Coláiste na hOllscoile Corcaigh, Éire
University College Cork, Ireland

PRELIMINARY PROJECT REPORT

YANN DONNELLY

Supervisor: Dr. Colin Murphy
School of Engineering
University College Cork

December 2013 – version 1.0

DECLARATION

This report was written entirely by the author, except where stated otherwise. The source of any material not created by the author has been clearly referenced. The work described in this report was conducted by the author, except where stated otherwise.

Yann Donnelly, December 1, 2013

ABSTRACT

This report provides a preliminary summary of a final year project currently being undertaken by the author. The project goals and background are described and an overview is given of the work undertaken to date. A plan of action for the next half of the allocated time period is given. The project has succeeding in demonstrating sub-optimal, correctable receiver performance in the presence of synchronisation errors and the performance in the more general case of a fading channel are to be studied next. The report concludes with a brief discussion of the ethical implications of this research.

CONTENTS

ACRONYMS

AWGN Additive White Gaussian Noise

BPSK Binary Phase Shift Keying

EGC Equal Gain Combining

ISI Inter-Symbol Interference

MAP Maximum A Posteriori detector

ML Maximum-Likelihood detector

PAM Pulse Amplitude Modulation

PDF Probability Density Function

PSK Phase Shift Keying

TP Teaching Period

1 SUMMARY OF ACHIEVEMENTS TO DATE

- The author has familiarised himself with relevant communications theory, including modulation schemes, probability theory and optimum detector design.
- The author has become acquainted with the Mathematica programming environment and used it to program a number of mathematical simulations describing communications problems.
- The author has examined the effects of receiver timing error on BPSK and 4-PAM signalling schemes using root raised cosine filters, and demonstrated sub-optimal detection performance with 4-PAM using standard detector designs and assuming a memoryless communications channel with additive white Gaussian noise.
- The author has characterised the change of optimum decision region boundaries in the presence of non-deterministic timing errors following a known probabilistic model.
- A clear plan of work for the first half of Teaching Period 2 has been laid out, and is expected to centre around the effects of timing error offset in the presence of Rayleigh fading.

Part I

PROJECT DESCRIPTION

A description of the project, including an explanation of the underlying theory, followed by a summary of the work undertaken to date.

2

INTRODUCTION AND BACKGROUND TO THE PROJECT

This project seeks to examine the effects of receiver timing error in radio communications systems. Understanding how radio system performance decreases in sub-optimal conditions is key to assessing a system's performance and designing more robust and better performing radio systems. The principle goal is to be able to characterise the effects of timing error such that a receiver may be optimised to account for it.

A typical radio communications system consists of a transmitter, a receiver and a communications channel, that may contain any number of non-idealities. For this reason, if a transmitter sends a particular symbol down a real communications channel, the received symbol may be substantially different to the sent symbol. Therefore, careful attention is required to the design of the receiver symbol detector.

One could imagine a binary transmission system that sends one of two possible signals: a $1V_{RMS}$ wave if a '0' is to be sent, and a $3V_{RMS}$ if a '1' is to be sent. After being distorted by the communications channel, the receiver could in theory see a signal of any amplitude, and must make a decision as to which amplitude was originally sent. It would be helpful to know the probability density function (PDF) of the received signal, ie. the probability of receiving a signal amplitude if a known amplitude was sent. If one assumes the communications channel is memoryless and distorts the signal by adding zero-mean white Gaussian noise, the received signal PDF will be a Gaussian distribution centred on the sent signal amplitude, as shown below. If both symbols are equiprobable, their PDF's will be of equal height, and therefore shifted copies of each other.

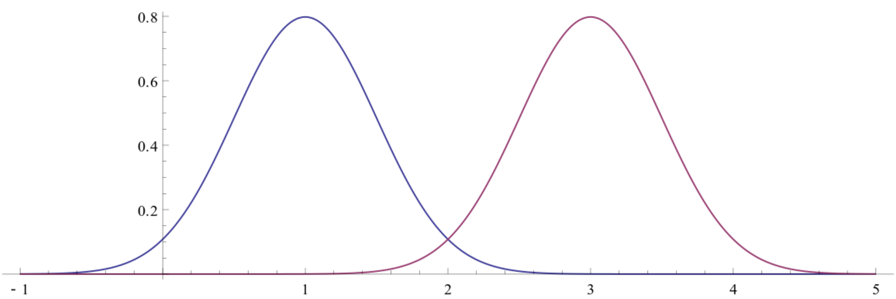

Figure 1: Gaussian noise corrupts a sent signal, resulting in a probability density function for each possible sent symbol

