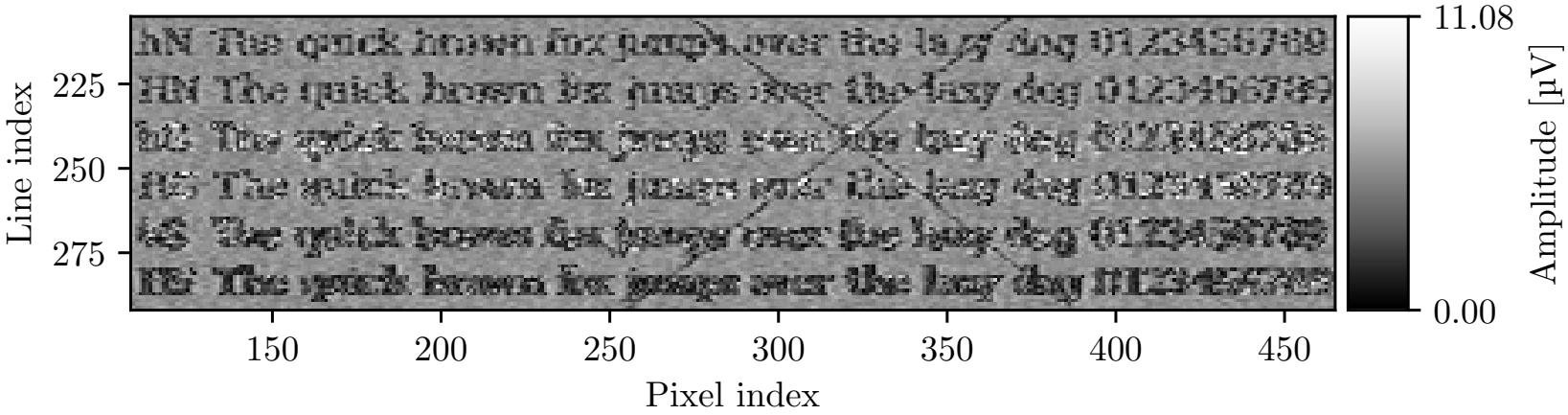
$$\Xi = 20 \text{ dB} \cdot \log_{10} \frac{E}{E_\mathrm{s}} \,. \tag{4.57}$$

A countermeasure designer may decide the threshold SNR at which an eavesdropper could recover sensitive information is ϱ_T. This value will depend on factors such as the features of the sensitive data, e.g. small text or large images. SNR is essentially a measure of

[1]Here I use the term shielding to refer to all approaches to attenuate the signal available to an eavesdropper.



(a) Real-domain averaging.



(b) Complex-domain averaging.

Figure 4.15. When unrotating by $f_\mathrm{d} = 120.5$ Hz and tracking, but averaging (1728 frames) in the real domain (a), the noise in the raster demonstrates structure, obscuring the video data. Comparatively, when averaging in the complex domain (b) we recover clearly defined features: the text becomes readable, and the cross is now clearly observable. The amplitude is much lower than the real-valued average, where both the maximum and minimum amplitudes are greater with lower dynamic range as the noise contributes significantly instead of being averaged out.

risk: the noise is most likely not stationary and can change over time. Therefore any assumptions should be for the worst-case scenario. They place their shielded device and measure the SNR at distance d_0 as ϱ_0. Assuming free-space propagation, the SNR at d is estimated:

$$\varrho_d = \varrho_0 + 20 \text{ dB} \cdot \log_{10} \frac{d_0}{d} . \tag{4.58}$$

This relationship can be observed between the curves in Figure 4.16. The designer proposes a controlled zone around the device of distance d to allow emanations to attenuate to $\varrho_d < \varrho_\text{T}$ before they can be intercepted. However the eavesdropper can track the clock signal and average the video signal to bring the SNR back up to the vulnerable ϱ_T level by averaging for

$$\tau_\text{a} = {f_\text{v}}^{-1} \cdot 10^{(\varrho_\text{T} - \varrho_d)/10 \text{ dB}} , \tag{4.59}$$

which will increase the SNR by

$$\varrho_{\Delta,\tau_\text{a}} = 20 \text{ dB} \cdot \log_{10} \sqrt{\tau_\text{a} \cdot f_\text{v}} . \tag{4.60}$$

The efficacy should now be considered reduced for every additional frame of signal an eavesdropper can acquire. I present a modified formula to measure shielding efficacy given the duration of the sensitive task, i.e. the longest time an eavesdropper could average for in the worst-case scenario:

$$\Xi_\text{a} = 20 \text{ dB} \cdot \left(\log_{10} \frac{E}{E_\text{s}} - \log_{10} \sqrt{\tau_\text{a} \cdot f_\text{v}} \right) . \tag{4.61}$$

For standards that define limit curves, the curves should be considered reduced by $\varrho_{\Delta,\hat{\tau}}$, where $\hat{\tau}$ is the maximum duration an instance of sensitive screen content will be displayed. This ensures the eavesdropper cannot collect enough signal to negate the shielding strategy. Thus, shorter tasks are inherently more secure. For example a VDU with quickly scrolling text will present fewer frames to a potential eavesdropper, allowing for a smaller ϱ_Δ; while a surveillance camera display is unlikely to change quickly and would allow more frames to be collected. As these calculations cannot account for local environmental features, practical measurement and experimentation will always be required to fine-tune precautionary measures, and must account for differences in the SNR between harmonics (§3.5).

4.7 Summary

I have demonstrated how pixel clock recovery can stabilise the drifting raster effect and unrotate the IQ down converted video signal to permit averaging in the complex domain. I summarise the findings:

- A slightly inaccurate estimate of f_p can be used to recover image content, but continual changes from oscillator drift will cause corresponding movement in the

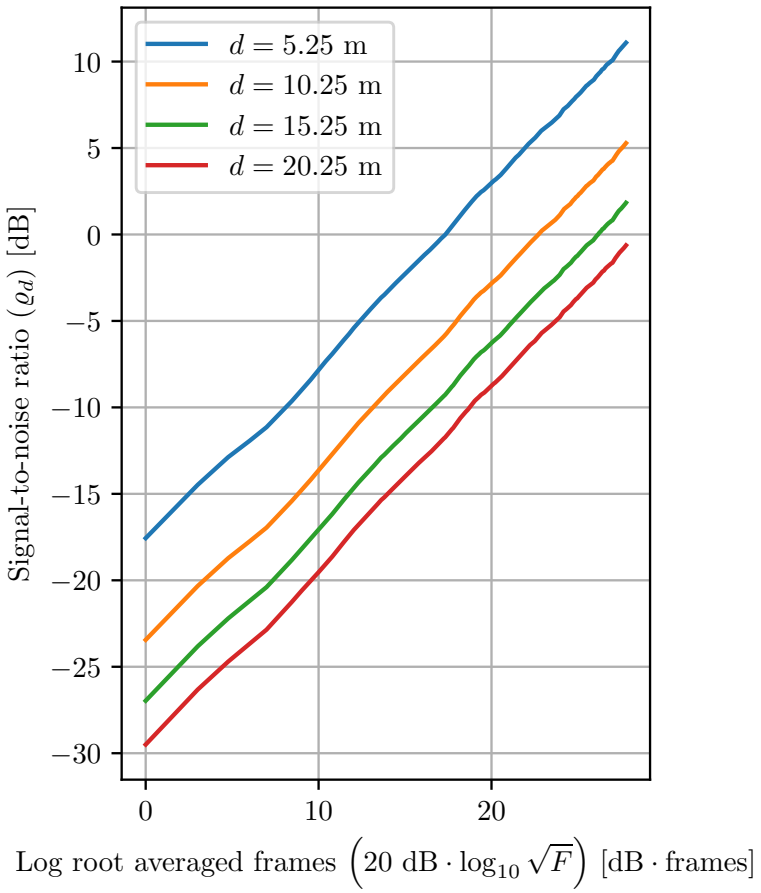


Figure 4.16. The estimated SNR of rasters produced by complex averaging F frames of tracked and unrotated video signal. The curves approximate the relationship $\varrho \propto \sqrt{F}$ demonstrating the assumption that the noise is AWGN is sufficient. This relationship allows prediction of the amount of signal data that must be collected by an eavesdropper to increase the SNR to a level at which sensitive data can be recovered.

reconstructed video image. We can eliminate much of the frequency discrepancy between the LOs in the target VDU and receiving SDR front-end by using the detected value for f_p (Chapter 3) to further up or down convert the signal. However, the pixel-clock frequency is not static. Rather, it will continually drift due to factors such as temperatures changes and invalidate the estimate in the long term.

- The frequency drift of the pixel clock can be measured as the relative change of phase in the down converted IQ samples. I designed a PLL which low-pass filters and downsamples the IQ samples before passing them though a phase detector. A PID controller then converges upon the phase of the pixel-clock component.

- Using the frequencies detected by the PLL, I was able to resample the eavesdropped samples to the exact pixel-clock frequency, eliminating raster drift. By using variable-resampling ratios we can ensure long-term stability of the video data, which permits long-period averaging to be applied without averaging out the video

data.

- I also use the PLL to drive an NCO, creating a phasor phase locked to the drifting pixel-clock component. Multiplying the conjugate of the NCO phasor with the IQ samples both frequency and phase aligns the pixel clock component with DC (0 Hz). As such, the same pixel value results in the same expected phase across frames. This allows for complex averaging, minimising the attenuation of video data, whilst maximising the attenuation of uncorrelated noise compared to averaging in the real domain.

- TEMPEST standards use emission-limit curves to determine acceptable emanation levels. A controlled zone allows emissions to dissipate to the desired level, mitigating the risk of interception by an eavesdropper. Averaging increases the range at which an eavesdropping attack can be performed in a predictable manner. Shielding recommendations should be adjusted accordingly to maintain a desired level of EmSec based on the duration of the sensitive task and the size of the controlled zone.

Chapter 5

Stitching multiple narrowband recordings into a single wideband

So far we have estimated screen content from the phase and amplitude of up to 20 MHz wide bandpass filtered emanations. While this allows readable text by distinguishing between the foreground and background, it leaves us unable to recover features such as relative brightness and colour. To recover more information we must intercept enough bandwidth to discern bit data transmitted across the RGB lines (Figure 5.1). Eavesdropping bits via radiative coupling is not a new concept. Smulders' investigation of emissions from RS-232 serial data transmission cables [108] highlighted high bit amplitudes, short transition times, serial transmission, and unbalanced connections as aides to eavesdropping bit data. There are two main differences from our scenario: first, while each encoder transmits bits serially, a total of four wire pairs transmit concurrently (three video and one clock), creating a semi-parallel interface. Secondly, RS-232 has a comparatively low bit rate (e.g. 9.6 kbit/s), meaning information is contained within a bandwidth small enough to allow data capture using a simple low bandwidth AM radio receiver. Video data is emanated across a wide bandwidth, which increases in size with the sample rate of the source i.e. f_{p} and f_{b}. For high resolution VDUs, the bandwidth required to recover individual bits becomes significantly wider than what commercially available SDR front-end's are capable of. Recording capabilities are limited by the bandwidth of the SDR front-end. However, as it can be tuned over a wide frequency range, we can estimate wideband frequency content by combining multiple narrowband measurements. Because making multiple recordings simultaneously would require a rack of SDR front-ends, we can take advantage of the signal's quasiperiodic properties to recombine consecutive recordings made by a single front-end. In this chapter I recover wideband estimates from multiple narrowbands in the Stitch data set. We can then use a hidden Markov model (HMM) to identify sets of concurrently emanated TMDS characters.

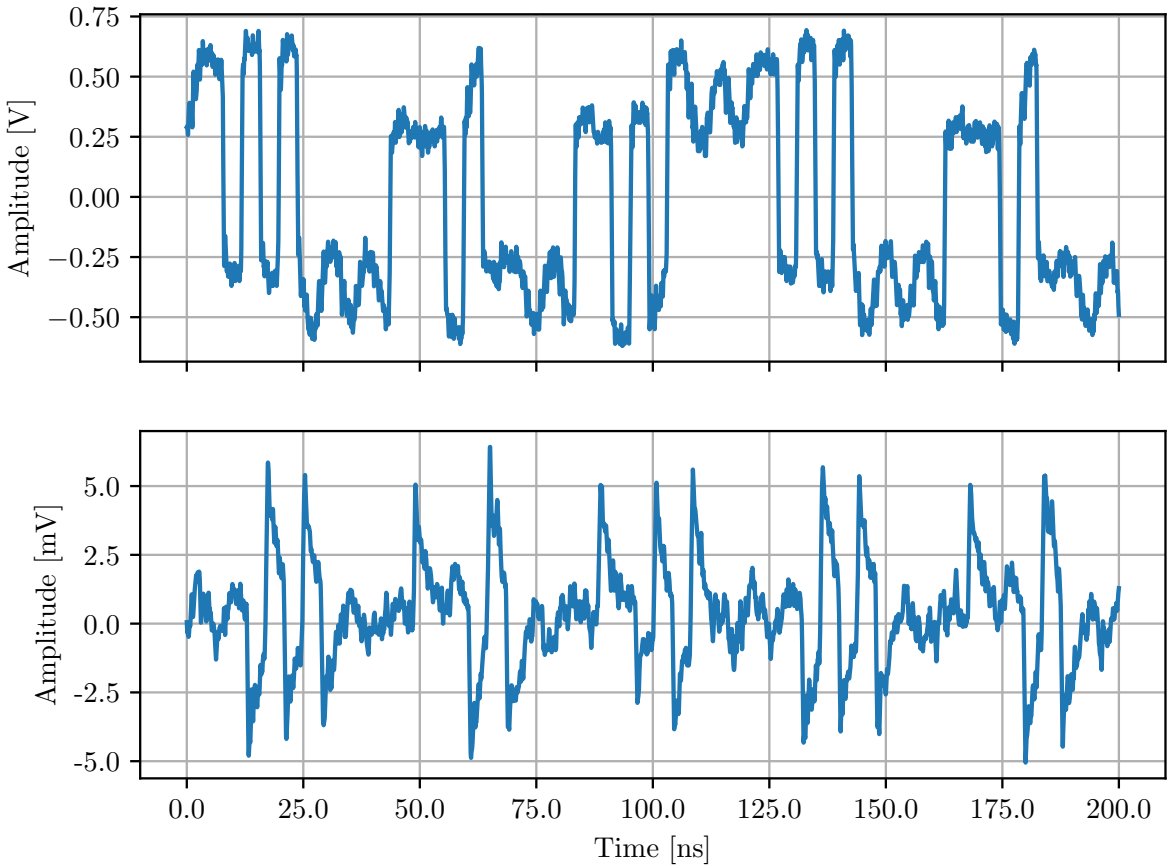


Figure 5.1. A section of a brightness 14 grey video signal transmitting at $f_\mathrm{b} = 252$ MHz captured using a Tektronix TDS7254B oscilloscope (top, recorded on the red channel from Pins 1 and 3 using a Tektronix P7330 differential probe at $f_\mathrm{s} = 10$ GHz) demonstrates a distinctive waveform characteristic. This creates emanations for a TMDS character specific to a device (bottom, recorded with a near field probe positioned over the connector). Until now, bandpassed recordings of emanations have been used to produce an estimate of brightness. A wider bandwidth will allow estimation of bit transitions.

5.1 Frequency band selection

Square waves have a (theoretically) infinite bandwidth, but a bandpass-filtered recording is a practical necessity. However, with a wide enough band $B' > f_\mathrm{b}$ we will recover enough information to discern bit data. We must also record enough bandwidth to prevent the impulse response of the bandpass filter causing intersymbol interference. We will assume front-end D is only capable of recording a bandwidth $\breve{B} \ll B'$ at sample rate $\breve{f}_\mathrm{s}$, but is able to tune to any frequency within $f_\mathrm{c} \pm 0.5 \cdot B'$. We will need to make

$$C = \left\lceil \frac{B' - \Upsilon}{\breve{B} - \Upsilon} \right\rceil \tag{5.1}$$

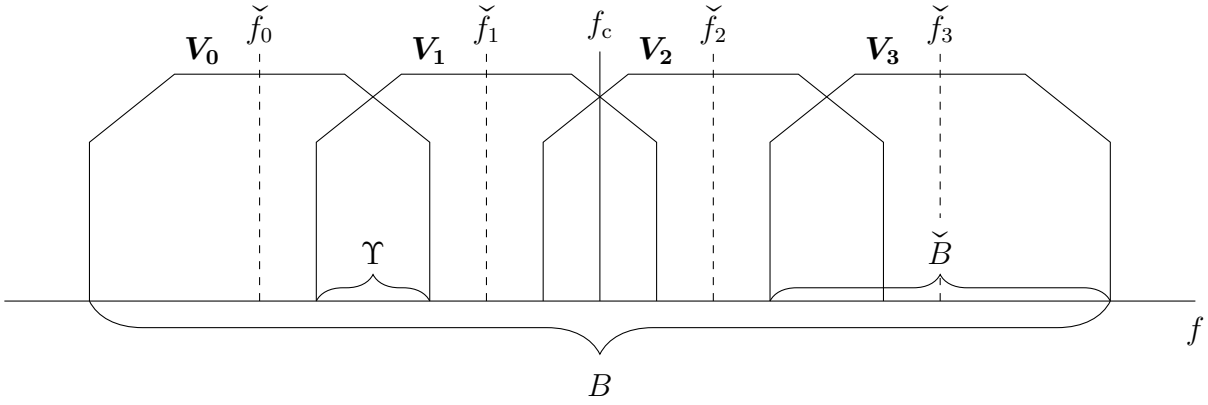


Figure 5.2. We make a series of recordings $\boldsymbol{v}_b$ centred at $\breve{f}_b$ with bandwidth $\breve{B}$ to cover the frequency range B. The bands overlap by Υ in the frequency domain to provide redundancy used to combine the narrowbands into a single estimate of the wideband frequency information contained within B.

narrowband[1] measurements, overlapping by Υ (Figure 5.2), to cover B', from which we will be able to estimate the frequency content of a

$$B = C \cdot \left(\breve{B} - \Upsilon\right) + \Upsilon \tag{5.2}$$

wide frequency band. To centre the frequency band around f_c, the constituent narrowbands have centre frequencies:

$$\breve{\boldsymbol{f}} = \left\{ \breve{f}_b : b \in \mathbb{Z}_C \right\} , \tag{5.3}$$

where

$$\breve{f}_b = f_\mathrm{c} + b \cdot \left(\breve{B} - \Upsilon\right) + 0.5 \cdot \left(\breve{B} - B\right) , \tag{5.4}$$

which we use to make the narrowband recordings

$$\boldsymbol{v}_b = \mathrm{D}_{\breve{f}_b}\{\boldsymbol{v}_\mathbf{R}\} . \tag{5.5}$$

5.2 Stitching

Let us define signal $\boldsymbol{y}$ of bandwidth B into which we will *stitch* the narrowbands to create the wideband estimate. The sample rate of $\boldsymbol{y}$ is $f_\mathrm{s} = 2 \cdot B$, twice the Nyquist frequency, to allow representation of $\boldsymbol{y}$ in its analytic form so we can later apply real-valued filters (§5.2.3). Initially it is an empty signal:

$$y_0[n] = 0 , \tag{5.6}$$

[1] I loosely use the term *narrowband* to describe a signal with a bandwidth significantly smaller than the desired bandwidth B'.

where the subscript indicates the number of narrowbands stitched in. We now iterate over the narrowbands in ascending order of index b. For each narrowband we can optionally apply periodic averaging to increase the SNR (Chapter 4), and use the SDR timestamps with the estimate for f_p to provide a crude alignment of the video content prior to fine tuning in §5.2.2.