A *maximum-likelihood (ML) detector* seeks to minimise the probability of error by always picking whichever signal was most likely to be

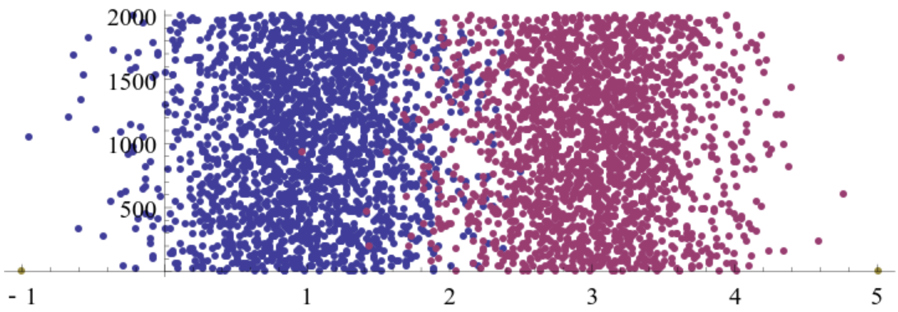

Figure 2: As an example, 2000 of each symbol has been send down the communications channel. The received values distorted by noise are plotted above. Note the overlap in received values corresponding to both symbols: it is impossible to detect the signal with 100% accuracy.

sent, given the received signal. Therefore, the threshold between picking one value or the other will be where both symbols are equally likely to be sent, the point of intersection of both PDF's. Since the Gaussian distribution is symmetric about its mean, in this case the threshold (or *Decision Region Boundary*) is exactly midway between both sent amplitudes. Intuitively this makes sense: it says that in the simplest case if one receives a particular signal, one should assume whichever possible sent signal is nearest. However in more complicated cases picking the point of intersection of both PDF's is a more general solution.

One issue that complicates detection is *inter-symbol interference* (ISI). It is possible for signals representing symbols in the future or past to bleed into the current symbol clock period, distorting the signal further. For this reason transmitters and receivers are designed such that they apply a *raised cosine function* that is fully transparent at exactly the symbol's sample time, and blocks completely at every other symbol's sample time, ie at every interval of $T_{\mathtt{clk}}$ for $T \neq 0$. If both receiver and transmitter are perfectly synchronised, this ensures that the receiver will only see signals corresponding to the current transmitted symbol, after distortion via the communications channel.

If the receiver and the transmitter are poorly synchronised, however, the root cosine filter response is shifted in time with respect to the receiver and looses some of its essential property. Current literature has considered the effects of a Tikhonov-distributed timing offset in a Nagakami-*m* channel with rectangular signalling and diversity combining [2], and a uniformly distributed timing offset in a Rayleigh fading channel [6]. The effects of timing offset in the BPSK and QAM cases have been studied in [3] & [4], and [5], respectively However, the effects of a poorly synchronised receiver on detection performance in the root-cosine filter case has not to date been published. This project seeks to examine this matter further, and ascertain whether design decisions can be made to mitigate its effects.