5.2.1 Upsampling

We must resample $\boldsymbol{v}_b$ by

$$R = \frac{\breve{f}_\mathrm{s}}{f_\mathrm{s}} \tag{5.7}$$

to match the sample rate of $\boldsymbol{y}$. We upsample by a factor

$$L = \frac{\mathrm{LCM}\left(\breve{f}_\mathrm{s}, f_\mathrm{s}\right)}{\breve{f}_\mathrm{s}} \tag{5.8}$$

by zero stuffing the signal:

$$\hat{v}_b[n] = \begin{cases} v_b\left[\frac{n}{L}\right] & n \bmod L = 0 \\ 0 & \text{otherwise} \end{cases} . \tag{5.9}$$

The spectral aliases are then filtered out:

$$\breve{v}_b[n] = \mathrm{FIR}_{0.5\breve{f}_\mathrm{s}}\left\{\hat{\boldsymbol{v}}_b\right\}[n] , \tag{5.10}$$

before downsampling by

$$M = \frac{\mathrm{LCM}\left(\breve{f}_\mathrm{s}, f_\mathrm{s}\right)}{f_\mathrm{s}} , \tag{5.11}$$

resulting in

$$w_b[n] = \breve{v}_b[nM] , \tag{5.12}$$

which matches the sample rate, and hence bandwidth capacity, of $\boldsymbol{y}$.

5.2.2 Frequency, phase, and delay alignment

We frequency convert $\boldsymbol{w}_b$ relative to f_c:

$$\mathring{w}_b[n] = w_b[n] \cdot \mathrm{e}^{2\pi\mathrm{j}(\breve{f}_b - f_\mathrm{c} + 0.5B)n/f_\mathrm{s}} , \tag{5.13}$$

aligning the frequency information such that f_c is located at $0.5B$. This positions the information in the positive frequency domain in preparation for real-valued filtering. As we are using a single SDR front-end to make consecutive measurements, the quasiperiodic component within each narrowband will exhibit a time shift relative to the others. The bands will also exhibit a random phase rotation due to changes in the offset of the front-end's LO[2]. For the first band ($b = 0$), no further alignment or filtering is necessary: both

[2]The UHD only guarantees the front-end's LO phase offset will remain constant between recordings for the same centre frequency; once retuned the offset will change by some random amount.

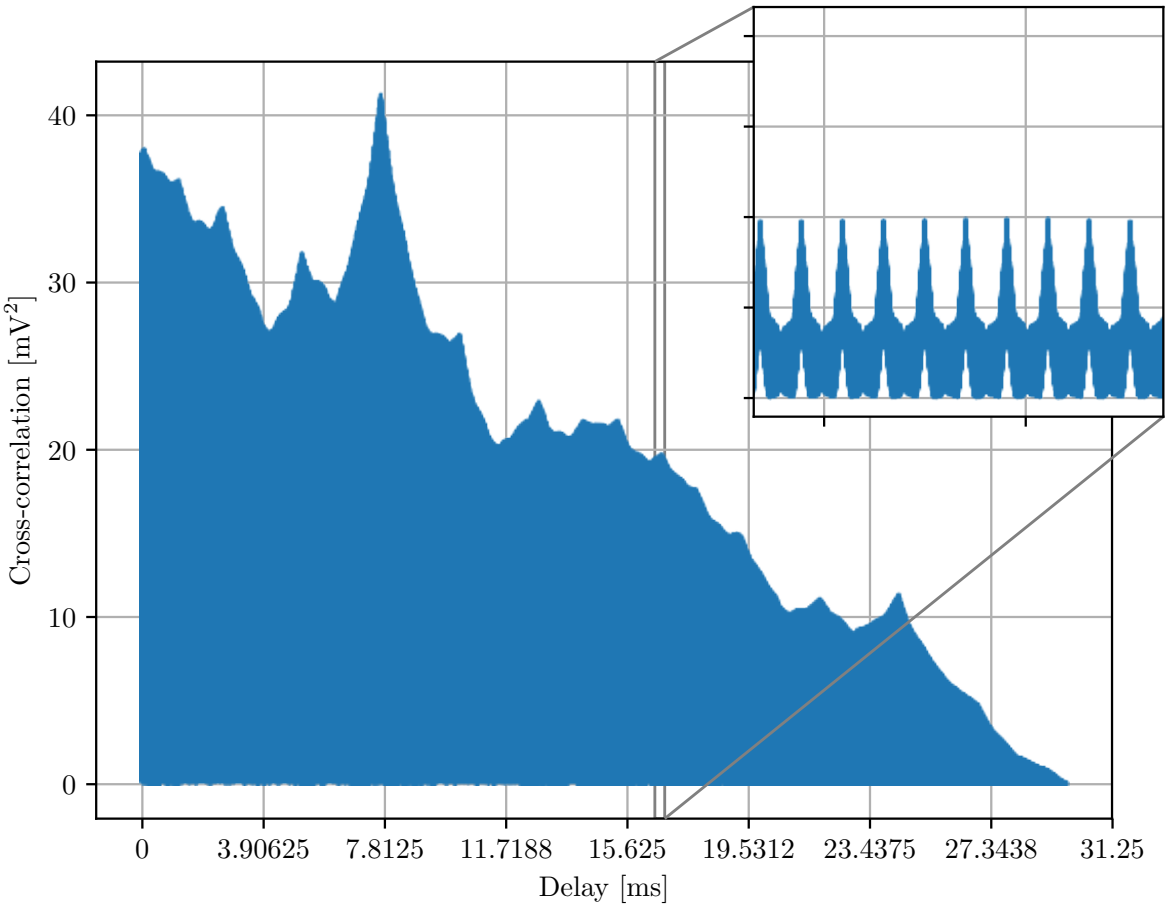


Figure 5.3. The magnitude plot of the cross-correlation between two overlapping narrowbands containing an emanated video signal produces peaks at the point of alignment. Here, peaks occur every $f_{\mathrm{v}}{}^{-1}$ as the frame data aligns. As with the autocorrelation, smaller peaks occur every $f_{\mathrm{h}}{}^{-1}$. We use this information to identify the sample delay between the video content in the two bands and align the signals in the time domain. The angle of cross-correlation at the point of alignment provides the phase offset between the two narrowbands and is used to correct the phasor offset introduced by the SDR front-end between recordings.

the phase and delay offsets are implicitly zero and we can skip to stitching in the band (§5.2.4). For $b > 0$, we turn to the cross-correlation to estimate the alignment parameters. Similar to autocorrelation (§3.4.1), cross-correlation computes the dot product between $\mathring{\boldsymbol{w}}_b$ of length P and $\boldsymbol{y}_b$ of length N for delay τ:

$$X\{\boldsymbol{y}_b, \mathring{\boldsymbol{w}}_b\}[\tau] = \sum_{n=0}^{N-1} y_b[n] \cdot \mathring{w}_b[n+\tau]^* \,. \tag{5.14}$$

Peaks in the cross-correlation magnitude indicate delays at which the two signals appear similar. Cross-correlation is the optimal method for finding a signal embedded within AWGN [109], which we determined to be an appropriate model for the noise in §4.6 after applying sufficient complex averaging. Like with the autocorrelation, we can use the Wiener-Khinchin theorem to reduce the computational complexity. We zero pad $\mathring{\boldsymbol{w}}_b$ and

$\boldsymbol{y}_b$ to twice the next highest power of two of their combined lengths:

$$y'_b[n] = \begin{cases} y_b[n] & 0 \leq n < N \\ 0 & N \leq n < 2^{\lfloor \log_2(N+P-1) \rfloor + 1} \end{cases}, \tag{5.15}$$

$$\mathring{w}'_b[n] = \begin{cases} \mathring{w}_b[n] & 0 \leq n < P \\ 0 & P \leq n < 2^{\lfloor \log_2(N+P-1) \rfloor + 1} \end{cases}, \tag{5.16}$$

to ensure convolution without wrap-around, and compute the cross-correlation:

$$X\{\boldsymbol{y}'_b, \mathring{\boldsymbol{w}}'_b\}[\tau] = \mathcal{F}^{-1}\{\mathcal{F}\{\mathring{\boldsymbol{w}}'_b\} \cdot \mathcal{F}\{\boldsymbol{y}'_b\}^*\}[\tau]\,. \tag{5.17}$$

Using this formulation, we can see the cross-correlation will only operate on the overlapped frequencies; those outside the overlapped band are effectively multiplied by zero. This means we do not have to ensure the same spectral content via bandpass filtering the inputs beforehand. The largest peaks in $\{|X\{\boldsymbol{y}'_b, \mathring{\boldsymbol{w}}'_b\}[\tau]|\}$ (Figure 5.3) appear every ${f_\mathrm{v}}^{-1}$ with smaller peaks every ${f_\mathrm{h}}^{-1}$, similar to the effect we saw when using the autocorrelation (Figure 3.5), indicating the location at which $\mathring{\boldsymbol{w}}_b$ aligns with $\boldsymbol{y}_b$; $\tau = 0$ is now potentially a valid peak, so we consider it as a candidate for selection. We select the largest peak

$$\hat{\tau}_b = \underset{\tau \in \mathbb{Z}_{\lceil \frac{f_\mathrm{s}}{f_\mathrm{v}} \rceil}}{\arg\max} |X\{\boldsymbol{y}'_b, \mathring{\boldsymbol{w}}'_b\}[\tau]| \tag{5.18}$$

as the estimate for the delay between $\mathring{\boldsymbol{w}}_b$ and $\boldsymbol{y}_b$. The complex multiplication performed by the cross-correlation will also unrotate the samples. This results in the relative phase offset between the components in the overlapped regions of $\mathring{\boldsymbol{w}}_b$ and $\boldsymbol{y}_b$ being expressed in the complex output. Thus we take the angle of the cross-correlation at $\hat{\tau}_b$:

$$\phi_b = \angle X\{\boldsymbol{y}'_b, \mathring{\boldsymbol{w}}'_b\}[\hat{\tau}_b] \tag{5.19}$$

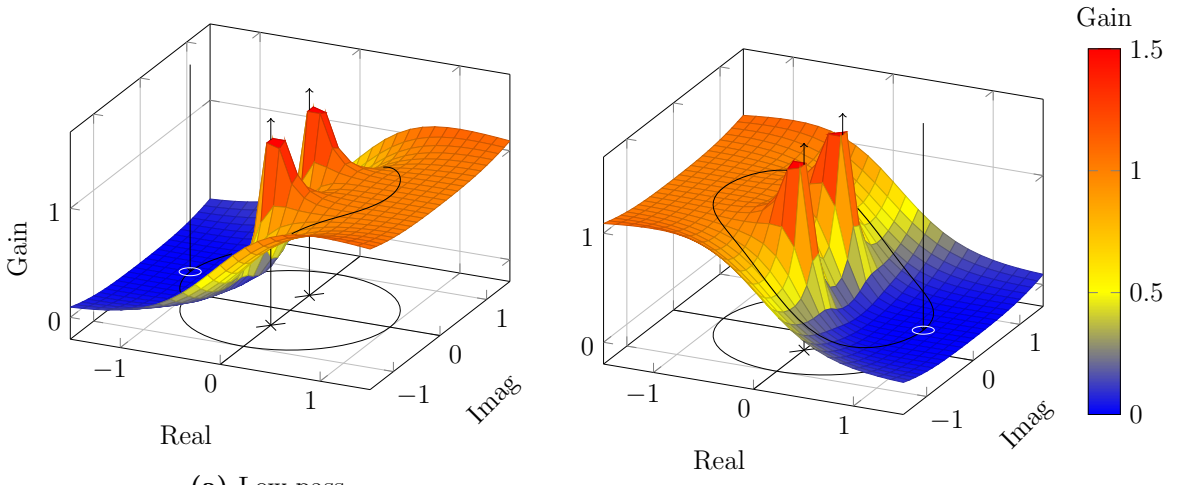
as the estimate for the phase offset between $\boldsymbol{y}_b$ and $\mathring{\boldsymbol{w}}_b$. We can now introduce a delay and phase correction term:

$$x_b[n] = \mathring{w}_b[n - \hat{\tau}_b] \cdot \mathrm{e}^{-\mathrm{j}\phi_b} \tag{5.20}$$

to produce $\boldsymbol{x}_b$, which is aligned with $\boldsymbol{y}_b$. (The video content in the stitched output will exhibit the same translational offset as $\boldsymbol{v}_0$. This can be removed by cross-correlating $\boldsymbol{v}_0$ with:

$$m[i] = \mathbb{1}_{(i \bmod w_\mathrm{T}) \geq w_\mathrm{D} \vee (i \bmod w_\mathrm{T} h_\mathrm{T}) \geq w_\mathrm{T} h_\mathrm{D}} \quad \forall i \in \mathbb{Z}_{w_\mathrm{T} h_\mathrm{T}}\,, \tag{5.21}$$

which models the location of the generally brighter blanking interval (Figure 2.13), where the first sample of the first line of the content area is positioned at $m[0]$. Selecting the largest peak in $\{|X\{\mathring{\boldsymbol{w}}'_0, \boldsymbol{m}'\}[\tau]|\}$ will estimate the translational offset which can then be removed using Equation 5.20.)



(a) Low-pass

(b) High-pass

Figure 5.4. The z-domain plots of the transfer functions of $H_{\mathrm{L}}(z)\cdot H_{\mathrm{L}}(z^{-1})$ and $H_{\mathrm{H}}(z)\cdot H_{\mathrm{H}}(z^{-1})$. Each filter has four poles at the roots of the denominator polynomial ($z = \pm\mathrm{j}\sqrt{3 \pm 2\sqrt{2}}$): identical for the high-pass and low-pass filters. They are symmetric around the real axis making the filters real valued. Two poles are within the unit circle, and two outside (not pictured). As the filters are non-causal, this property does not cause instability. The zeros—the roots of the numerator polynomial—change position to create the high-pass or low-pass variant. The gain applied by the filter is given by the value of the function at frequency z on the unit circle, outlined in black. Summation of the responses is uniformly one on the unit circle.

5.2.3 Crossover filtering

Before we can add $\boldsymbol{x}_b$ to $\boldsymbol{y}_b$ we must attenuate the frequency information in the overlapping regions to prevent creating a *bump* in the frequency magnitudes. A Linkwitz-Riley (L-R) filter [110] consists of a complementary biquadratic high-pass and low-pass Butterworth filter whose magnitude responses sum to unity, giving a flat magnitude response across the spectrum. It was originally developed for audio crossover applications to separate an audio signal into high and low frequency bands before being fed into a speaker suited to that frequency range; the flat magnitude response prevents the audio sounding distorted to a listener. We will use the L-R filter for the opposite purpose: filtering already separated frequency bands to allow us to recombine them into a single band with minimal magnitude distortion. The high-pass and low-pass filters have the same 6 dB cut-off frequency. We select the centre point of the overlapping region:

$$f_{\mathrm{x}} = \breve{f}_b - f_{\mathrm{c}} + 0.5 \cdot \left(B + \Upsilon - \breve{B}\right) \tag{5.22}$$

as the cut-off point. By way of example I will demonstrate a $W = 4$ order L-R filter constructed from two second-order sections. The transfer function of an analogue, continuous-time Butterworth filter can be described in terms of frequency s by its Laplace transform:

$$H(s) = \frac{\omega_{\mathrm{x}}}{s^2 + s\sqrt{2}\omega_{\mathrm{x}} + \omega_{\mathrm{x}}^2}\,, \tag{5.23}$$

the continuous-time counterpart to the z-transform, where

$$\omega_{\mathrm{x}} = f_{\mathrm{x}} \cdot 2\pi \text{ rad} \tag{5.24}$$

is the angular 3 dB cut-off frequency and $H(\mathrm{j}\omega)$ is the Fourier transform of the filter's impulse response. To convert to a digital filter we use the bilinear transform: first, the Fourier transform of the filter's impulse response, found on the infinite imaginary axis of the complex s-plane, is mapped to the finite unit circle on the z-plane using the cut-off frequency

$$\widetilde{\omega}_{\mathrm{x}} = \tan \frac{\omega_{\mathrm{x}}}{2 \cdot f_{\mathrm{s}}}\,, \tag{5.25}$$

a process known as *pre-warping*. For simplicity I will demonstrate cut-off frequency $\omega_{\mathrm{x}} = 0.5\pi \cdot f_{\mathrm{s}}$ (i.e. $f_{\mathrm{x}} = f_{\mathrm{s}}/4$) which, using Equation 5.25, evaluates to $\widetilde{\omega}_{\mathrm{x}} = 1$. We plug $\widetilde{\omega}_{\mathrm{x}}$ into $H(s)$ and evaluate for

$$s = \frac{1 - z^{-1}}{1 + z^{-1}} \tag{5.26}$$

to obtain the z-transform of the low-pass digital Butterworth's impulse response:

$$\begin{aligned} H_{\mathrm{L}}(z) &= H(s)\big|_{s=\frac{1-z^{-1}}{1+z^{-1}}} \\ &= \frac{(1+z^{-1})^2}{(2+\sqrt{2}) + (2-\sqrt{2})z^{-2}}\,, \end{aligned} \tag{5.27}$$

which is real-valued on the unit circle. The L-R filter applies $H_{\mathrm{L}}(z)$ twice as two cascaded sections, squaring the impulse response and producing 6 dB of attenuation at the cut-off frequency. IIR filters have a non-linear phase response which results in an offset between the outputs of the high-pass and low-pass filters. We can utilise the time reversal property of the z-transform:

$$x[n] \xrightarrow{z} X(z) \implies x[-n] \xrightarrow{z} X(z^{-1})\,, \tag{5.28}$$

by time reversing the signal before the second stage of the cascaded filter, producing the autocorrelation of the impulse response of H_{L}:

$$H_{\mathrm{L}}(z) \cdot H_{\mathrm{L}}(z^{-1}) = \frac{z^2 + 4z + 6 + 4z^{-1} + z^{-2}}{2z^2 + 12 + 2z^{-2}}\,. \tag{5.29}$$

The roots of the numerator and denominator—zeros and poles respectively—produce an even function symmetric around DC, seen in the plot of the z-plane in Figure 5.4. The result is a zero-phase response on the unit circle (the filter's response):

$$\angle\left(H_{\mathrm{L}}(\mathrm{e}^{\mathrm{j}\omega}) \cdot H_{\mathrm{L}}(\mathrm{e}^{-\mathrm{j}\omega})\right) = 0\,. \tag{5.30}$$

The positive exponents mean the zero-phase property comes at the cost of causality. As processing is performed offline, however, this is acceptable. We filter $\boldsymbol{y}_b$:

$$\grave{Y}_b(z) = Y_b(z) \cdot H_{\mathrm{L}}(z) \cdot H_{\mathrm{L}}(z^{-1}) \tag{5.31}$$

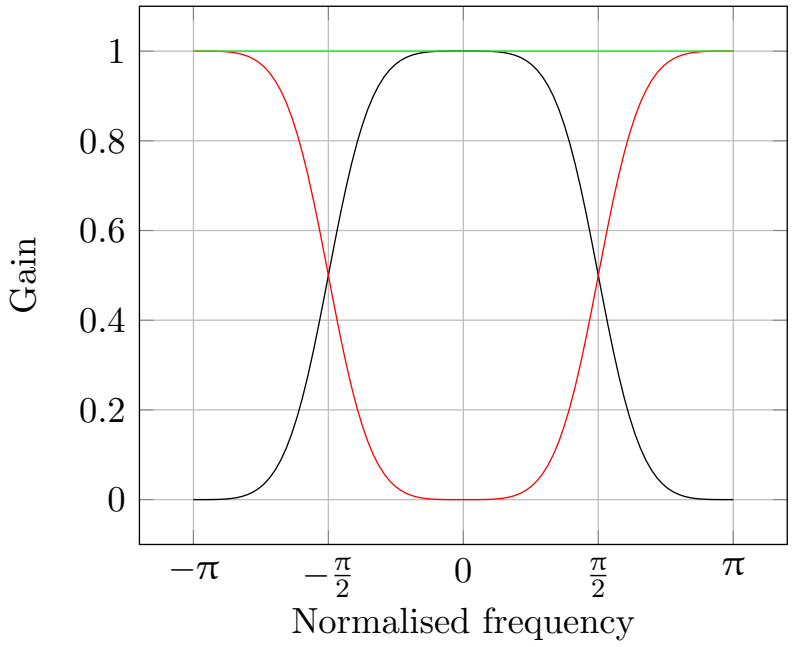


Figure 5.5. The low-pass $H_{\mathrm{L}}(z) \cdot H_{\mathrm{L}}(z^{-1})$ (black) and high-pass $H_{\mathrm{H}}(z) \cdot H_{\mathrm{H}}(z^{-1})$ (red) filters constituting an L-R crossover filter sum to give a flat magnitude response with unity gain (green). This important characteristic allows us to filter two overlapping bands and combine the result via summation without introducing additional amplitude distortion. Each filter is implemented using two cascaded stages, resulting in 6 dB of attenuation at the cut-off frequency (here shown as $\frac{\pi}{2}$). We use this structure to create zero-phase filters by time reversing the signal between the two filtering stages.

to obtain $\grave{\boldsymbol{y}}_b$. We repeat the process to design the complementary high-pass filter:

$$\begin{aligned} H_{\mathrm{H}}(z) &= H(s)\big|_{s=\frac{1+z^{-1}}{1-z^{-1}}} \\ &= \frac{(1-z^{-1})^2}{(2+\sqrt{2})+(2-\sqrt{2})z^{-2}}\,, \end{aligned} \tag{5.32}$$

which produces the zero-phase cascaded filter:

$$H_{\mathrm{H}}(z) \cdot H_{\mathrm{H}}(z^{-1}) = \frac{z^2 - 4z + 6 - 4z^{-1} + z^{-2}}{2z^2 + 12 + 2z^{-2}}\,, \tag{5.33}$$

which we then use to high-pass filter $\boldsymbol{x}_b$:

$$\acute{X}_b(z) = X_b(z) \cdot H_{\mathrm{H}}(z) \cdot H_{\mathrm{H}}(z^{-1})\,, \tag{5.34}$$

to obtain $\acute{\boldsymbol{x}}_b$.