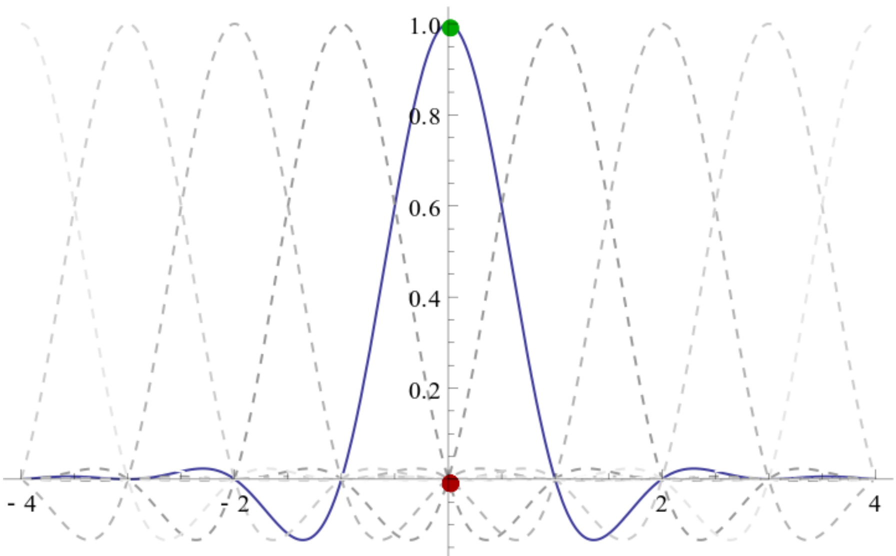

Figure 3: A filter with a root raised cosine function is often used, as it evaluates to 1 at the current sample time and 0 at all other sample times

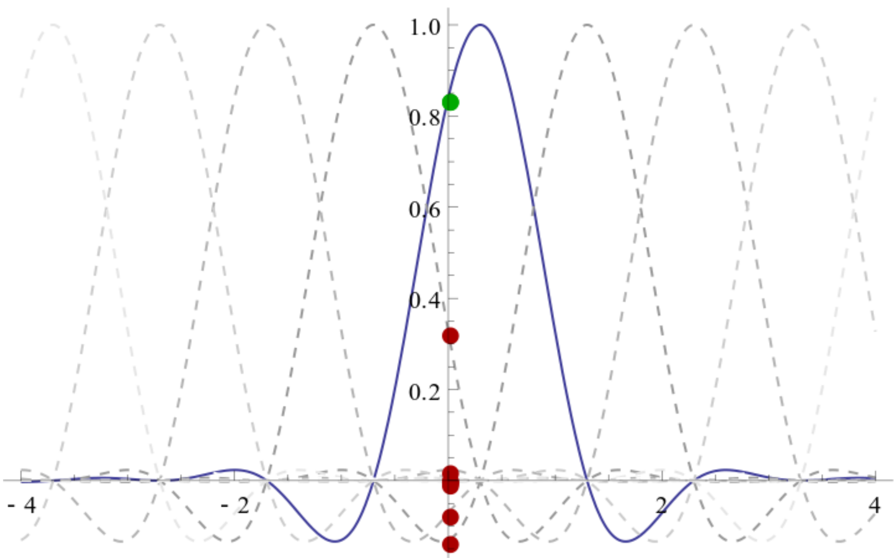

Figure 4: If a timing offset is added to the root raised cosine it no longer evaluates to 0 or 1 at the sampling time. This results in reduced receiver performance when the receiver and transmitter are not properly synchronised.

3 DESCRIPTION OF THE PROJECT

The author started by researching the underlying theory. [8] was used as a reference textbook, and explains probability theory and optimum receiver design. The concepts of representing non-deterministic signals using probabilistic models, maximum-likelihood (ML) and maximum a posteriori (MAP) receivers and designing optimum receivers given a known probabilistic model for the receiver, were studied in depth. In addition, a method for estimating received signal PDF's with a Gram-Charlier PDF developed by UCC graduate David McCarthy in [1] was studied. Finally, the Mathematica language had to be learnt from scratch, with emphasis on statistics and plot generation.

In order to familiarise myself further with these concepts, I built a simple model for a two-symbol BPSK communications system in Mathematica. I extended this to the 4-symbol 4-PAM system. With some help from David I also implemented the Gram-Charlier distribution. Using these programs as a base I then wrote a simulation that can produce a PDF of the receiver input with a known timing offset. Using this, I was able to examine how the sent symbol is distorted by the channel and timing error, and found that increasing amounts of offset reduced the effective amplitude of the signal while increasing noise due to ISI. Additionally, it was found that the optimum decision region boundary was decreased by a factor equal to the value of the root raised cosine function at the timing offset.

My attention at this stage turned to increasing the speed of the simulations, and much work was done studying Mathematica's potential for parallelisation. I was able to acquire remote access to a number of Unix machines, and ported the simulations to run across multiple machines, paying attention to taking advantage of the dual-core processors and making the code resilient to premature termination (due to power outages, illiterate students etc.).