5.2.4 Summation

The sum of the responses is

$$\begin{aligned} H_{\mathrm{L}}(z) \cdot H_{\mathrm{L}}(z^{-1}) + H_{\mathrm{H}}(z) \cdot H_{\mathrm{H}}(z^{-1}) &= \frac{2z^2 + 12 + 2z^{-2}}{2z^2 + 12 + 2z^{-2}} \\ &= 1\,, \end{aligned} \tag{5.35}$$

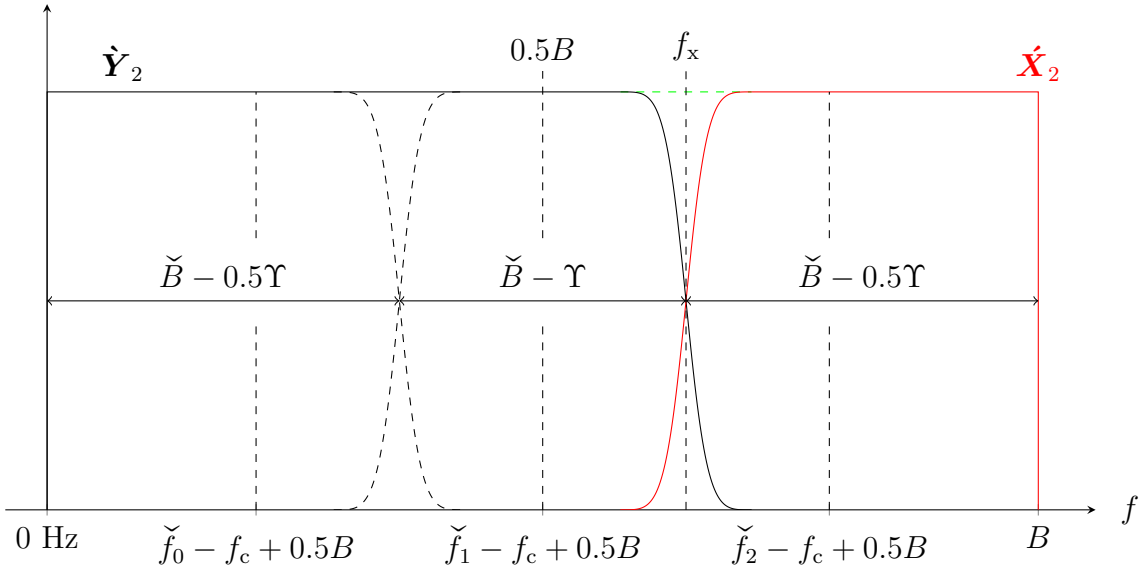


Figure 5.6. The $C = 3$ narrowbands are combined in the positive frequency domain, enabling real-valued filtering. The narrowband $\acute{\boldsymbol{x}}_b$ (red) is positioned relative to $0.5B$, which equates to the centre frequency of the wideband. Having been positioned in the frequency domain, the L-R filtered regions of the narrowband and the partially completed wideband $\grave{\boldsymbol{y}}_b$ (black) overlap. Having corrected for phase and delay offsets summation restores the attenuated frequency information in the overlapping regions (green). Attenuation outside of the overlapping region is not restored, therefore the filter order should be high enough to produce a small transition band, preventing significant attenuation outside of the overlapped regions.

a unit gain factor independent of z. When plotting the magnitude response of $H_{\mathrm{L}}(z) \cdot H_{\mathrm{L}}(z^{-1})$ and $H_{\mathrm{H}}(z) \cdot H_{\mathrm{H}}(z^{-1})$, and their sum in Figure 5.5, we see they form a complementing pair which sum to unity across the frequency response. This property means we can combine $\acute{\boldsymbol{x}}_b$ and $\grave{\boldsymbol{y}}_b$ without introducing distortion in the overlap band via addition:

$$y_{b+1}[n] = \grave{y}_b[n] + \acute{x}_b[n]\,. \tag{5.36}$$

We repeat the stitching process until all bands have been stitched together producing $\boldsymbol{y}_{|\breve{\boldsymbol{f}}|}$ (Figure 5.6), which is then down converted to baseband:

$$y[n] = y_{|\breve{\boldsymbol{f}}|}[n] \cdot \mathrm{e}^{-\pi \mathrm{j} B n / f_{\mathrm{s}}} \tag{5.37}$$

to obtain the estimate of the down converted wideband frequency information. The procedure is summarised in Algorithm 4.

The main parameter to consider when stitching is the size of the overlapping region Υ. A larger Υ provides more information which can be used to properly align the narrowbands, and permits a more gradual roll-off for the L-R filters, allowing them to be implemented using a lower filter order. A L-R filter has $W \cdot (20\ \mathrm{dB})/\mathrm{decade}$ roll-off. If the roll-off is too gentle there will be more attenuation in the non-overlapping regions, resulting in distortion in the stitched result, shown in Figure 5.7. Increasing Υ reduces the roll-off

Algorithm 4 Stitching of overlapping narrowbands into a wideband signal.

procedure STITCH(B, $\boldsymbol{v}_0$, $\boldsymbol{v}_1$, …, $\boldsymbol{v}_{C-1}$)
  $y[n] \leftarrow 0$ ▷ Initialise output
  $f_\mathrm{s} = 2 \cdot B$ ▷ Set output sample rate
  for $b \in (0, 1, \ldots, C-1)$ **do** ▷ For each narrowband
    $\boldsymbol{w}_b \leftarrow \mathrm{resample}_{f_\mathrm{s}}(\boldsymbol{v}_b)$ ▷ Upsample to output rate
    $w_b[n] \leftarrow w_b[n] \cdot \mathrm{e}^{2\pi\mathrm{j}(\breve{f}_b - f_\mathrm{c} + 0.5 \cdot B)n/f_\mathrm{s}}$ ▷ Frequency alignment
    $\boldsymbol{x} \leftarrow \mathrm{xcorr}(\boldsymbol{y}, \boldsymbol{w}_b)$ ▷ Compute cross-correlation
    $\hat{\tau} \leftarrow \arg\max\{|x[\tau]|\}$ ▷ Locate delay peak
    $w_b[n] \leftarrow w_b[n - \hat{\tau}] \cdot \mathrm{e}^{-\mathrm{j}\angle x[\hat{\tau}]}$ ▷ Delay and phase correction
    $f_\mathrm{x} \leftarrow \breve{f}_b - f_\mathrm{c} + 0.5 \cdot (B + \Upsilon - \breve{B})$ ▷ Midpoint of overlap region
    $\boldsymbol{w}_b \leftarrow \mathrm{LR}_{\mathrm{H},f_\mathrm{x}}(\boldsymbol{w}_b)$ ▷ Linkwitz-Riley filtering
    $\boldsymbol{y} \leftarrow \mathrm{LR}_{\mathrm{L},f_\mathrm{x}}(\boldsymbol{y})$
    $y[n] \leftarrow y[n] + w_b[n]$ ▷ Summation
  end for
  $y[n] \leftarrow y[n] \cdot \mathrm{e}^{-\pi\mathrm{j}Bn/f_\mathrm{s}}$ ▷ Down convert to baseband
  return $\boldsymbol{y}$ ▷ Return stitched result
end procedure

contraint, but means more bands are required to cover the same frequency range; errors in phase alignment will accumulate increasing the noise in $\boldsymbol{y}$. I use $\Upsilon = 0.5\breve{B}$ with $W = 8$ L-R filters in the experiments with the Stitch data set.

Let us observe the effect of stitching. In Figure 5.8a I raster an 800⊗525@25.2 MHz video signal eavesdropped with a single $\breve{B} = 1$ MHz band (Stitch data set). The image is heavily blurred and only large features can be discerned, even at such a high SNR. Now let us stitch together 9 bands of $\breve{B} = 1$ MHz with $\Upsilon = 500$ kHz to estimate $B = 5$ MHz. A much clearer raster is recovered (Figure 5.8b), similar to what we can obtain with a single band of $\breve{B} = 5$ MHz. New features not previously discernible, such as the cross markings and rule marks, have become visible and readable text is recoverable.

5.3 Character recovery

With the ability to record bandwidths of several hundred megahertz, we are in a position to attempt recovery of TMDS character information. For an 800⊗525@25.2 MHz video signal we stitch three narrowbands of $\breve{B} = 160$ MHz with $\Upsilon = 80$ MHz to estimate a $B = 320$ MHz signal $\boldsymbol{y}$, centred at $f_\mathrm{c} = f_\mathrm{b} \cdot h = 756$ MHz where $h = 3$. I plot a section of $\boldsymbol{y}$ in Figure 5.9. We can observe in the amplitude plot transitions between bit values as impulses; the sign information indicating a rising or falling edge is lost. I found that the recovery of pulse sign information is non-trivial. Emanated information varies dras-

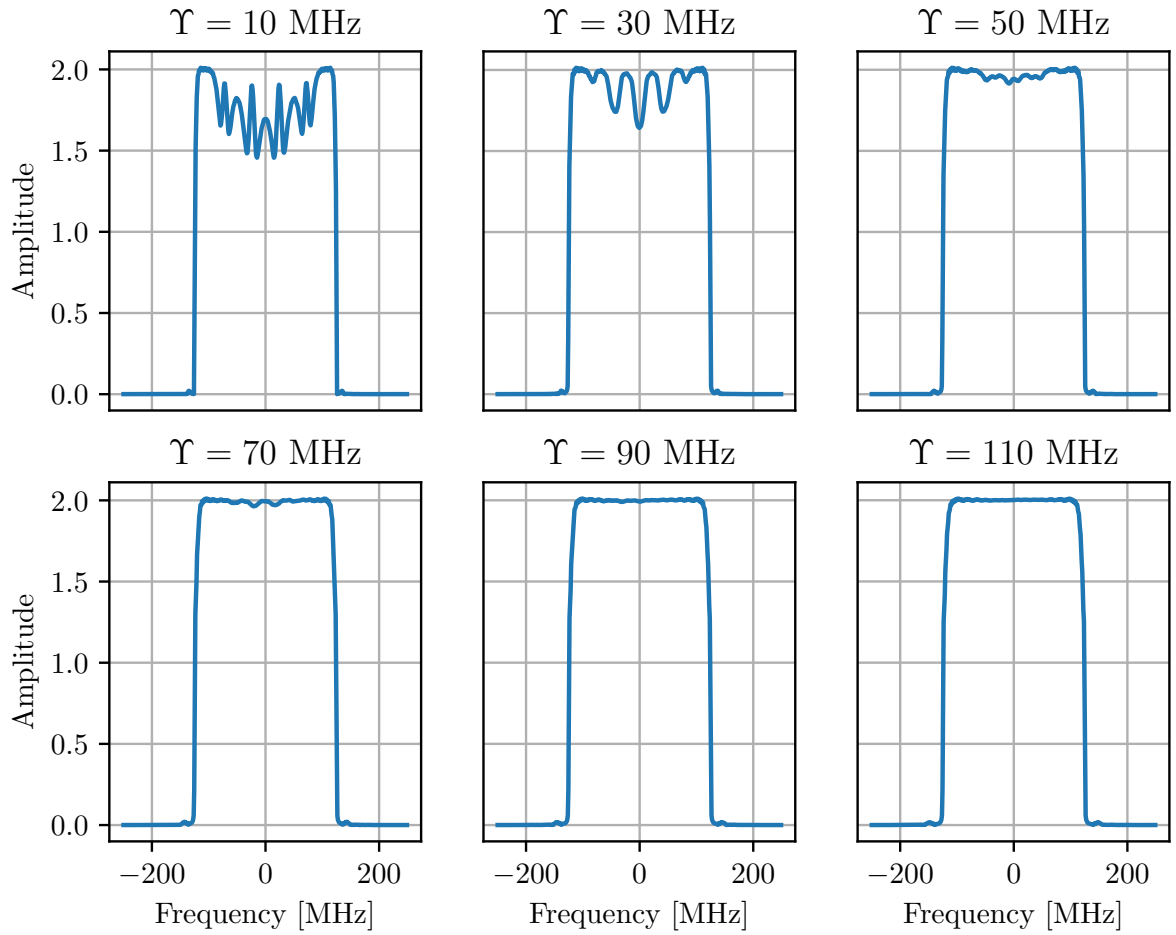
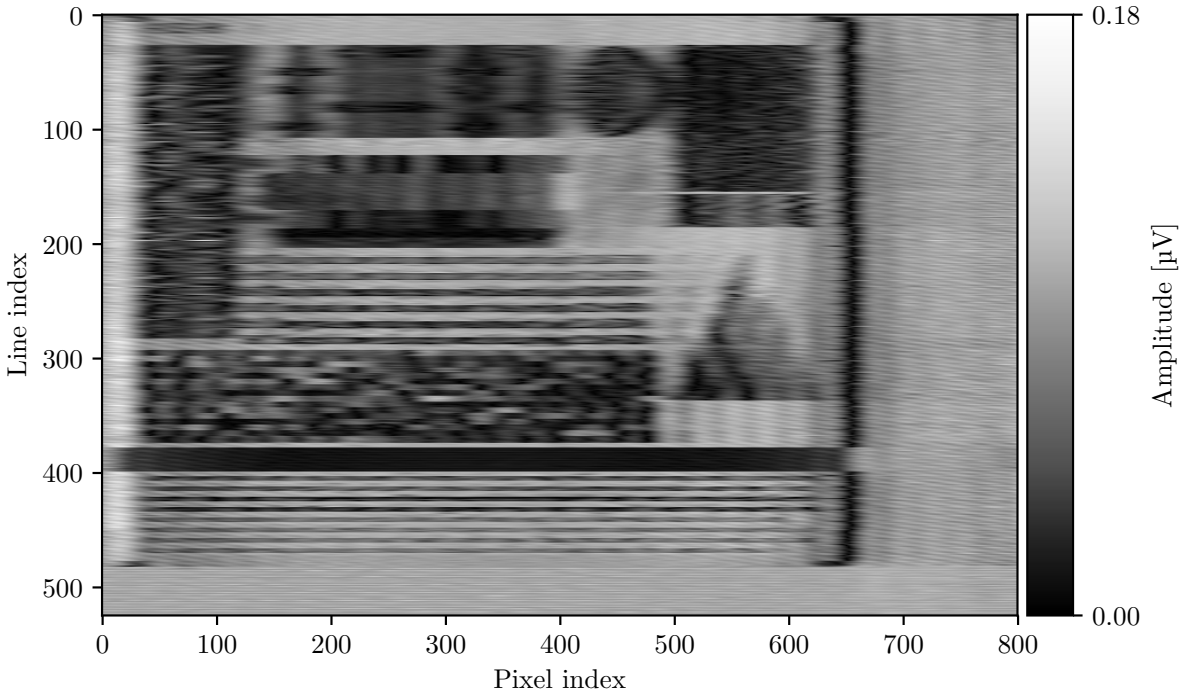
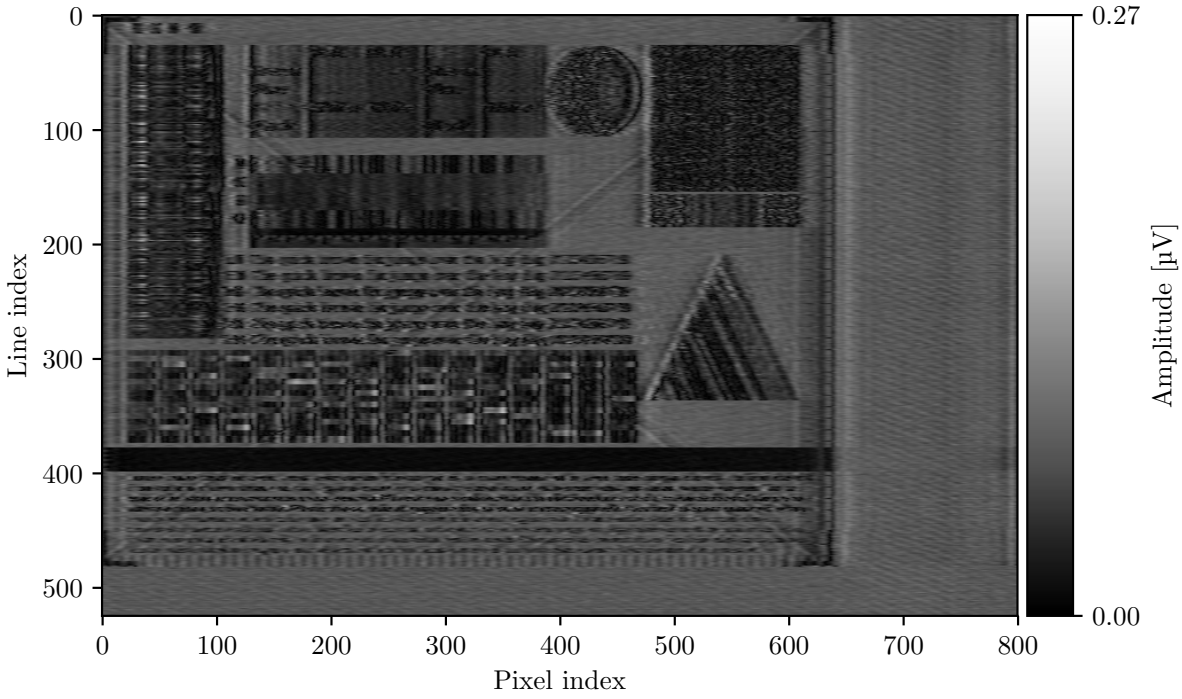


Figure 5.7. Distortion in the stitched result can occur when there is insufficient overlap between the narrowbands because significant attenuation by the L-R filter will occur outside the overlap band. Here I demonstrate the effect when attempting to reconstruct $B = 160$ MHz of a simulated impulse from $C = 5$ narrowbands, varying $\breve{B}$ to adjust the level of overlap Υ, using a $W = 8$ order (two 4^{th} order cascaded sections) L-R filter providing 160 dB/decade roll-off. One solution is to use higher-order filters to increase the filter roll-off, minimising the attenuation outside of the overlap band. However, this can risk filter instability. An alternative is to increase the level of overlap as demonstrated. This comes at a cost of requiring more bands to cover the same frequency range which can cause the residual phase error between aligned bands to accumulate and negatively affect the result.

tically between devices due to varying field strengths and polarities from the individual transmission lines, creating unique patterns. Additionally, the received emanations are a mix of four main processes: the three colour channels and the clock signal. Thus, one cannot directly observe bit transition information for a specific channel. Instead, we will attempt to model character information for a specific device (a model for one device is not necessarily valid for another). For the following experiments, I limit recovery efforts to five colours: red, green, blue, white, and black using an enlarged section of the colour boxes in Figure B.1, where 0 and 255 are the low and high brightness levels respectively.



(a) Single band of $\breve{B} = 1$ MHz.



(b) $C = 9$ bands of $\breve{B} = 1$ MHz stitched together to form an estimate of $B = 5$ MHz.

Figure 5.8. The raster of a video signal recorded with a low bandwidth (top) exhibits significant blurring, making the features difficult to discern. In this example, I present the narrowband centred over a pixel-clock harmonic, which provided the best SNR of the recorded narrowbands. Stitching together multiple narrowband recordings allows us to estimate the wideband video signal content (bottom). The stitched raster is similar to estimates made with a single wideband recording (Figure 2.12). The raster is no longer blurred and new features, such as the cross and recognisable text, can be observed.

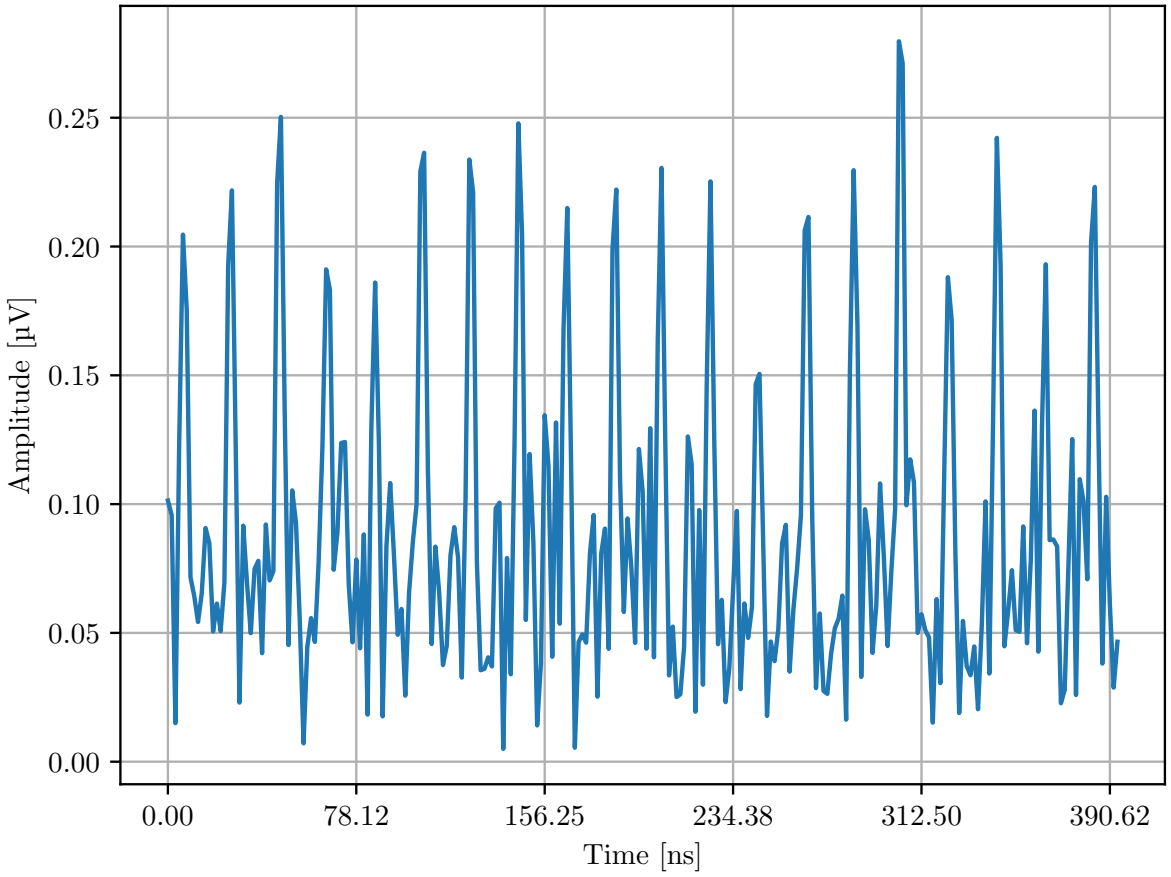


Figure 5.9. The amplitude of the wideband signal $\boldsymbol{y}$ temporarily increases as a transition between bit values occurs on the wire, causing an increase in the emission of electromagnetic radiation. Here, a brightness 16 greyscale value produces a clear peak every five bits. The emanated pattern varies greatly between devices; other cables I tested produced different patterns, such as one peak every ten bits, despite being the same underlying TMDS character sequence. This is due to the interaction between the magnitude and polarity of the magnetic fields unique to individual transmitting lines.

To estimate the underlying state, we will need to discretise the observations. We generate a binary string $\boldsymbol{d}$, where $d[n] = 1$ indicates a possible bit transition, estimated by performing a peak search for samples with a magnitude greater than a factor $T = 0.3$ of the maximum amplitude in $\boldsymbol{y}$, and at least

$$N_\mathrm{b} = \left\lfloor \frac{f_\mathrm{s}}{f_\mathrm{b}} + 0.5 \right\rfloor \tag{5.38}$$

samples from neighbouring peaks (the duration of a single bit), to obtain

$$d[n] = \mathbb{1}_{\max_{|j-nN_\mathrm{b}|\leq 0.5N_\mathrm{b}} \frac{y[j]}{\max \boldsymbol{y}} > T} \,. \tag{5.39}$$

The TMDS encoders' signal balance counters are reset during the blanking interval. Thus we define a line, delimited by the blanking interval, as an independent message for processing. Having recovered video alignment information in the narrowbands prior

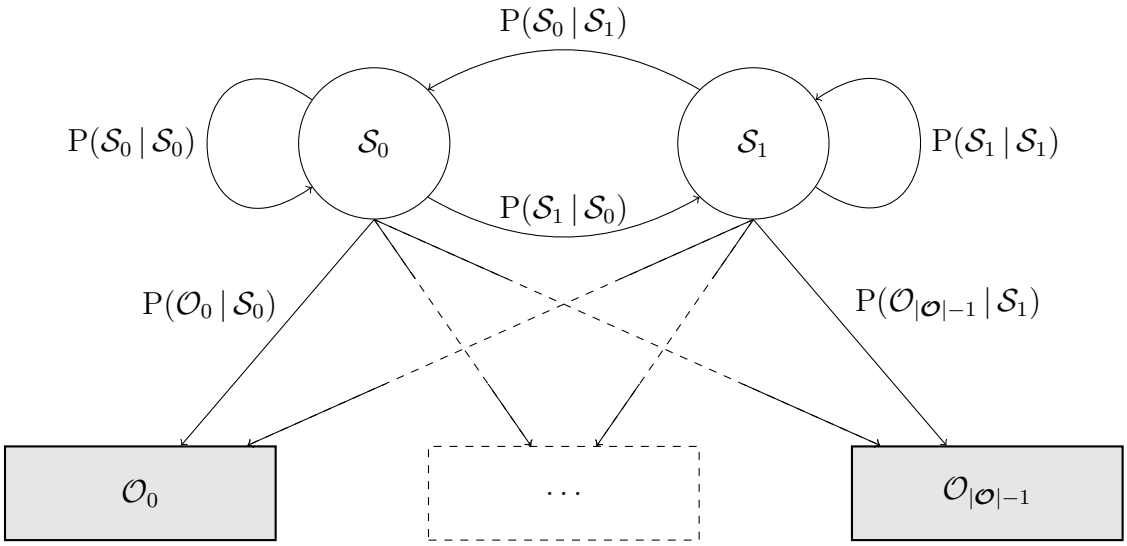


Figure 5.10. A Markov chain models the probabilities of transitioning between states in a discrete system. In our model a state is a tuple of three TMDS characters transmitted concurrently across colour channels. In a hidden Markov model the chain is not directly observable, but we can make a series of indirect observations that depend on the underlying state. Using the Viterbi algorithm we can infer the most likely sequence of states the system took, simultaneously performing the conversion from observations to TMDS characters, and error correction.

to stitching, we will already have an approximate alignment. We perform precise message alignment [111, p.143] by cross-correlating $\boldsymbol{d}$ with

$$g[i] = \begin{cases} 0 & (i+1) \bmod 10 < 3 \\ 1 & \text{otherwise} \end{cases}, \tag{5.40}$$

a template of the derivative of the high transition count blanking characters (Table A.1) of length $N = 10 \cdot (w_\mathrm{T} - w_\mathrm{D})$. Appropriately zero padded to create $\boldsymbol{d'}$ and $\boldsymbol{g'}$, we locate the starting sample of the next full line:

$$\hat{\tau} = \underset{\tau \in \mathbb{Z}_{w_\mathrm{T}}}{\arg\max} \, |X\{\boldsymbol{d'}, \boldsymbol{g'}\}[\tau]| + N \,. \tag{5.41}$$

We discard any partial lines and split the string into individual messages indexed by $m \in \mathbb{N}$

$$d_m[n] = d[m \cdot 10 \cdot w_\mathrm{T} + n + \hat{\tau}] \,, \tag{5.42}$$

where each message $\boldsymbol{d}_m$ is a binary string of $10 \cdot w_\mathrm{T}$ bit transition estimates. For convenience, $\boldsymbol{d}_m$ is converted into a sequence of w_T 10 bit integers:

$$o_m[n] = \sum_{i=0}^{9} d_m[10n + i] \cdot 2^i \,. \tag{5.43}$$

5.3.1 Hidden Markov model

Recovering a valid TMDS character sequence from $\boldsymbol{o}_m$ is non-trivial. In addition to noise in the transmission channel degrading the signal, the RGB channel emanations will be mixed upon reception rendering us unable to observe individual channels. However while we may not have access to all the transmitted signals individually, we can using our knowledge of the TMDS algorithm itself to perform error correction. From an eavesdropper's perspective the transition between pixels is a stochastic process. A Markov chain models a discrete stochastic process as a graph of states $\boldsymbol{\mathcal{S}}$ with a probability of transitioning between them as edge weights. A Markov chain is memoryless: the probability of the next state can only depend on the current one, known as the Markov property. We will construct the Markov chain from states which are tuples of concurrently transmitted RGB TMDS characters:

$$\mathcal{S} = (\mathcal{S}_\mathrm{R}, \mathcal{S}_\mathrm{G}, \mathcal{S}_\mathrm{B}) \ . \tag{5.44}$$

Due to DC balancing, each brightness level in our model can be represented by two TMDS characters, controlled by a signal balance counter in the encoder (§A.1). The counter determines valid transitions between states. Thus, to model the transition probabilities it would be necessary to include the counter for the three encoders in the state tuple. This would necessitate a large state space: to model the combinations of five colours and the signal balance counters of three channels, $|\boldsymbol{\mathcal{S}}| = 2\,773$ states would be required. For the moment let us ignore the signal balance counters; we will account for them when decoding the model (§5.3.2). Without the counter, each RGB colour can be represented by 2^3 combinations of TMDS characters, meaning our Markov chain uses a total of $|\boldsymbol{\mathcal{S}}| = 40$ states to represent the five pixel colours. To model the transition probabilities we turn to our knowledge of the video content: pixels on a line are unlikely to change frequently (§2.1). We define a parameter γ which controls the propensity for pixel change, and model the probability of transitioning from $\mathcal{S}'$ to $\mathcal{S}$:

$$\mathrm{P}(\mathcal{S} \,|\, \mathcal{S}') = \begin{cases} 2^{-3}(1 - \gamma(|\boldsymbol{\mathcal{S}}| - 2^3)) & \mathrm{dec}(\mathcal{S}_k) = \mathrm{dec}(\mathcal{S}'_k) \quad \forall k \in \{\mathrm{R}, \mathrm{G}, \mathrm{B}\} \\ \gamma & \text{otherwise} \end{cases} , \tag{5.45}$$

where $\mathrm{dec}(\cdot)$ returns the corresponding brightness value for a TMDS character (§A.2), such that $\sum_{\mathcal{S} \in \boldsymbol{\mathcal{S}}} \mathrm{P}(\mathcal{S} \,|\, \mathcal{S}') = 1$, so that a higher γ indicates a signal where pixels are more likely to change.

The actual sequence of transmitted TMDS characters is not directly observable. An HMM treats the Markov chain as a hidden process of which we only have a series of indirect observations $\boldsymbol{o}_m$ (Figure 5.10), in our case the estimates of transitions from the mixed emanating channels. From these we must infer the underlying process. Emission probabilities model the likelihood of the HMM being in a particular state given an observation. An observation of a state is a sequence of transitions with a corresponding integer value $\mathcal{O} \in \mathbb{Z}_{2^{10}}$. We profile the target device to estimate the emission probabilities for each

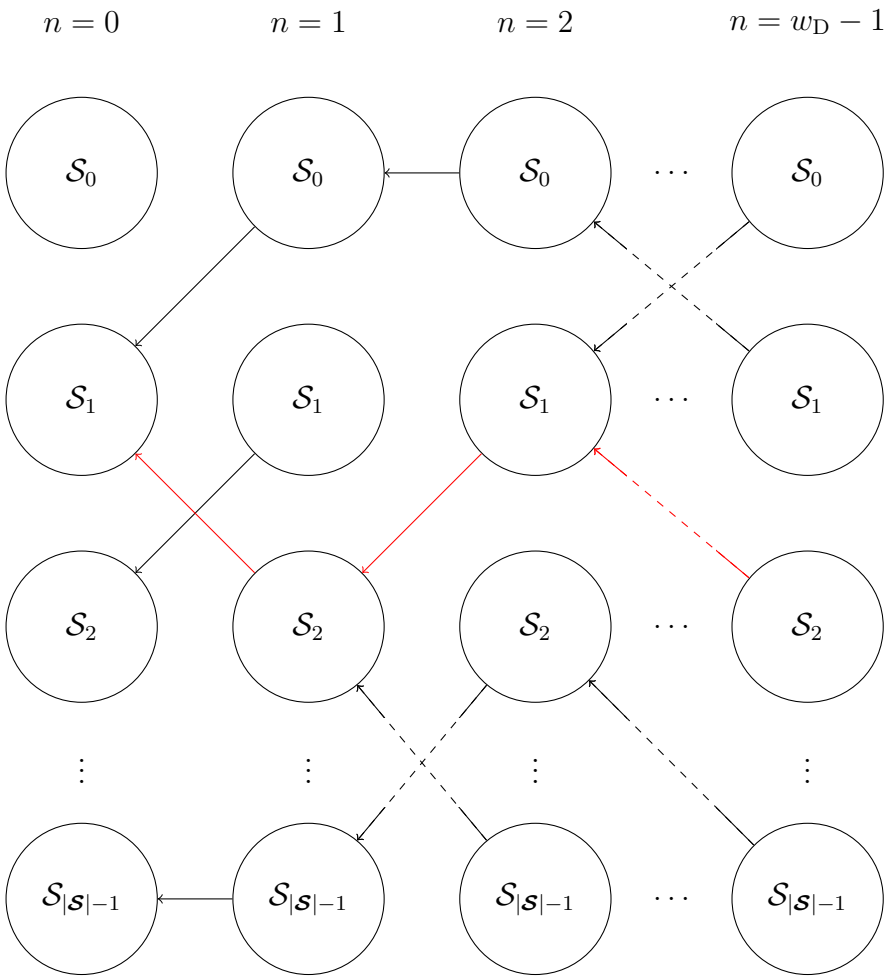


Figure 5.11. The Viterbi algorithm generates a trellis $\boldsymbol{\Phi}$. At iteration n, for each state $\mathcal{S}$ in the HMM the most probable preceding state is stored in $\Phi[n]_{\mathcal{S}}$, acting as a pointer. The data structure is computed with $O\left(w_\mathrm{D} \cdot |\boldsymbol{\mathcal{S}}|^2\right)$ time. Discarding the encoder balance information reduces the number of states in the HMM and consequently the computation time of $\boldsymbol{\Phi}$. Instead, the partially completed $\boldsymbol{\Phi}$ can be used to compute the encoder balance information for arbitrary paths through $\boldsymbol{\Phi}$ with which the transition probabilities of the HMM can be adjusted. This approach allows computation of state sequences with fewer states. Once $\boldsymbol{\Phi}$ has been fully constructed, the most probable sequence of states (red) can be read out as the path which culminates in the most probable final state.

$\mathcal{S} \in \boldsymbol{\mathcal{S}}$. We make a wideband signal estimate of the target device while displaying states $\boldsymbol{\mathcal{S}}$ in sequence $\boldsymbol{l}$, of which we make observations $\boldsymbol{o}_\mathbf{p}$. We generate a histogram for each $\mathcal{S} \in \boldsymbol{\mathcal{S}}$ and normalise this by the number of trials to estimate the emission probabilities:

$$\mathrm{P}(\mathcal{O} \,|\, \mathcal{S}) = \frac{|\{i \,:\, o_\mathrm{p}[i] = \mathcal{O} \wedge l[i] = \mathcal{S}\}|}{|\{j \,:\, l[j] = \mathcal{S}\}|}\,, \tag{5.46}$$

for the target device.

5.3.2 Viterbi decoding

To find the most likely underlying sequence of states $\boldsymbol{q}_m$ given observations $\boldsymbol{o}_m$, we use the Viterbi algorithm [112, p.86], a dynamic programming approach to convert $\boldsymbol{o}_m$ into a sequence of valid TMDS states. The Viterbi algorithm computes a table $\boldsymbol{\Phi}$, also known as a trellis (Figure 5.11), of paths through the HMM. At time n, $\Phi[n]_{\mathcal{S}}$ tells us the most likely state at $n-1$ which transitions to state $\mathcal{S}$. $\boldsymbol{\Phi}$ is constructed with a time complexity of $O(w_{\mathrm{D}} \cdot |\boldsymbol{\mathcal{S}}|^2)$; by discarding the encoder balance information in §5.3.1 to reduce the state space, we achieve a factor 4.8×10^3 reduction in time complexity. However, this comes at a cost of ignoring the encoder signal balance constraint which could be used to prevent invalid state transitions. I modify the Viterbi algorithm to adjust the transition probabilities on-the-fly by keeping track of the encoder's signal balances for paths through $\boldsymbol{\Phi}$, removing the need to store the encoder's balance-counter values in the states, which would result in combinatorial explosion, thus still preventing consideration of impossible state transitions. This approach is similar to the one presented by Backes *et al.* [6], who also adjusted the transition probabilities to reduce an HMM's state space when decoding the acoustic emissions of a dot-matrix printer. Let

$$\psi(i) = \begin{cases} \text{Positive column} & i > 0 \\ \text{Starred values} & i = 0 \\ \text{Negative column} & i < 0 \end{cases} \tag{5.47}$$

be a function that returns the set of valid TMDS characters for some signal balance i from Table A.2, and define

$$\Psi_{\mathcal{S}}(\boldsymbol{c}) = \prod_{j \in \{\mathrm{R,G,B}\}} \mathbb{1}_{\mathcal{S}_j \in \psi(\mathrm{sgn}(c_j))} \tag{5.48}$$

as an indicator function for whether state $\mathcal{S}$ is valid given a tuple of encoder signal balances. Then we can compute the tuple of signal balances for $\Phi[n]_{\mathcal{S}}$ as:

$$c[n]_{\mathcal{S}} = \left(\phi\left(\mathrm{dec}\left(\mathcal{S}_j\right), c[n-1]_{\Phi[n]_{\mathcal{S}}}\right) \,:\, j \in \{\mathrm{R,\ G,\ B}\}\right), \tag{5.49}$$

where $\phi(u, i)$ is a function which computes the signal balance produced by brightness u when the existing balance is i (Figure A.1). For each raster line m, we iterate through $\boldsymbol{o}_m$ for pixels $n \in \mathbb{Z}_{w_{\mathrm{D}}}$ (the blanking interval can be ignored as it does not contain video information). Starting at pixel $n = 0$, we compute the prior probability of each $\mathcal{S} \in \boldsymbol{\mathcal{S}}$ being the first state in the output sequence $q_m[0]$:

$$\mathrm{P}(q_m[0] = \mathcal{S}) = \frac{\Psi_{\mathcal{S}}((0,0,0))}{\sum_{\hat{\mathcal{S}} \in \boldsymbol{\mathcal{S}}} \Psi_{\hat{\mathcal{S}}}((0,0,0))}, \tag{5.50}$$

where $(0, 0, 0)$ is the initial encoder balance count for the red, green, and blue channels respectively. Now given $o_m[0]$, the probability each $\mathcal{S} \in \boldsymbol{\mathcal{S}}$ is $q_m[0]$ becomes[3]

$$\mathrm{P}(q_m[0] = \mathcal{S} \,|\, o_m[0]) = \mathrm{P}(o_m[0] \,|\, \mathcal{S}) \cdot \mathrm{P}(q_m[0] = \mathcal{S})\,. \tag{5.51}$$