A more realistic model was created by assuming that the timing error is not a fixed value, but could be described by a Tikhonov distribution. Two approaches were taken: an initial, quick approach was to calculated the optimum decision region boundary of each timing error, and average this over the probability of each timing error occurring. This assumes that multiple possible optimum decision region boundary locations could be averaged to give the overall optimum decision region boundary location, which I learnt was not generally the case, and this was later proven analytically by David. I therefore tried a second, more realistic approach, where for each simulated transmission a new timing offset was chosen from the Tikhonov distribution. An optimum decision region boundary was then determined for multiple Tikhonov distribution widths (or variances). It was found that the optimum decision region boundary decreased as the variance of the Tikhonov distribution increased.

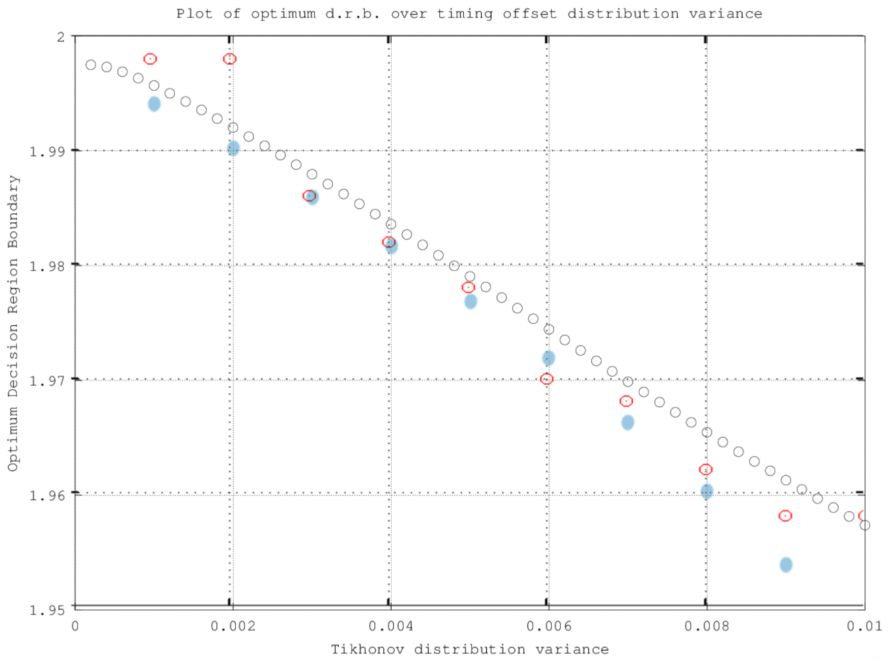

Figure 5: Optimum Decision Region Boundary for various timing error probability distributions. Black: estimate found by averaging fixed timing errors over Tikhonov distribution; Blue: analytical result determined by Dave; Red: simulation results.

To summarise, my work in TP1 helped introduce me to radio communications systems, and concepts of signal corruption and optimum detectors. I learnt how to translate signal transmission into mathematical simulations, and how to extract meaningful conclusions from the output. Using the theory and simulations developed during this time I was able to demonstrate a changing optimum decision region boundary in the presence of a non-deterministic timing offset using 4-PAM signalling. It is believed this behaviour can be generalised to L-PAM signalling. This will serve as a foundation from which to explore the effects of a non-deterministic timing offset in more general cases.

By L-PAM *I mean PAM signalling with an arbitrary number L of possible symbols. L=4 reduces to 4-PAM.*

Part II

GOING FORWARD

A plan of work for Teaching Period 2, and a brief discussion of ethical issues pertaining to the project.

4 PLAN FOR TEACHING PERIOD 2

In TP2, the author aims to extend the scope of the project to address the issue of timing error offset in communications channels subject to fading effects such as in urban areas or for very long-range communications. In either case, line-of-sight communication is very weak or non-existent, and the signal is scattered and received as a combination of multiple "bounced" signals with different propagation delays and amplitudes. In order to improve reception in this case, the signals from multiple antennas are combined before being presented to the symbol detector. I will model these using a Rayleigh distribution, and assuming an Equal Gain Combining (EGC or additive) receiver, determine the effects of timing error described above on this type of model. the decision has been made to stick to L-PAM signalling, as PSK signalling formats rely solely on phase detection and therefore the optimum decision region boundaries are unaffected by amplitude changes.

Once a description of the optimum detector in the presence of Rayleigh fading has been formulated, the next goal will be to compare it to the optimum detector described in [2], which assumes perfect synchronisation (ie. no timing offset).

A project plan is presented here as a guide to how work is expected to proceed in TP2. This plan is to be considered a mere guide, as no project plan can be accepted as gospel truth (Hopfstadter's Law [11], among others). In order to account for this timescales have been made purposefully conservative. This plan purposely only covers up to mid-February, as the results of the work below will dictate the direction to take the project in during the second half of TP2, therefore the author is unable to predict with reasonable certainty what work will be carried out.

Description	Time Allowed	Start Date	Goals
Review of Rayleigh fading	1 week	6 Jan	• Develop an understanding of Rayleigh fading. • Have built a basic Mathematica model of Rayleigh fading assuming perfect synchronisation.
Implement Rayleigh fading model with timing error	1 week	13 Jan	• Extend previous model to account for timing errors following a Tikhonov distribution model. • Evaluate optimum decision region boundary in the presence of timing error offsets and Rayleigh fading.
Additional simulation time	1 week	20 Jan	• Characterise the effect of non-deterministic timing error on the optimum decision region boundary for L-PAM signalling in the presence of Rayleigh fading.
Compare described receiver to optimum receiver described in literature	2 weeks	27 Jan	• Provide a detailed comparison of the optimum described above, to the optimum receiver described in [2], paying particular attention to performance in the presence of non-deterministic timing error.

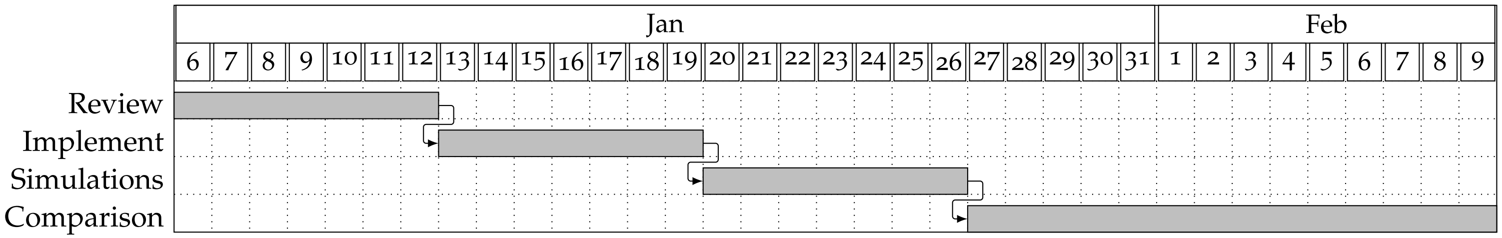

Figure 6: Gantt Chart

5 DISCUSSION OF ETHICAL ISSUES

In communications systems, as in any area seeing considerable technological developments, the question of whether these developments are ethically sound naturally arises. The advent and spread of radio throughout the 20th century brought about an era of increased connectedness and information transfer, as information could be rapidly spread across large distances with little effort. The advent of the internet in 1969 and the mobile phone boom of the 90s accelerated these changes as almost-instant, asynchronous and on-demand information transfer became available to the general public, and current Web 2.0 developments such as social networking are but continuations of this trend. With the smart-phone industry putting internet access into the pockets of consumers, many believe that these changes, driven by advances in communications technology, have had a considerable impact on our societies. While the positives are too numerous to note, the skeptic would also readily see the disadvantages.

Engineering developments can lend themselves to applications with both positive and negative intentions, as with any scientific advancement. Radio technology developments in particular has throughout history been closely linked to the military. Improving wireless communications can play into the hands of military and terrorist leaders, as cheap and reliable wireless communications can aid the organisation and execution of manoeuvres. In addition the remote detonation of explosives using mobile phones is considered a real threat to many cities [10]. Therefore, communications research enters the same ethics discussion as other defence research.