For $n > 0$, we compute the probability of transitioning from $\mathcal{S}' \in \boldsymbol{\mathcal{S}}$ at iteration $n-1$ to $\mathcal{S} \in \boldsymbol{\mathcal{S}}$ at iteration n. Naïvely using the transition probability in Equation 5.45 does not account for the encoder balance information which constrains the set of valid transitions. For each colour represented in the HMM there will be one valid transition to one of the 2^3 states given the signal balance $c[n-1]_{\mathcal{S}'}$. I define the modified transition probability:

$$\mathrm{P}(\mathcal{S} \,|\, \mathcal{S}', c[n-1]_{\mathcal{S}'}) = 2^3 \cdot \Psi_{\mathcal{S}}(c[n-1]_{\mathcal{S}'}) \cdot \mathrm{P}(\mathcal{S} \,|\, \mathcal{S}')\,. \tag{5.52}$$

If the transition is invalid, i.e. $\mathcal{S}$ cannot be reached from $\mathcal{S}'$ given $c[n-1]_{\mathcal{S}'}$, the transition probability is treated as zero. The valid transitions are multiplied by 2^3 to ensure $\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}} \mathrm{P}(\mathcal{S} \,|\, \mathcal{S}', c[n-1]_{\mathcal{S}'}) = 1$. Thus for $n > 0$ the probability of transitioning from $\mathcal{S}'$ to $\mathcal{S}$ becomes

$$\mathrm{P}(q_m[n] = \mathcal{S} \,|\, q_m[n-1] = \mathcal{S}', c[n-1]_{\mathcal{S}'}) = \mathrm{P}(q_m[n-1] = \mathcal{S}' \,|\, o_m[n-1]) \cdot \mathrm{P}(\mathcal{S} \,|\, \mathcal{S}', c[n-1]_{\mathcal{S}'})\,. \tag{5.53}$$

The most likely state transition is stored in the table

$$\Phi[n]_{\mathcal{S}} = \operatorname*{arg\,max}_{\mathcal{S}'\in\boldsymbol{\mathcal{S}}} \mathrm{P}(q_m[n] = \mathcal{S} \,|\, q_m[n-1] = \mathcal{S}', c[n-1]_{\mathcal{S}'})\,. \tag{5.54}$$

The probability of this transition given $o_m[n]$ is

$$\mathrm{P}(q_m[n] = \mathcal{S} \,|\, o_m[n]) = \mathrm{P}(o_m[n] \,|\, \mathcal{S}) \cdot \mathrm{P}(q_m[n] = \mathcal{S} \,|\, q_m[n-1] = \Phi[n]_{\mathcal{S}}, c[n-1]_{\Phi[n]_{\mathcal{S}}})\,. \tag{5.55}$$

Equations 5.53 to 5.55 are iterated[4] for all observations $o_m[n]$. The most likely final state

$$q_m[w_{\mathrm{D}} - 1] = \operatorname*{arg\,max}_{S\in\boldsymbol{\mathcal{S}}} \mathrm{P}(q_m[w_{\mathrm{D}} - 1] = \mathcal{S} \,|\, o_m[w_{\mathrm{D}} - 1]) \tag{5.56}$$

is selected and we backtrack through the table:

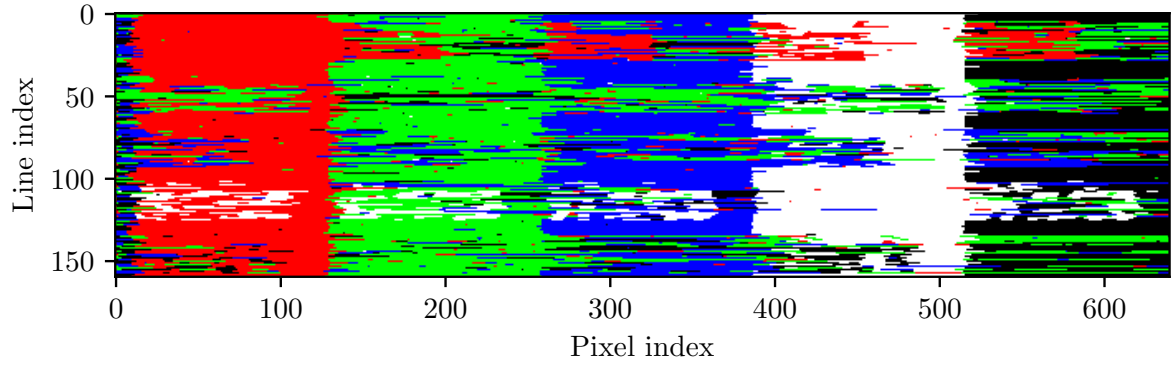
$$q_m[n-1] = \Phi[n]_{q_m[n]} \tag{5.57}$$

[3]The Viterbi algorithm omits the constant-valued denominator from Bayes' rule as it serves no purpose for comparison.

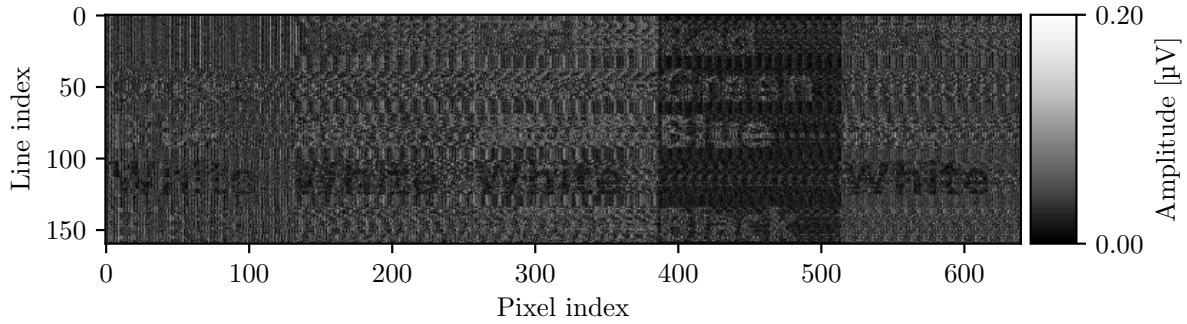
[4]In practice probabilities will become vanishingly small after a few iterations, making them unable to be represented using floating-point numbers. I used the negative log-probability

$$\mathbb{P}(\cdot) = -1 \cdot \begin{cases} \kappa & \mathrm{P}(\cdot) = 0 \\ \log \mathrm{P}(\cdot) & \text{otherwise} \end{cases}$$

to make them representable. Log-probabilities cannot represent zero values so using a suitably large $\kappa = 100$ makes the probability in effect equivalent to zero, but implicitly gives a small probability to possible observations not seen when profiling the device. When using this representation one must use addition instead of multiplication.



(a) HMM decoded signal.



(b) Amplitude-demodulated signal.

Figure 5.12. Having recovered the most probable sequence of TMDS states using a hidden Markov model we can raster the character sequence directly (a), rather than using phase or amplitude (b) information to form an estimate. This allows recovery of basic colour information. Here, I performed recovery on an enlarged section of the colour boxes from the test card (Figure B.1) using the Stitch data set for $C = 3$ where $\breve{B} = 160$ MHz and $\Upsilon = 80$ MHz to estimate a $B = 320$ MHz. The white and red states are best represented. Boundaries between the colour boxes not distinguishable in (b) are clearly defined in (a), and text not visible in the amplitude raster becomes visible, such as the 'Red' label in the black box.

to extract the most probable sequence of states for the HMM as our estimate of the transmitted character sequence $\boldsymbol{q}_m$. These values can be decoded and placed into the raster buffer (§2.6):

$$b[x, m, k] = \operatorname{dec}\left(q_m[x]_k\right) \quad \forall k \in \{\mathrm{R}, \mathrm{G}, \mathrm{B}\} \tag{5.58}$$

without further resampling or demodulation. An example of decoded data, where $\gamma = 10^{-4}$, is plotted in Figure 5.12a.

Let us observe how adjusting γ affects the pixel error rate

$$e = \frac{|\{(x, m) \,:\, b[x, m] = p[x, m]\}|}{N}, \tag{5.59}$$

where $\boldsymbol{p}$ are the true pixel values of length N. Figure 5.13 shows a curve indicating an optimal value for γ. Performance is seriously impacted if γ is too high resulting in

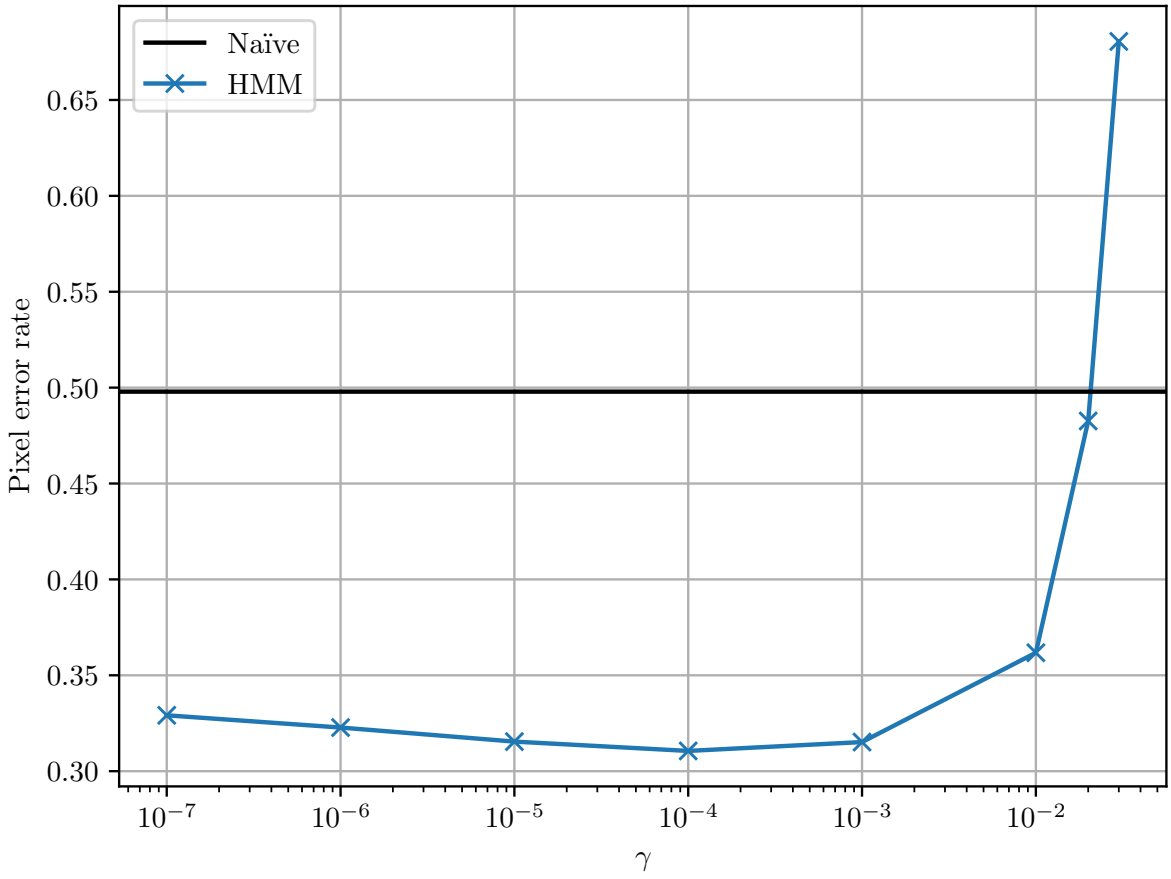


Figure 5.13. The pixel error rate as γ is varied. We can see there is an optimal propensity for character change. When comparing the performance of the HMM to the naïve approach of only using the emission probabilities there is a marked improvement in the error rate. This shows how using knowledge of the TMDS algorithm to decode samples can improve accuracy by preventing invalid state estimation.

the Viterbi decoder favouring a high rate of pixel change, which is inline with what we expected as the pixel data does not change very often. For comparison, I plot the pixel error rate when naïvly selecting the most probable pixel based only on the largest emission probability

$$q_m[n] = \arg\max_{\mathcal{S}\in\boldsymbol{\mathcal{S}}} \mathrm{P}(o_m[n]|\mathcal{S}) \tag{5.60}$$

This does not use any knowledge of the TMDS algorithm to decode the observations, and is equivalent to a simple template attack. We can see that in most cases using prior knowledge to decode an HMM provides significant improvement in the pixel error rate. The exception is when γ becomes too high such that the Viterbi decoder favours changing the pixel value over all other factors.

5.4 Summary

This chapter presented an approach for estimating wideband signal content from multiple overlapping narrowband measurements made using a single SDR front-end. In summary:

- Video information emanated by a VDU is spread across a wide frequency band. While measurements at bandwidths less than the bit rate can provide an intelligible estimation of screen content, it can result in a significant loss of available information. Making measurements with a wider bandwidth is often not an option due to hardware limitations of off-the-shelf SDR front-ends.

- A quasiperiodic video signal provides repeat opportunities to capture information, allowing us to obtain multiple narrowband measurements from across the RF spectrum. We can combine the narrowband measurements into an estimate of wideband content using the stitching algorithm: we use overlapping frequency regions to correct time and phase offsets between the narrowbands, filtering using a digital zero-phase L-R filter pair, and summing the result. The level of overlap must be enough to perform alignment and permit filter roll-off, but not so much as to require an unnecessary number of narrowband recordings risking an accumulation of phase noise from alignment.

- Estimating higher bandwidths reduced the level of blurring present in rasters made from low bandwidth recordings. Features that could not be observed at lower bandwidths became visible. With a wide enough bandwidth we can observe bit transitions. These can be used to identify specific combinations of TMDS characters being transmitted across a DVI cable.

- The information emanated by a given character varies across devices. Coupled with multiple concurrent emanating channels and additive noise, character recognition becomes a non-trivial task. We profiled a device to model the emanation for a set of characters and used an HMM to perform character recovery. Using this approach we recovered simple colour information from high-bandwidth estimates. Further research could investigate methods of separating the mixed channels and recovering sign information.

- The HMM required a large state space to represent the TMDS states which resulted in a prohibitively large computation time. By accounting for TMDS properties when decoding the HMM using the Viterbi algorithm instead, we were able to significantly reduce the time complexity.

Chapter 6

Conclusions

Many forms of side-channel attack require physical access to the device to extract a stored secret, such as a key or list of executed operations. Video eavesdropping attacks can circumvent the need to steal authentication credentials, allowing sensitive information to be extracted as it is accessed by a verified user. This is advantageous because no direct access is needed. However, it is a more opportunistic style of attack where the attacker must *take what they can get*, rather than targeting specific sensitive information. Nevertheless, eavesdropping can be used to spy on target habits for the purposes of blackmail and extortion, and combined with malware [113] eavesdropping can exfiltrate sensitive data on air-gapped networks at the attacker's leisure. Using SDR systems in eavesdropping attacks moves the majority of processing into software, allowing a wider range of processing techniques to be performed. When the effort for an attack can be reduced by purchasing inexpensive off-the-shelf equipment and installing the appropriate software packages, video eavesdropping can be executed with almost no knowledge of the underlying principles. In this thesis I investigated how the quasiperiodic property of emanated video signals can be exploited to assist eavesdropping attacks carried out using a software-defined radio.

The literature revealed little detailed information on carrying out a modern eavesdropping attack. Thus in Chapter 2 I presented a comprehensive guide to the current state of video eavesdropping using SDR. We saw the relative lack of sophistication, often extending to an amplitude raster with hand-tuned parameters. Such attacks do not scale, and valid criticisms were made of TEMPEST attacks not being cost effective. Even with falling hardware prices, the cost of manually carrying out an attack was greater than alternative methods to steal data—van Eck described a few days work for a knowledgeable person [15]. I went on to discuss methods of data presentation, including dynamic range compression, and the incorporation of phase information to the raster using HSV which improved contrast between foreground and background once the signal had been unrotated.

We started our investigation of eavesdropping techniques on methods of parameter recovery. There had been some work in the literature exploring parameter recovery from

a single known VDU, usually using autocorrelation for periodicity detection. I was more interested in locating and eavesdropping as yet unknown devices. In Chapter 3 we began by investigating the characteristics of an emanated video signal. We saw how, as a square wave, the pixel clock decomposes into harmonics throughout the frequency spectrum. I presented a method to identify sets of peaks which may be harmonics of an emanating pixel clock, however it could not differentiate clock harmonics from noise, nor identify the line or frame rates. On closer inspection, the frequency characteristic of the video signal revealed line and frame rate periodicities surrounding the harmonic peak. These were more easily investigated in the time domain using the autocorrelation. Using the recovered clock frequency information to locate the line and frame rates, large values in the autocorrelation indicated the presence of a given video mode. However, the results were difficult to separate from noise, even at high SNRs. Instead, I demonstrated how cepstral analysis can provide better distinction between periodicities in contrast to the established use of autocorrelation. The increased accuracy allowed identification of previously unknown devices. This was the case even when the video information was not discernible in a raster, allowing an attacker to discover the presence of devices even when the SNR was poor.

To counter the low SNR often encountered in eavesdropped data I attempted to perform periodic averaging of the signal, with limited success. Temperature changes in the clock quickly invalidated the detected pixel rate, causing the raster image to drift. Consequently, over a long averaging period the video information averaged itself out. I used a PLL to track changes in the harmonic we were attempting to eavesdrop. The PLL frequency controlled an NCO to match the phase of the clock signal. I presented an algorithm to implement the NCO with very high phase resolution using a DDS with integer arithmetic. I then used the tracked frequency information determined by the PLL to resample the video signal from the fixed sampling rate to variable rate using a resampling algorithm I derived from the DDS. By interpolating the signal using the detected pixel rate we stabilised the raster drift which permitted long-period averaging. The attenuation level of the noise was consistent with what was expected of AWGN, making it a suitable model for the background noise. While this improved the performance of periodic averaging, and successfully lifted the video signal out of the noise, the RF spectrum is a noisy place and I found that significant amounts of interference with defined structure obscured the result. To attenuate the non-AWGN noise I used a complex-domain averaging estimator: dividing the complex IQ samples with the NCO unrotated them with respect to the target's clock signal, aligning the phasor components of the video signal. This allowed averaging to take place in the complex domain, prior to demodulation, and resulted in attenuation of uncorrelated noise components whose arguments were not affected by the unrotation procedure. This significantly improved the clarity of eavesdropped rasters and the distance at which an attack could be carried out. The literature often drew a somewhat arbitrary distinction between information carrying and non-information carrying signal

components. Frequencies exceeding limit curves were considered acceptable if they did not directly carry information. However, this research shows that even emanating clock signals and refresh rates can have unwanted consequences for the privacy minded. In this case they allowed the detection of devices and averaging for arbitrary periods, allowing the SNR to be increased for as long as the data is shown. As such they should be considered information carrying, even if indirectly.

Regardless of how much we were able to raise the SNR, we were fundamentally limited by the narrow bandwidth provided by the SDR front-end. This can become a serious problem when attempting to eavesdrop high-resolution displays as insufficient bandwidth results in blurred raster features of little use to an eavesdropper. Quasiperiodic features can be recorded across a range of frequencies using a single device with consecutive recordings. The stitching algorithm I presented in Chapter 5 allowed estimation of wideband signal content from which we could recover significantly sharper raster images. I collected overlapping recordings of the emanated video signal from the UHF spectrum. Using cross-correlation, we were able to estimate and correct the time and phase offsets between the narrowbands, and filter them using a L-R filter pair to permit recombination via summation.