These same developments have created privacy concerns as most citizens find themselves communicating daily through the internet and the cell phone network. At the bottom of the internet hierarchy, endpoint routers are naturally trusting, and prone to exploitation; in addition, they are generally visible to all other internet users. At the top, most communications are routed through a small number of Tier 1 networks, which have proven themselves prone to eavesdropping. As the distance over which an individual can send a piece of information to its intended recipient has increased, so has the number of individuals capable of intercepting this information. Thus as users embrace the internet and smart phones, they also find themselves increasingly exposed to spying from a number of groups with varying intentions. Even outside of the "online" world, increasing the efficiency of radio receivers can lead to the exploitation of security-sensitive short-range communications. For example, contactless payment devices rely on short range communications to transfer identification information for payment purposes. Equipped with a suitable exploit, such as described in [9], a high-accuracy receiver would increase the number of these devices in range of an attacker.

The author notes that these issues relate to the broad area of communications within which this project falls. However he does not feel there is a direct ethical concern with any of the work described within here.

Part III

APPENDIX

A
LOGBOOK

A.1 WEEK 1

A.1.1 *30/09/13 - Exploring simple case with PAM modulation*

I received the *PAM.pdf* file outlining the case where a signal is sent through a channel with AWGN and received with a timing error at the receiver. I read through the file several times to get an understanding of the underlying equations.

Leaving the Gram-Charlier series aside for the moment, I started getting to grips with Mathematica and implementing the transmission system model:

$$X = \omega_0 g_0 + \sum_{k=1}^{40} (\omega_{-k} g_k + \omega_k g_k) + \nu$$

where $g_k = g((\Delta + k)T)$, $g(t) = (u_T * h_l * u_R)(t) \times \cos(\theta)$ and ν is a zero-mean Gaussian random variate with $\sigma_\nu^2 = N_0 \varepsilon_R$.

I learnt the basics of the interface, and began inplementing the filter and channel impulse responses (I.R.). I need to double-check the definition of the Root-Raised Cosine (RRC) Filter, as the impulse response wasn't as expected.

Later, I found the correct form for the RRC [12] and double-checked it using Octave. The equation used is listed below. A plot showed that this equation is invalid at $t = \left[-\frac{T_s}{4\beta}, 0, \frac{T_s}{4\beta}\right]$, so I plan to find its limit at these points using Mathematica to obtain the complete solution.

$$h_{RRC}(t) = \frac{2\beta}{\pi\sqrt{T_s}} \frac{\cos\left[(1+\beta)\frac{\pi t}{T_s}\right] + \frac{\sin\left[(1-\beta)\frac{\pi t}{T_s}\right]}{\frac{4\beta t}{T_s}}}{1 - \left(\frac{4\beta t}{T_s}\right)^2}$$

A.1.2 *01/10/13 - Implementing Raised Cosine functions*

I implemented the function above in Mathematica, and using the `Limit` function found the value of the function at the following undetermined points:

$$h_{RRC}(t) = \begin{cases} \dfrac{4\beta + \pi(1-\beta)}{2\pi\sqrt{T_s}} & t = 0 \\ \dfrac{\beta}{2\pi\sqrt{T_s}}\left(\pi \sin\left[\frac{(1+\beta)\pi}{4\beta}\right] - 2\cos\left[\frac{(1+\beta)\pi}{4\beta}\right]\right) & t = \pm\frac{T_s}{4\beta} \\ \dfrac{2\beta}{\pi\sqrt{T_s}} \dfrac{\cos\left[(1+\beta)\frac{\pi t}{T_s}\right] + \dfrac{\sin\left[(1-\beta)\frac{\pi t}{T_s}\right]}{\frac{4\beta t}{T_s}}}{1 - \left(\frac{4\beta t}{T_s}\right)^2} & \text{otherwise} \end{cases}$$

I also implemented the Raised Cosine function for the channel function, using the impulse response below[1]. I was unable however to convolve the receiver and transmitter filter functions using the `Convolve` function, even when I limited the impulse response using a `UnitBox`.

$$h_{RC}(t) = \frac{\operatorname{sinc}\left(\frac{\pi t}{T}\right)\cos\left(\beta\frac{\pi t}{T}\right)}{1 - \left(2\beta\frac{t}{T}\right)^2}$$

I looked into Mathematica's treatment of the Gaussian distribution, and figured out how to generate random noise vectors following a Gaussian distribution, as well as how to generate a list of random binary symbols.