In amplitude rasters, signal features, such as relative brightness and colour, are lost and we are only able to estimate regions of similar brightness. While in some contexts this may be an acceptable result, such as for differentiating text from the background, I was interested in recovery of the underlying TMDS characters of the emanated signal, and reconstruction of the original pixel sequence. With enough bandwidth, the current-mode signal transitions between bit states were observable in the amplitude of the eavesdropped signal, and could be identified using a peak search. Recovering character information was complicated by eavesdropping on multiple concurrent emanating sources (the three colour transmission lines and the clock line), and unique emanation patterns between devices. I profiled a device for a subset of TMDS characters and used an HMM to extract the most likely sequence of valid TMDS characters. From this we were able to decode the characters using the TMDS decoding algorithm and recover true colour information.

6.1 Future work

I outline some ideas which can be explored to further the field of video eavesdropping:

- In §4.6 I briefly considered the effect averaging might have on TEMPEST limit curves, and how task duration should be factored into shielding considerations. I made assumptions regarding the theory of electromagnetic wave propagation which do not necessarily reflect the practice. There is scope to investigate more complex models of wave propagation and how they may affect eavesdropping limit curves.

- There are many other estimators than complex-averaging which can be used to attenuate non-AWGN noise e.g. a Kalman filter, or suppression of narrowband noise sources using notch filters. Comparison between different estimators could reveal interesting performance differences when applied to video eavesdropping, such as the suppression of background sources which are not completely uncorrelated with the target video signal. One could incorporate the operational parameters estimated in Chapter 3, or investigate alternative methods of parameter detection such as the spectral correlation density [69, p.366] to leverage the cyclostationary property.

- A limiting factor of the stitching algorithm was the accumulation of residual phase error from the alignment stage. This could be mitigated via additional processing to optimise phase alignment e.g. using gradient descent methods to adjust the phase offset between narrowbands, while observing the effect on a section of the signal with a known value, such as the blanking interval. The problem of accumulated phase error could be removed entirely by using a rack of low-cost SDR front-ends operating from the same clock signal. This would remove the need to perform time and phase alignment altogether, allowing the overlapping region to be reduced (to prevent gaps in the stitched result, a small overlapping region may still be desirable). As an extension to using a rack of front-ends, performing an online attack using stitched data would present an interesting challenge. Consideration towards the phase response of an online L-R filter pair, which could no long be implemented as a zero-phase filter, would need to be taken into account; otherwise alternate methods of filtering might be considered, such as quadrature mirror filters [114, p.277].

- I presented a method to recover colour information from high bandwidth signals, however additional research will be required before this can be deployed as a viable attack scenario. Some areas which would need to be developed:

  - My attempts to recover sign information were unsuccessful, resulting in the loss of useful information. As each clock harmonic in the stitched signal exhibited small differences in frequency due to being recorded across different times, I found unrotation did not permit coherent demodulation. Using a Costas loop [115, p.166] instead may allow a linear projection of the complex-valued samples to recover sign information.

  - Refinements to the HMM would foremost involve improved statistical modelling of the emission probabilities. Instead of using a binary string to represent an observation, a template of a character against which intercepted signals can be compared would permit a template attack to take advantage of all present information, such as the differing emission levels.

  - Instead of attempting to disentangle the superposition of colour and clock channels using an HMM, independent component analysis [116] might be em-

ployed to attempt separation of the additive signal components, permitting each colour channel to be analysed using a separate model.

  - I used a profiling stage to recover a subset of known TMDS characters. It may be possible to use a basic amplitude raster to estimate the colour of regions given expert knowledge, e.g. text is often a black font on a white background, removing the need for a separate profiling stage.

  - Finally, the effectiveness of recovering TMDS characters using stitched signals could be compared against other methods of recovering high bandwidths such as undersampling, or using an oscilloscope to recover baseband recordings.

- There has been comparatively little work on processing eavesdropped data outside of producing rasters for human consumption. Automatic feature recognition could be used in a number of contexts: Image processing techniques may perform character recognition to produce automatic transcripts of eavesdropped text. A simple template attack could be used for known fonts, or a neural network classifier for more general character recognition. To counter changing noise levels, an eavesdropping system placed at the boundary of a controlled zone could automatically shutdown RED carrying VDUs if the background noise level drops low enough for text to become recognisable.

- Smartphones are increasingly becoming desirable targets. However they present the additional challenge of being a moving target. We saw in §2.7 that changes in position can have a dramatic effect on the received SNR. Methods of countering for positional changes in a device would make eavesdropping mobile devices a practical attack vector. Antenna arrays could be used to collect multiple readings before applying principal component analysis to suppress background noise and provide a clear raster, even if the device exhibits movement.

- My focus was on passive attacks. However active attacks where the eavesdropper illuminates the device using their own carrier wave, or using other nearby sources similar to *screaming channel* attacks [117], can increase the SNR. Further to this, it may be possible to affect screen content remotely in a controlled manner. While precise control of the video data would seem unlikely, it is not unreasonable to consider a situation where interference could temporarily interrupt the video signal. Combined with clock tracking, and locating the target pixels in a raster, an attacker could potentially blank selective screen regions, censoring displayed information against the user's will.

6.2 Outlook

This thesis has endeavoured to expand the toolkit available to eavesdroppers. In the current threat model of video eavesdropping attacks, the risk posed is low enough that most need not concern themselves with protection. However, by continuing to make the attack more economical and easier to carry out without specialist knowledge, eavesdropping could prove a fertile ground for compromise, with little to no protection routinely in place. Highlighting the vulnerability of current VDUs serves to show that manufacturers should actively suppress emissions with a view to prevent eavesdropping, rather than just to comply with EMC requirements. This is no simple task: the exploits demonstrated relied heavily on periodicities inherent in video signals. In some cases removing components such as GUI widgets or similar consecutive frames, would not be practical. However components such as a separate clock signal and blanking interval could be removed or attenuated to increase the difficulty for an eavesdropper. For the time being however, it appears VDUs will remain vulnerable to eavesdropping.

Bibliography

[1] S. Sebastiani, "Characterization to a TEMPEST testing laboratory and methodology for control to compromising emanation [sic]," in *International Symposium on Electromagnetic Compatibility*, vol. 1, pp. 165–170 vol.1, Aug 1998.

[2] Rowe, N. C., "Electronic Protection," in *The Handbook of Information Security* (Bidgoli, H., ed.), New York: Wiley, 2006.

[3] M. Vuagnoux and S. Pasini, "An improved technique to discover compromising electromagnetic emanations," in *IEEE International Symposium on Electromagnetic Compatibility*, pp. 121–126, Jul 2010.

[4] Y. Hayashi, N. Homma, M. Miura, T. Aoki, and H. Sone, "A Threat for Tablet PCs in Public Space: Remote Visualization of Screen Images Using EM Emanation," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, (New York, NY, USA), pp. 954–965, ACM, 2014.

[5] M. Vuagnoux and S. Pasini, "Compromising Electromagnetic Emanations of Wired and Wireless Keyboards," in *Proceedings of the 18th Conference on USENIX Security Symposium*, SSYM '09, (Berkeley, CA, USA), pp. 1–16, USENIX Association, 2009.

[6] M. Backes, M. Dürmuth, S. Gerling, M. Pinkal, and C. Sporleder, "Acoustic Side-channel Attacks on Printers," in *Proceedings of the 19th USENIX Conference on Security*, USENIX Security '10, (Berkeley, CA, USA), p. 20, USENIX Association, 2010.

[7] C. Ulaş, U. Aşik, and C. Karadeniz, "The Search and Reconstruction of Compromising Emanations of Laser Printers in Three Media," in *International Journal of Information Security Science*, vol. 4, 2015.

[8] K. A. Ghani, K. Dimyati, K. Ismail, and L. S. Supian, "Radiated Emission from Handheld Devices with Touch-Screen LCDs," in *European Intelligence and Security Informatics Conference*, pp. 219–219, Aug 2013.

[9] G. P. Hancke, "Eavesdropping Attacks on High-Frequency RFID Tokens." Smart Card Centre, Royal Holloway, University of London `http://www.rfidblog.org.uk/Hancke-RFIDsec08-Eavesdropping.pdf`. [Accessed: 2018-02-08].

[10] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM Side-Channel(s)," in *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '02, (London, UK), pp. 29–45, Springer-Verlag, 2003.

[11] K. Iokibe, T. Amano, K. Okamoto, and Y. Toyota, "Equivalent Circuit Modeling of Cryptographic Integrated Circuit for Information Security Design," *IEEE Transactions on Electromagnetic Compatibility*, vol. 55, pp. 581–588, Jun 2013.

[12] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic Analysis: Concrete Results," in *Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '01, (London, UK), pp. 251–261, Springer-Verlag, 2001.

[13] R. Gehling, C. R. Ashley, and T. Griffin, "Electronic Emissions Security: Danger in the Air," *Information Systems Management*, vol. 24, no. 4, pp. 305–310, 2007.

[14] R. J. Anderson, *Security engineering: A guide to building dependable distributed systems*. Indianapolis, Ind.: Wiley, 2nd ed., 2008.

[15] W. van Eck, "Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?," *Comput. Secur.*, vol. 4, pp. 269–286, Dec 1985.

[16] H. J. Highland and V. Fåk, "Random Bits & Bytes: Electromagnetic Radiation Revisited, Part II," *Comput. Secur.*, vol. 5, pp. 181–184, Sep 1986.

[17] H. Fang, S. Changsheng, L. Deyun, and L. Guoding, "Measurement of radiated emission from PC computer system," in *IEEE International Symposium on Electromagnetic Compatibility*, pp. 208–209, Aug 1991.

[18] S. Dong, J. Xu, H. Zhang, and C. Wu, "On compromising emanations from computer VDU and its interception," in *3rd International Symposium on Electromagnetic Compatibility*, pp. 692–695, May 2002.

[19] H. Fang, "Radiated emission from CRT of computer VDU," in *IEEE International Symposium on Electromagnetic Compatibility*, pp. 58–61, Aug 1990.

[20] L. Jinming, Z. Jiemin, L. Taikang, and L. Yongmei, "The reconstitution of LCD compromising emanations based on wavelet denoising," in *12th International Conference on Computer Science and Education (ICCSE 2017)*, pp. 294–297, Aug 2017.

[21] M. G. Kuhn, "Compromising Emanations of LCD TV Sets," *IEEE Transactions on Electromagnetic Compatibility*, vol. 55, pp. 564–570, Jun 2013.

[22] H. S. Lee, J. G. Yook, and K. Sim, "Analysis of information leakage from display devices with LCD," in *URSI Asia-Pacific Radio Science Conference (URSI AP-RASC)*, pp. 1277–1280, Aug 2016.

[23] N. Zhang, Y. Lu, Q. Cui, and Y. Wang, "Investigation of Unintentional Video Emanations From a VGA Connector in the Desktop Computers," *IEEE Transactions on Electromagnetic Compatibility*, vol. 59, pp. 1826–1834, Dec 2017.

[24] E. Thiele, "Tempest for Eliza." `http://www.erikyyy.de/tempest/`, 2001. [Accessed: 2018-01-12].

[25] M. G. Kuhn and R. J. Anderson, "Soft Tempest: Hidden Data Transmission Using Electromagnetic Emanations," in *Information Hiding* (D. Aucsmith, ed.), (Berlin, Heidelberg), pp. 124–142, Springer Berlin Heidelberg, 1998.

[26] R. Gonggrijp and W. J. Hengeveld, "Studying the Nedap/Groenendaal ES3B Voting Computer: A Computer Security Perspective," in *Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology*, EVT '07, (Berkeley, CA, USA), p. 1, USENIX Association, 2007.

[27] Y. I. Hayashi, N. Homma, T. Watanabe, W. O. Price, and W. A. Radasky, "Introduction to the Special Section on Electromagnetic Information Security," *IEEE Transactions on Electromagnetic Compatibility*, vol. 55, pp. 539–546, Jun 2013.

[28] A. O. Bauer, "Some aspects of military line communications as deployed by the German armed forces prior to 1945," in *The History of Military Communications, Proceedings of the 5th Annual Colloquium*, Sep 1999.

[29] R. F. H. Nalder, *The Royal Corps of Signals: A history of its antecedents and development, circa 1800–1955*. London: Royal Signals Institute, 1958.

[30] National Security Telecommunications and Information Systems Security Committee, "Maintenance and disposition of TEMPEST equipment." NSTISSAM TEMPEST/1-00, Partially declassified `http://www.cryptome.org/nstissam1-00.htm`, Dec 2000. [Accessed: 2018-03-04].

[31] "NSA TEMPEST Documents." Partially declassified `http://cryptome.org/nsa-tempest.htm`. [Accessed: 2018-05-03].

[32] National Security Agency, "NACSIM 5000 TEMPEST Fundamentals." Partially declassified `http://cryptome.org/jya/nacsim-5000/nacsim-5000.htm`, Feb 1982. [Accessed: 2018-05-03].

[33] D. Russell and G. T. Gangemi, Sr., *Computer Security Basics*. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 1991.

[34] A. J. Mauriello, "Join a government program to unveil Tempest-spec mysteries," *EDN Magazine—Electrical Design News*, vol. 28, no. 13, p. 191, 1983.

[35] G. Bayraktar, "The new Requirement for a Fifth Dimension of war: Cyber Intelligence," in *Proceedings of the 9th International Conference on Cyber Warfare and Security*, pp. 9–16, 2014.

[36] H. J. Highland, "The Tempest over Leaking Computers," *Abacus*, vol. 5, pp. 10–18, Jan 1988.

[37] T. L. Song, Y. R. Jeong, H. S. Jo, and J. G. Yook, "Noise-Jamming Effect as a Countermeasure Against TEMPEST During High-Speed Signaling," *IEEE Transactions on Electromagnetic Compatibility*, vol. 57, pp. 1491–1500, Dec 2015.

[38] B. Pawliw, "Tempest." `https://web.archive.org/web/20060909205846/https://searchsecurity.techtarget.com/sdefinition/0,290660,sid14_gci522583,00.html`, May 2001. [Accessed: 2018-07-01].

[39] J. McNamara, "The Complete, Unofficial TEMPEST Information Page." `https://cryptome.org/tip/tempestintro.html`. [Accessed: 2018-09-06].

[40] National communications security committee, subcommittee on compromising emanations, "TEMPEST GLOSSARY." Partially declassified `https://cryptome.org/ncsc-3.htm`, Mar 1981. [Accessed: 2018-05-03].

[41] H. Sekiguchi and S. Seto, "Study on Maximum Receivable Distance for Radiated Emission of Information Technology Equipment Causing Information Leakage," *IEEE Transactions on Electromagnetic Compatibility*, vol. 55, pp. 547–554, Jun 2013.

[42] Y. Suzuki and Y. Akiyama, "Jamming technique to prevent information leakage caused by unintentional emissions of PC video signals," in *IEEE International Symposium on Electromagnetic Compatibility*, pp. 132–137, Jul 2010.

[43] C. Ulas, S. Şahin, E. Memişoğlu, U. Aşik, C. Karadeniz, B. Kılıç, and U. Saraç, "Development of an Automatic TEMPEST Test and Analysis System," in *Scientific Cooperations International Workshops on Electrical and Computer Engineering Subfields*, Aug 2014.

[44] E. Le Bail, "L'anticompromission Electromagnetique," Master's thesis, Université du Maine.

[45] K. Beckman, "Läckande Datorer [Leaking Computers]," 1984. Swedish Government Commission on the Vulnerability of EDP Equipment and the National Council for Crime Prevention [Swedish language].

[46] G. Ward, "TEMPEST in a teapot." cypherpunks-list #1317 `http://imchris.org/projects/cpunk.html`, 1993. [Accessed: 2018-03-16].

[47] M. G. Kuhn, "Security Limits for Compromising Emanations," in *Proceedings of the 7th International Conference on Cryptographic Hardware and Embedded Systems*, CHES '05, (Berlin, Heidelberg), pp. 265–279, Springer-Verlag, 2005.

[48] H. J. Highland, "Random Bits & Bytes: Electromagnetic Radiation Revisited," *Comput. Secur.*, vol. 5, pp. 85–93, Jun 1986.

[49] C. J. Seline, "Eavesdropping On the Electromagnetic Emanations of Digital Equipment: The Laws of Canada, England and the United States." `https://cryptome.org/tempest-law.htm`, Nov 1999. [Accessed: 2018-05-03].

[50] D. Stoica, A. Boteanu, M. Enache, and A. Plosnita, "Influence of the interconnecting cables on equipments electromagnetic emissions spectrum," in *International Conference of Scientific Paper AFASES 2015*, May 2015.

[51] Y. I. Hayashi, N. Homma, T. Mizuki, H. Shimada, T. Aoki, H. Sone, L. Sauvage, and J. L. Danger, "Efficient Evaluation of EM Radiation Associated With Information Leakage From Cryptographic Devices," *IEEE Transactions on Electromagnetic Compatibility*, vol. 55, pp. 555–563, Jun 2013.

[52] E. Möller, "Protective Measures Against Compromising Electro Magnetic Radiation Emitted by Video Display Terminals," *Phrack Magazine*, vol. 4, no. 44.

[53] R. J. Anderson, "Tempest—A Guide for the Perplexed." Computer Laboratory, University of Cambridge, United Kingdom.

[54] M. Urban, *UK Eyes Alpha: The Inside Story of British Intelligence*. Faber and Faber, 1996.

[55] L. Sauvage, J. L. Danger, S. Guilley, N. Homma, and Y. I. Hayashi, "Advanced Analysis of Faults Injected Through Conducted Intentional Electromagnetic Interferences," *IEEE Transactions on Electromagnetic Compatibility*, vol. 55, pp. 589–596, Jun 2013.

[56] "Tempest comes in from the cold?," *Electronics and Power*, vol. 33, p. 424, Jul 1987.

[57] Y. I. Hayashi, N. Homma, T. Mizuki, T. Aoki, H. Sone, L. Sauvage, and J. L. Danger, "Analysis of Electromagnetic Information Leakage From Cryptographic Devices With Different Physical Structures," *IEEE Transactions on Electromagnetic Compatibility*, vol. 55, pp. 571–580, Jun 2013.

[58] J. McConnell, "RED/BLACK Installation Guidance." US Navy, NSTISSAM TEMPEST/2-95, Partially declassified `http://cryptome.org/tempest-2-95.htm`, Dec 1995. [Accessed: 2015-10-05].

[59] B. Audone and F. Bresciani, "Signal processing in active shielding and direction-finding techniques," *IEEE Transactions on Electromagnetic Compatibility*, vol. 38, pp. 334–340, Aug 1996.

[60] T. Watanabe, H. Nagayoshi, T. Urano, T. Uemura, and H. Sako, "Countermeasure for electromagnetic screen image leakage based on color mixing in human brain," in *IEEE International Symposium on Electromagnetic Compatibility*, pp. 138–142, Jul 2010.

[61] M. G. Kuhn, "AntiSky—an Image Processing Attack on VideoCrypt." `https://www.cl.cam.ac.uk/~mgk25/tv-crypt/image-processing/antisky.html`, 1994. [Accessed: 2015-02-15].

[62] E. MacAskill, "New NSA leaks show how US is bugging its European allies." The Guardian `http://www.theguardian.com/world/2013/jun/30/nsa-leaks-us-bugging-european-allies`, Jun 2013. [Accessed: 2015-06-29].

[63] M. Engelhardt, F. Pfeiffer, K. Finkenzeller, and E. Biebl, "Extending ISO/IEC 14443 Type A Eavesdropping Range using Higher Harmonics," in *Smart SysTech 2013; European Conference on Smart Objects, Systems and Technologies*, pp. 1–8, Jun 2013.

[64] F. Elibol, U. Sarac, and I. Erer, "Realistic eavesdropping attacks on computer displays with low-cost and mobile receiver system," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pp. 1767–1771, Aug 2012.

[65] M. Marinov, "Remote video eavesdropping using a software-defined radio platform," Master's thesis, Computer Laboratory, University of Cambridge, United Kingdom, Jun 2014. `https://github.com/martinmarinov/TempestSDR` [Accessed: 2018-09-06].

[66] M. Guri, G. Kedma, A. Kachlon, and Y. Elovici, "AirHopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies," in *2014 9th International Conference on Malicious and Unwanted Software: The Americas (MALWARE)*, pp. 58–67, Oct 2014.

[67] Z. Yang, J. Zhang, Z. Wang, and Q. Zhang, "Lightweight Display-to-device Communication Using Electromagnetic Radiation and FM Radio," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Mobihoc '17, (New York, NY, USA), pp. 1–2, ACM, 2017.

[68] Digital Display Working Group, "Digital Visual Interface, DVI," Apr 1999. Revision 1.0.

[69] A. Napolitano, *Generalizations of cyclostationary signal processing: spectral analysis and applications*. Chichester: Wiley, 2012.

[70] W. Wolf and H. Deubel, "P31 phosphor persistence at photopic mean luminance level," vol. 10, pp. 323–333, Feb 1997.

[71] HDMI Licensing, "HDMI Specification 1.3a," Nov 2006. Section 5.1.2.

[72] Fairchild Semiconductor, "AN-5017 LVDS Fundamentals." `http://s.eeweb.com/articles/2012/03/06/LVDS-Fundamentals-1331048990.pdf`, Dec 2005. [Accessed: 2015-02-12].

[73] T. L. Song, Y. R. Jeong, and J. G. Yook, "Modeling of Leaked Digital Video Signal and Information Recovery Rate as a Function of SNR," *IEEE Transactions on Electromagnetic Compatibility*, vol. 57, pp. 164–172, Apr 2015.

[74] H. S. Lee, J. G. Yook, and K. Sim, "Measurement and analysis of the electromagnetic emanations from video display interface," in *IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*, pp. 71–73, Dec 2015.

[75] M. Bach, T. Meigen, and H. Strasburger, "Raster-scan cathode-ray tubes for vision research—Limits of resolution in space, time and intensity, and some solutions," vol. 10, pp. 403–414, Feb 1997.

[76] R. M. Gray and L. D. Davisson, *An Introduction to Statistical Signal Processing*. Cambridge University Press, 2004.

[77] M. Gruber, "Synchronous Detection of AM Signals: What is it and how does it work?," *QEX Magazine*, pp. 9–16, Sep 1992.

[78] N. Instruments, "Configuration: Gain Property." `http://zone.ni.com/reference/en-XX/help/373380G-01/usrppropref/pniusrp_gain/`, Jan 2017. [Accessed: 2018-04-02].

[79] "Receiver Dynamics," *Cryptologic Quarterly*, vol. 15, no. 4, 1996. `https://www.nsa.gov/news-features/declassified-documents/cryptologic-quarterly/assets/files/Receiver_Dynamics.pdf` [Accessed: 2018-02-02].

[80] M. G. Kuhn, "Electromagnetic Eavesdropping Risks of Flat-Panel Displays," in *Privacy Enhancing Technologies* (D. Martin and A. Serjantov, eds.), (Berlin, Heidelberg), pp. 88–107, Springer Berlin Heidelberg, 2005.

[81] L. Nowosielski, R. Przesmycki, and M. Nowosielski, "Compromising Emanations from VGA and DVI Interface," in *Progress in Electromagnetic Research Symposium*, pp. 1024–1028, Aug 2016.

[82] E. Reinhard, E. A. Khan, A. O. Akyz, and G. M. Johnson, *Color Imaging: Fundamentals and Applications*. Natick, MA, USA: A. K. Peters, Ltd., 2008.

[83] M. G. Kuhn, "Compromising video emanations of Sdu NewVote machines," Nov 2007. University of Cambridge, Computer Laboratory.

[84] A. Zajic, M. Prvulovic, and D. Chu, "Path loss prediction for electromagnetic side-channel signals," in *11th European Conference on Antennas and Propagation (EUCAP)*, pp. 3877–3881, Mar 2017.

[85] H. S. Efendioglu, M. Yabuloglu, and H. Ozer, "Effectiveness of shielded window films used for electromagnetic leakages in communication security," in *International Electromagnetic Compatibility Conference (EMC Turkiye)*, pp. 1–4, Sep 2017.

[86] Y. Du and H. Zhang, "Estimating the eavesdropping distance for radiated emission and conducted emission from information technology equipment," in *IEEE 5th International Symposium on Electromagnetic Compatibility (EMC-Beijing)*, pp. 1–7, Oct 2017.

[87] J. L. Press, "EMP response of a generic groundbased facility," in *IEEE International Symposium on Electromagnetic Compatibility*, pp. 74–79, Aug 1990.

[88] Consumer Electronics Association, "CEA Standard: A DTV Profile for Uncompressed High Speed Digital Interfaces," Aug 2005. CEA-816-C.

[89] K. Martin and S. A. L. Ward, "Television detector vans," *British Telecommunications engineering.*, pp. 180–186, Oct 1984.

[90] P. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, pp. 70–73, Jun 1967.

[91] S. P. Peterson, A. A. Sarkady, and C. B. Cameron, "Reconstruction of Periodic Sonar Signals Hidden in Wideband Noise Using Ensemble Averaging and Multi-Rate DSP." United States Naval Academy `https://apps.dtic.mil/dtic/tr/fulltext/u2/a575186.pdf`, Apr 2003. [Accessed: 2018-02-08].

[92] Video Electronics Standards Association, "VESA and Industry Standards and Guidelines for Computer Display Monitor Timing (DMT)," Nov 2008. Version 1.0, Revision 12.

[93] M. G. Kuhn, "Compromising emanations: eavesdropping risks of computer displays," Tech. Rep. UCAM-CL-TR-577, University of Cambridge, Computer Laboratory, Dec 2003.

[94] J. Shi, W. Q. Huang, D. Wei, and D. G. Sun, "A novel method for computer video leaking signal detection," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 918–923, Aug 2014.

[95] D. G. Sun, J. Shi, D. Wei, and M. Zhang, "A low-cost and efficient method of determining the best frequency band for video leaking signal reconstruction," in *2nd International Conference on Systems and Informatics (ICSAI 2014)*, pp. 624–628, Nov 2014.

[96] L. Cohen, "Generalization of the Wiener-Khinchin theorem," *IEEE Signal Processing Letters*, vol. 5, pp. 292–294, Nov 1998.

[97] B. P. Bogert, M. J. R. Healy, and J. W. Tukey, "The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo Autocovariance, Cross-Cepstrum and Saphe Cracking," in *Proceedings of the Symposium on Time Series Analysis* (M. Rosenblatt, ed.), pp. 209–243, New York: Wiley, 1963.

[98] M. Hall, "Predicting bed thickness with cepstral decomposition," *The Leading Edge*, vol. 25, no. 2, p. 199, 2006.

[99] R. B. Randall, "A history of cepstrum analysis and its application to mechanical problems," *Mechanical Systems and Signal Processing*, vol. 97, pp. 3–19, 2017. Special Issue on Surveillance.

[100] International Telecommunication Union, "Radio noise: Recommendation ITU-R P.372-13." `https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.372-13-201609-I!!PDF-E.pdf`, Sep 2016. [Accessed: 2018-07-10].

[101] W. R. Stevens, *UNIX network programming Vol.1, Networking APIs: sockets and XTI.* Upper Saddle River, N.J.: Prentice Hall PTR, 2nd ed., 1998.

[102] A. Visioli, *Practical PID control.* Advances in industrial control, Springer, 2006.

[103] E. Rubiola, *Phase Noise and Frequency Stability in Oscillators.* Cambridge University Press, Nov 2008.

[104] H. S. Lee, K. Sim, S. Oh, and J. G. Yook, "Analysis of the Electromagnetic Leakage from Liquid Crystal Display Monitors," vol. 27, pp. 844–853, Sep 2016.

[105] J. Blinn, *Jim Blinn's corner: dirty pixels*. Morgan Kaufmann series in computer graphics and geometric modeling, San Francisco, CA.: Morgan Kaufmann Publishers, 1998.

[106] J. Idier and G. Collewet, "Properties of Fisher information for Rician distributions and consequences in MRI." [Accessed: 2018-09-01], Oct 2014.

[107] L. Nowosielski and J. Michalak, "Shielding effectiveness required for IT equipment enclosures," in *Progress in Electromagnetics Research Symposium—Fall (PIERS—FALL)*, pp. 427–431, Nov 2017.

[108] P. Smulders, "The threat of information theft by reception of electromagnetic radiation from RS-232 cables," *Computers & Security*, vol. 9, no. 1, pp. 53–58, 1990.

[109] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Pub., 1999.

[110] S. H. Linkwitz, "Active crossover networks for noncoincident drivers," *J. Audio Eng. Soc*, vol. 24, no. 1, pp. 2–8, 1976.

[111] *Handbook of data communications*. Manchester: National Computing Centre, 1975.

[112] G. A. Fink, *Hidden Markov Models*, pp. 71–106. London: Springer, 2014.

[113] M. Guri, B. Zadov, A. Daidakulov, and Y. Elovici, "ODINI : Escaping Sensitive Data from Faraday-Caged, Air-Gapped Computers via Magnetic Fields," *ArXiv e-prints*, Feb 2018.

[114] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of signal processing*. Cambridge University Press, 2013.

[115] P. W. Ward, J. W. Betz, and C. J. Hegarty, "Satellite signal acquisition, tracking, and data demodulation," in *Understanding GPS: principles and applications* (E. D. Kaplan and C. J. Hegarty, eds.), Artech House mobile communications series, (Boston, Mass.; London), Artech House, 2006.

[116] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Netw.*, vol. 13, pp. 411–430, May 2000.

[117] G. Camurati, S. Poeplau, M. Muench, T. Hayes, and A. Francillon, "Screaming channels: When electromagnetic side channels meet radio transceivers," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, (New York, NY, USA), pp. 163–177, ACM, 2018.

Appendix A

Transition-minimized differential signalling

The TMDS encoding algorithm is designed to transform 8 bit brightness values into 10 bit characters (8b/10b encoding scheme) for transmission over a cable using differential signalling. It is commonly used in DVI and HDMI. The purpose of encoding bytes is twofold: minimise the number of bit transitions to reduce electromagnetic emissions caused by current change, and remove the DC bias by transmitting zero and one bits in equal measure, ensuring the waveform has a mean value of zero. To transmit colour, three encoders work in parallel, encoding the red, green, and blue components respectively. A clock signal is also transmitted along its own wire pair. The 256 brightness values can be encoded into 460 characters depending on the signal balance, with between two and five transitions per character. Valid characters are listed for reference in Table A.2. During the blanking interval there are just four valid characters, two pairs of complementary 10 bit characters allowing two bits of control information (C_0 and C_1), with eight transitions in order to make them easily identifiable, listed in Table A.1.

C_0	C_1	Bit pattern	Imbalance	Transitions
0	0	0010101011	0	8
0	1	1101010100	0	8
1	0	0010101010	−2	8
1	1	1101010101	2	8

Table A.1. The TMDS characters (big endian) used to transmit the control bits C_0 and C_1 during the blanking interval, along with the imbalance and circular transition count.

A.1 Encoding

Let us define an input byte as a tuple of 8 bits

$$\boldsymbol{b} = (b_0, b_1, \ldots, b_7) \tag{A.1}$$

which represent a brightness value where $b[k] \in \mathbb{Z}_2$ and $b[0]$ is the least significant bit (LSB)

$$B = \sum_{k=0}^{7} b[k] \cdot 2^k \,. \tag{A.2}$$

Likewise we will define the 10 bit output character

$$\boldsymbol{q} = (q_0, q_1, \ldots, q_9) \tag{A.3}$$

First, transition minimisation is performed by applying either an XOR or XNOR operation on consecutive bits, acting as a kind of first-order derivative. The ninth bit of the output character indicates if XOR was applied

$$q[8] = \mathbb{1}_{(N_1\{\boldsymbol{b}\}\leq 4)\wedge(N_1\{\boldsymbol{b}\}\neq 4 \vee b[0]=1)} \,, \tag{A.4}$$

where the count of a bit type in a byte is defined as

$$N_i\{\boldsymbol{b}\} = \sum_{b\in\boldsymbol{b}} \mathbb{1}_{b=i} \,. \tag{A.5}$$

The appropriate operation is applied to produce the intermediate result

$$b'[k] = \begin{cases} b[0] & k = 0 \\ b'[k-1] \oplus b[k] & q[8] = 1 \\ b'[k-1] \mathbin{\overline{\oplus}} b[k] & \text{otherwise} \end{cases} \,. \tag{A.6}$$

The encoder keeps an approximate running total of the signal balance $c[n]$ (Figure A.1). If the stream exhibits an imbalance and $\boldsymbol{b'}$ will increase the magnitude of c the tenth bit of $\boldsymbol{q}$ is set

$$q[9] = \mathbb{1}_{\mathrm{sgn}(c[n-1])=\mathrm{sgn}(N_1\{b'\}-N_0\{b'\})} \,, \tag{A.7}$$

and the byte inverted

$$q[k] = b'[k] \oplus q[9] \quad \forall k \in \mathbb{Z}_8 \tag{A.8}$$

in an attempt to redress the balance. During the blanking interval two control bits C_0 and C_1 dictate the output

$$q[k] = \mathrm{C}_1 \oplus \begin{cases} \neg\mathrm{C}_0 & k = 9 \\ 0 & (k \bmod 2 = 1 \vee k = 0) \wedge k \neq 9 \\ 1 & \text{otherwise} \end{cases} \,, \tag{A.9}$$

and the counter c is reset to zero, meaning the encoder has no memory across lines. The encoded signal has a mean of approximately zero as the blanking interval is not fully balanced. The encoding algorithm is summarised by the flow diagram in Figure A.1, simplified from the standards document [68, p.29] in an attempt to remove unnecessary complexity introduced by implementation detail. Characters are transmitted LSB first.

A.2 Decoding

Unlike the encoding algorithm, the decoding process does not require any state information to recover the intensity value of a character. First the character is inverted if the tenth bit is set

$$b'[k] = q[k] \oplus q[9] \quad \forall k \in \mathbb{Z}_8\,, \tag{A.10}$$

before undoing the XOR or XNOR operation

$$b[k] = \begin{cases} b'[0] & k = 0 \\ b'[k] \mathbin{\overline{\oplus}} b'[k-1] & q[8] = 0 \\ b'[k] \oplus b'[k-1] & \text{otherwise} \end{cases}. \tag{A.11}$$

For the recovery of control bit information transmitted during the blanking interval a lookup table can be used by the implementation.

A.3 Example encodings

Some example encodings from brightness values to TMDS characters:

16	16	16	16	16	16	16	16	16	16	→	496	496	496	496	496	496	496	496	496	496
0	0	0	0	0	0	0	0	0	0	→	256	1023	256	1023	256	1023	256	1023	256	256
0	1	2	3	4	5	6	7	8	9	→	256	511	510	257	508	259	258	509	504	263
43	54	123	43	12	34	103	22	43	55	→	281	583	636	281	1019	286	648	781	281	696
0	0	0	255	255	255	255	0	0	0	→	256	1023	256	255	512	255	255	256	1023	256

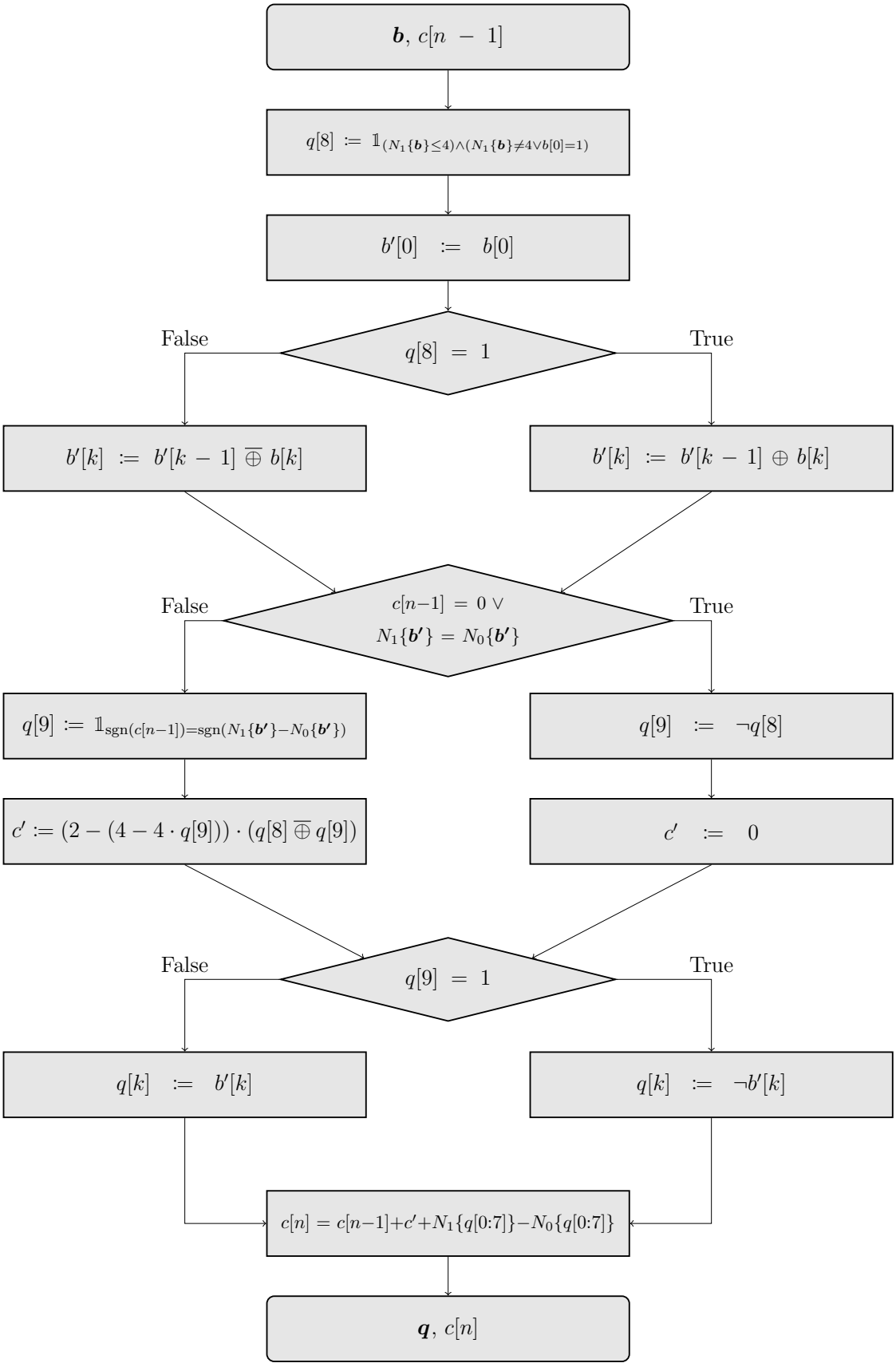


Figure A.1. The TMDS encoding algorithm. I have redrawn this version from the flow diagram presented in the standards documentation to simplify several programming quirks of the Verilog implementation which were originally included. I omit the blanking interval lookup operation for brevity.

Brightness	Positive	I	T	Negative	I	T	Brightness	Positive	I	T	Negative	I	T	Brightness	Positive	I	T	Negative	I	T
0	0100000000*	−8	2	1111111111	10	0	86	0010011000	−4	4	1001100111*	2	4	172	1000110001*	−2	4	0011001110	0	4
1	1100000000	−6	2	0111111111*	8	2	87	1010011000*	−2	6	0001100111	0	4	173	0000110001	−4	4	1011001110*	2	6
2	1100000001	−4	2	0111111110*	6	2	88	0111001000*	−2	4	1100110111	4	4	174	0000110000	−6	2	1011001111*	4	4
3	0100000001*	−6	4	1111111110	8	2	89	1111001000	0	4	0100110111*	2	6	175	1000110000*	−4	4	0011001111	2	4
4	1100000011	−2	2	0111111100*	4	2	90	1001100011	0	4	1001100111	0	4	176	0110010000*	−4	4	1101101111	6	4
5	0100000011*	−4	4	1111111100	6	2	91	1010011100	0	6	1010011100	0	6	177	1110010000	−2	4	0101101111*	4	6
6	0100000010*	−6	4	1111111101	8	2	92	1001100001*	−2	4	0010011110	0	4	178	0011000100	−4	4	1000111011*	2	4
7	1100000010	−4	4	0111111101*	6	4	93	0001100001	−4	4	1010011110*	2	6	179	1011000100*	−2	6	0000111011	0	4
8	1100000111	0	2	0111111000*	2	2	94	0001100000	−6	2	1010011111*	4	4	180	1000111001	0	4	1000111001	0	4
9	0100000111*	−2	4	1111111000	4	2	95	1001100000*	−4	4	0010011111	2	4	181	1011000110	0	6	1011000110	0	6
10	0100000110*	−4	4	1111111001	6	2	96	0100100000*	−6	4	1111011111	8	2	182	0000111000	−4	2	1011000111*	2	4
11	1100000110	−2	4	0111111001*	4	4	97	1100100000	−4	4	0111011111*	6	4	183	1000111000*	−2	4	0011000111	0	4
12	0100000100*	−6	4	1111111011	8	2	98	1100100001	−2	4	0111011110*	4	4	184	0011000010	−4	4	1000111101*	2	4
13	1100000100	−4	4	0111111011*	6	4	99	0100100001*	−4	6	1111011110	6	4	185	1011000010*	−2	6	0000111101	0	4
14	1100000101	−2	4	0111111010*	4	4	100	1100100011	0	4	0111011100*	2	4	186	1011000011	0	4	1011000011	0	4
15	0100000101*	−4	6	1111111010	6	4	101	0100100011*	−2	6	1111011100	4	4	187	1000111100	0	4	1000111100	0	4
16	0111110000	0	2	0111110000	0	2	102	0010001000	−6	4	1001110111*	4	4	188	1011000001*	−2	4	0000111110	0	2
17	0100001111	0	4	0100001111	0	4	103	1010001000*	−4	6	0001110111	2	4	189	0011000001	−4	4	1000111110*	2	4
18	0100001110*	−2	4	1111110001	4	2	104	0111011000	0	4	0111011000	0	4	190	0011000000	−6	2	1000111111*	4	2
19	1100001110	0	4	0111110001*	2	4	105	0100100111	0	6	0100100111	0	6	191	1011000000*	−4	4	0000111111	2	2
20	0100001100*	−4	4	1111110011	6	2	106	0010001100	−4	4	1001110011*	2	4	192	0101000000*	−6	4	1110111111	8	2
21	1100001100	−2	4	0111110011*	4	4	107	1010001100*	−2	6	0001110011	0	4	193	1101000000	−4	4	0110111111*	6	4
22	1100001101	0	4	0111110010*	2	4	108	1001110001	0	4	1001110001	0	4	194	1101000001	−2	4	0110111110*	4	4
23	0100001101*	−2	6	1111110010	4	4	109	1010001110	0	6	1010001110	0	6	195	0101000001*	−4	6	1110111110	6	4
24	0100001000*	−6	4	1111110111	8	2	110	0001110000	−4	2	1010001111*	2	4	196	1101000011	0	4	0110111100*	2	4
25	1100001000	−4	4	0111110111*	6	4	111	1001110000*	−2	4	0010001111	0	4	197	0101000011*	−2	6	1110111100	4	4
26	1100001001	−2	4	0111110110*	4	4	112	0111010000*	−2	4	1100101111	4	4	198	1000010111	0	4	1000010111	0	4
27	0100001001*	−4	6	1111110110	6	4	113	1111010000	0	4	0100101111*	2	6	199	1011101000	0	6	1011101000	0	6
28	1100001011	0	4	0111110100*	2	4	114	0010000100	−6	4	1001111011*	4	4	200	0110111000	0	4	0110111000	0	4
29	0100001011*	−2	6	1111110100	4	4	115	1010000100*	−4	6	0001111011	2	4	201	0101000111	0	6	0101000111	0	6
30	0010100000	−6	4	1001011111*	4	4	116	0010000110	−4	4	1001111001*	2	4	202	1000010011*	−2	4	0011101100	0	4
31	1010100000*	−4	6	0001011111	2	4	117	1010000110*	−2	6	0001111001	0	4	203	0000010011	−4	4	1011101100*	2	6
32	0111100000*	−2	2	1100011111	4	2	118	1010000111	0	4	1010000111	0	4	204	1000010001*	−4	4	0011101110	2	4
33	1111100000	0	2	0100011111*	2	4	119	1001111000	0	4	1001111000	0	4	205	0000010001	−6	4	1011101110*	4	6
34	0100011110	0	4	0100011110	0	4	120	0010000010	−6	4	1001111101*	4	4	206	0000010000	−8	2	1011101111*	6	4
35	0111100001	0	4	0111100001	0	4	121	1010000010*	−4	6	0001111101	2	4	207	1000010000*	−6	4	0011101111	4	4
36	0100011100*	−2	4	1111100011	4	2	122	1010000011*	−2	4	0001111100	0	2	208	0110110000*	−2	4	1101001111	4	4
37	1100011100	0	4	0111100011*	2	4	123	0010000011	−4	4	1001111100*	2	4	209	1110110000	0	4	0101001111*	2	6
38	0111100010	0	4	0111100010	0	4	124	1010000001*	−4	4	0001111110	2	2	210	1000011011	0	4	1000011011	0	4
39	0100011101	0	6	0100011101	0	6	125	0010000001	−6	4	1001111110*	4	4	211	1011100100	0	6	1011100100	0	6
40	0100011000*	−4	4	1111100111	6	2	126	0010000000	−8	2	1001111111*	6	2	212	1000011001*	−2	4	0011100110	0	4
41	1100011000	−2	4	0111100111*	4	4	127	1010000000*	−6	4	0001111111	4	2	213	0000011001	−4	4	1011100110*	2	6
42	1100011001	0	4	0111100110*	2	4	128	0110000000*	−6	2	1101111111	8	2	214	0000011000	−6	2	1011100111*	4	4
43	0100011001*	−2	6	1111100110	4	4	129	1110000000	−4	2	0101111111*	6	4	215	1000011000*	−4	4	0011100111	2	4
44	0111100100	0	4	0111100100	0	4	130	1110000001	−2	2	0101111110*	4	4	216	1000011101	0	4	1000011101	0	4
45	0100011011	0	6	0100011011	0	6	131	0110000001*	−4	4	1101111110	6	4	217	1011100010	0	6	1011100010	0	6
46	0010110000	−4	4	1001001111*	2	4	132	1110000011	0	2	0101111100*	2	4	218	0000011100	−4	2	1011100011*	2	4
47	1010110000*	−2	6	0001001111	0	4	133	0110000011*	−2	4	1101111100	4	4	219	1000011100*	−2	4	0011100011	0	4
48	0100010000*	−6	4	1111101111	8	2	134	0110000010*	−4	4	1101111101	6	4	220	1011100001	0	4	1011100001	0	4
49	1100010000	−4	4	0111101111*	6	4	135	1110000010	−2	4	0101111101*	4	6	221	1000011110	0	4	1000011110	0	4
50	1100010001	−2	4	0111101110*	4	4	136	0101111000	0	4	0101111000	0	4	222	0011100000	−4	2	1000011111*	2	2
51	0100010001*	−4	6	1111101110	6	4	137	0110000111	0	4	0110000111	0	4	223	1011100000*	−2	4	0000011111	0	2
52	1100010011	0	4	0111101100*	2	4	138	0110000110*	−2	4	1101111001	4	4	224	0110100000*	−4	4	1101011111	6	4
53	0100010011*	−2	6	1111101100	4	4	139	1110000110	0	4	0101111001*	2	6	225	1110100000	−2	4	0101011111*	4	6
54	1001000111	0	4	1001000111	0	4	140	0110000100*	−4	4	1101111011	6	4	226	1000001011*	−2	4	0011110100	0	4
55	1010111000	0	6	1010111000	0	6	141	1110000100	−2	4	0101111011*	4	6	227	0000001011	−4	4	1011110100*	2	6
56	0111101000	0	4	0111101000	0	4	142	0011010000	−4	4	1000101111*	2	4	228	1000001001*	−4	4	0011110110	2	4
57	0100010111	0	6	0100010111	0	6	143	1011010000*	−2	6	0000101111	0	4	229	0000001001	−6	4	1011110110*	4	6
58	1001000111*	−2	4	0010111100	0	4	144	0101110000*	−2	4	1110001111	4	2	230	0000001000	−8	2	1011110111*	6	4
59	0001000011	−4	4	1010111100*	2	6	145	1101110000	0	4	0110001111*	2	4	231	1000001000*	−6	4	0011110111	4	4
60	1001000001*	−4	4	0010111110	2	4	146	0110001110	0	4	0110001110	0	4	232	1000001101*	−2	4	0011110010	0	4
61	0001000001	−6	4	1010111110*	4	6	147	0101110001	0	6	0101110001	0	6	233	0000001101	−4	4	1011110010*	2	6
62	0001000000	−8	2	1010111111*	6	4	148	0110001100*	−2	4	1101110011	4	4	234	0000001100	−6	2	1011110011*	4	4
63	1001000000*	−6	4	0010111111	4	4	149	1110001100	0	4	0101110011*	2	6	235	1000001100*	−4	4	0011110011	2	4
64	0111000000*	−4	2	1100111111	6	2	150	1000100111	0	4	1000100111	0	4	236	0000001110	−4	2	1011110001*	2	4
65	1111000000	−2	2	0100111111*	4	4	151	1011011000	0	6	1011011000	0	6	237	1000001110*	−2	4	0011110001	0	4
66	1111000001	0	2	0100111110*	2	4	152	0110001000*	−4	4	1101110111	6	4	238	1000001111	0	2	1000001111	0	2
67	0111000001*	−2	4	1100111110	4	4	153	1110001000	−2	4	0101110111*	4	6	239	1011110000	0	4	1011110000	0	4
68	0100111100	0	4	0100111100	0	4	154	1000100011*	−2	4	0011011100	0	4	240	1000000101*	−4	4	0011111010	2	4
69	0111000011	0	4	0111000011	0	4	155	0000100011	−4	4	1011011100*	2	6	241	0000000101	−6	4	1011111010*	4	6
70	0111000010*	−2	4	1100111101	4	4	156	1000100001*	−4	4	0011011110	2	4	242	0000000100	−8	2	1011111011*	6	4
71	1111000010	0	4	0100111101*	2	6	157	0000100001	−6	4	1011011110*	4	6	243	1000000100*	−6	4	0011111011	4	4
72	0100111000*	−2	4	1111000111	4	2	158	0000100000	−8	2	1011011111*	6	4	244	0000000110	−6	2	1011111001*	4	4
73	1100111000	0	4	0111000111*	2	4	159	1000100000*	−6	4	0011011111	4	4	245	1000000110*	−4	4	0011111001	2	4
74	0111000110	0	4	0111000110	0	4	160	0101100000*	−4	4	1110011111	6	2	246	1000000111*	−2	2	0011111000	0	2
75	0100111001	0	6	0100111001	0	6	161	1101100000	−2	4	0110011111*	4	4	247	0000000111	−4	2	1011111000*	2	4
76	0111000100*	−2	4	1100111011	4	4	162	1101100001	0	4	0110011110*	2	4	248	0000000010	−8	2	1011111101*	6	4
77	1111000100	0	4	0100111011*	2	6	163	0101100001*	−2	6	1110011110	4	4	249	1000000010*	−6	4	0011111101	4	4
78	0010010000	−6	4	1001101111*	4	4	164	0110011100	0	4	0110011100	0	4	250	1000000011*	−4	2	0011111100	2	2
79	1010010000*	−4	6	0001101111	2	4	165	0101100011	0	6	0101100011	0	6	251	0000000011	−6	2	1011111100*	4	4
80	0100110000*	−4	4	1111001111	6	2	166	0011001000	−4	4	1000110111*	2	4	252	1000000001*	−6	2	0011111110	4	2
81	1100110000	−2	4	0111001111*	4	4	167	1011001000*	−2	6	0000110111	0	4	253	0000000001	−8	2	1011111110*	6	4
82	1100110001	0	4	0111001110*	2	4	168	0110011000*	−2	4	1101100111	4	4	254	0000000000	−10	0	1011111111*	8	2
83	0100110001*	−2	6	1111001110	4	4	169	1110011000	0	4	0101100111*	2	6	255	1000000000*	−8	2	0011111111	6	2
84	0111001100	0	4	0111001100	0	4	170	1000110011	0	4	1000110011	0	4							
85	0100110011	0	6	0100110011	0	6	171	1011001100	0	6	1011001100	0	6							

Table A.2. The TMDS characters (big endian) used to encode brightness given a stream with excess one bits (positive), or zero bits (negative), with the imbalance (I) and circular transition count (T). The character used when the stream is balanced is indicated by a *.

Appendix B

Test card

For the experiments I compiled some basic tests into a single test card (Figure B.1). It can be displayed on the target VDU during eavesdropping to perform basic analysis of eavesdropped emanations, and allow comparison of results across devices in a consistent manner. To prevent introducing variation between resolutions, all features of the test image are composed within a centred 640×480 region. I selected this size to match the VGA resolution supported by all VESA SMT-compliant devices. The corners of the image are marked to allow easy location of the content frame edges when observing it embedded within a blanking interval. The markers are coloured grey and the edge pixels coloured red. Two lines running from corner to corner and a single vertical line help detect image skew from incorrect parameters. The bottom and right edges are rule marked every 10 pixels with a longer tick mark every 100 pixels to act as reference points. The background is set to a brightness 16 greyscale to minimise transitions; the corresponding TMDS bit pattern is identical to a delayed version of the pixel clock (Table A.2).

To provide a reference for the colour palette, I drew a colour wheel of hues and saturations taken from the HSV colour space with a constant value $C_{\mathrm{V}} = 1$. For greyscale values, I render a block of each intensity in the range 0 to 255. The intensities increase left-to-right, top-to-bottom. This acts as a reference for how various bit patterns will appear when cross-referenced with Table A.2. To observe the difference between emanations from different transmission lines I render blocks of colour: red, green, blue, black, and white, overlaid with foreground text of each colour. Both the foreground and background have full brightness. When there is no difference in impedances in the transmission lines the text and background should be indistinguishable in the eavesdropped raster as the same information emanated, just from different wires. In practice, the received voltage levels $\boldsymbol{v}$ will be a mixture of the red $\boldsymbol{r}$, green $\boldsymbol{g}$, and blue $\boldsymbol{b}$ colour emanations:

$$v(t) = \alpha \cdot r(t) + \beta \cdot g(t) + \gamma \cdot b(t)\,, \tag{B.1}$$

scaled by some factors α, β, and γ. The colour triangle mixes the three colour channels

in quantities that result in constant brightness across the feature such that:

$$r[i] + g[i] + b[i] = 255 \tag{B.2}$$

for any pixel index i within the feature. This will assist in identifying any biases in emanation levels: if the wires exhibit significantly unequal level of emanation, we will see a brightness skew in the triangle when eavesdropped (Figure 2.12). To measure colour mixing, I generate three 2D arrays of uniformly-distributed random brightness values, rendered as three squares of colour. A fourth square displays a superposition of all three random blocks. To observe rapid brightness changes I include red, green, blue, and white gradient strips fading from full intensity (255) to none (0). I include two copies in orthogonal directions to observe if and how direction affects gradient rendering. Next to each intensity level I raster its value as a bit pattern, where 1 and 0 are represented by white and black respectively. This provides an area of high frequency with a unique pattern, which can act as a label for the gradient. The zone plate contains low frequencies toward the centre, which increase in frequency proportional to the radius, allowing us to observe aliasing effects. They are most commonly used for analogue VDUs. A chequerboard pattern with alternating black and white pixels provides a high-frequency component. As readable text is a common goal of eavesdropping, I used the Pango and Cairo libraries to render size 8 TrueType Serif font under different rendering conditions. Anti-aliasing of text can be performed in two ways: single colour anti-aliasing, which uses pixel brightness values, and the more advanced sub-pixel rendering, which utilises the separate sub-pixels used for each of the RGB outputs on the physical LCD. I repeat these with and without hinting: a method for interpolating pixels in a font to provide clearer text. Lines are prefixed with a lower case h to show no hinting has been applied, or a H to indicate hinting; and N, G, and S to indicate no anti-aliasing, single colour anti-aliasing, and sub-pixel anti-aliasing respectively. At the bottom, I render some text without anti-aliasing or hinting, varying the colour of the lines using red, green, and blue, along with a high-contrast grey.

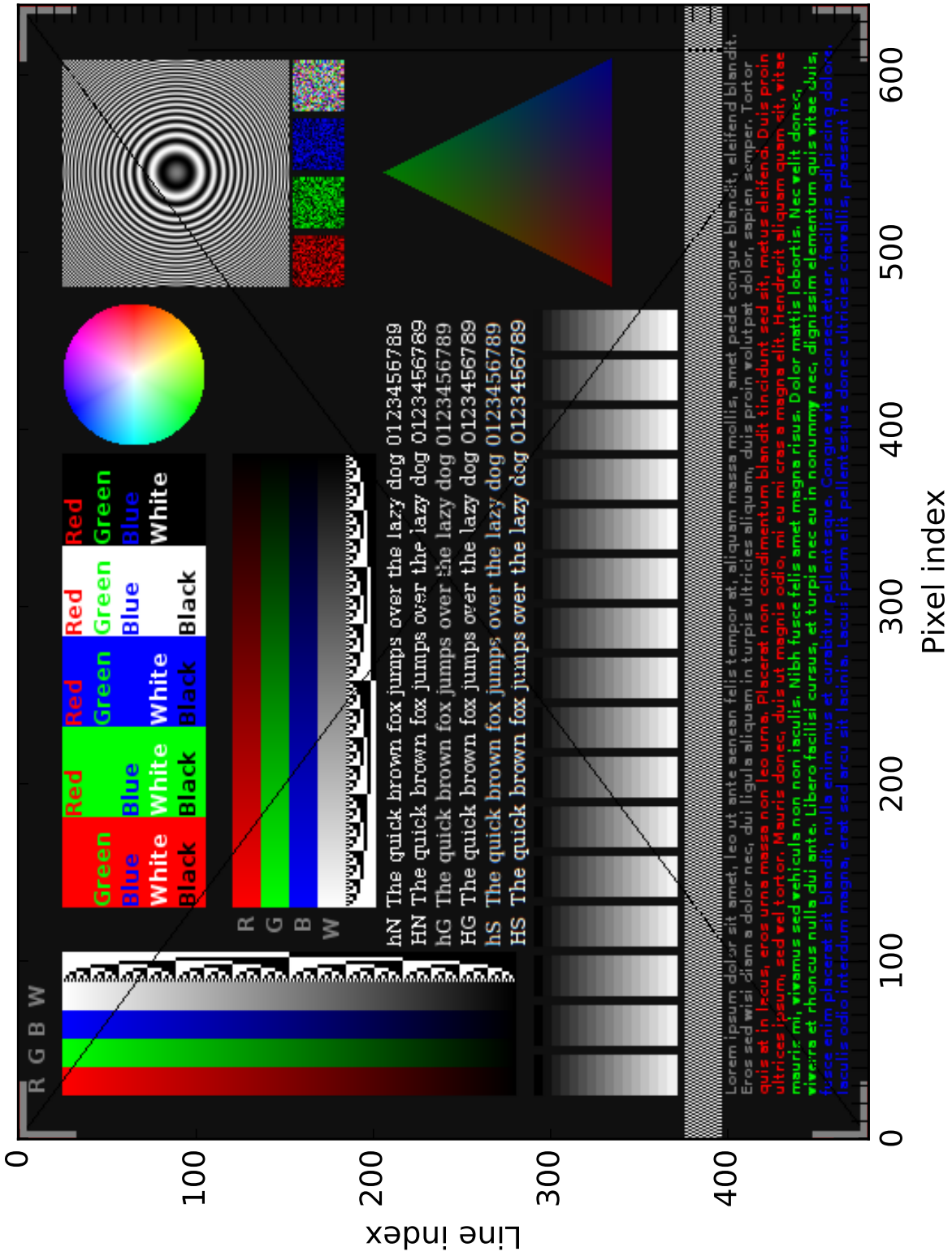


Figure B.1. Test card at 640 × 480