After discussing the convolution issue with David, he suggested that the channel should be initially modelled as ideal and therefore the overall channel and filter I.R. $g(t)$ can be defined as a Raised Cosine function, as defined above. I should therefore be ready to implement the simple ISI model tomorrow.

A.1.3 *02/10/13 - Wrapping Up the Initial PAM Model*

I pulled together the Raised Cosine function and random number generator to impiment the given simplified function for the PAM receiver output, given below. Playing around with the settings, I was able to show how the g_k function increases with the timing error. I decided to study the Mathematica environment a little more before carrying on with any programming.

$$X = \omega_0 g_0 + \sum_{k=1}^{40} (\omega_{-k} g_{-k} + \omega_k g_k) + \nu$$

A.1.4 *03/10/13 - Delving deeper into Mathematica*

I devoted some time into looking through Michael Quinlan's notebooks and better understanding the workings of the `Table` functions

1 Proakis, "Digital Communications"

and the various plotting options. Fortunately my notebook was corrupted so I was able to rewrite it and understand the model a bit more. I need to figure out what variance value the noise PDF should take on, as the noise appears to be overwhelming the timing error effects. Translating the resulting PDF's into patterns is another question that needs some thought.

A.1.5 *Week 1 Summary*

Week 1 was mostly spent becoming acquainted with Mathematica and getting a feel for the equations underlying PAM transmissions. A simple model of a PAM receiver was constructed.

A.1.6 *Goals for Week 2*

- The PAM model will need to be extended to calculate the optimum decision region boundary from the estimated PDF.
- A better setup will be required to perform large-scale simulations within an acceptable time period. We will look into applying for an account on the Boole cluster.

A.2 WEEK 2

A.2.1 *07/10/13 - Matrix manipulations*

I decided to spend another day learning about the Mathematica environment, in particular matrix manipulation and generation. I looked into the `Apply`, `Map` and `Partition` functions and wrote some examples to figure out how to convert mathematical problems to Mathematica notation using matrices. I hope to convert the code to use matrices tomorrow to hopefully simplify and speed things up.

I also implemented David's equation for properly calculating the AWGN function variance from SNR[2], from last Friday's meeting.

2 See Dave's notes

A.2.2 *08/10/13 - Fixed I.R. and Kernel Density Estimation*

The first job was to rewrite the code to make use of the simple dot operator to calculate all the ISI components[3]. With the new code I was able to carry out many more runs and get much more detailed output. In addition, when I was rewriting the code I noticed a typo in the Raised Cosine I.R. that was degrading performance in the perfectly synchronised case. With both of these changes made, I decided to use Kernel Density Estimation to see what effects the timing offset has.

Using offsets of 10^{-15}, 0.05, 0.1 & 0.15, the following values of $g_k, k \in \{-40 \cdots -1, 1 \ldots 40\}$ were calculated.

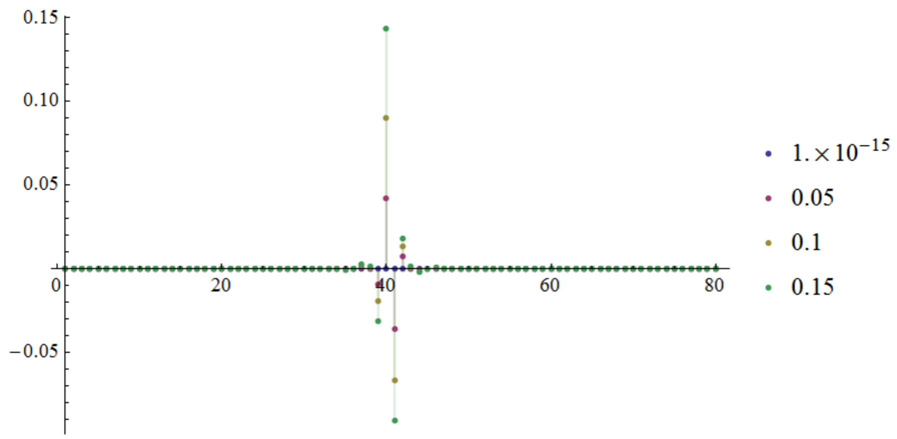

Figure 7: g_k linear plot

Using `SmoothKernelDistribution` to perform Kernel Density Estimation with 1 million points produced the following estimated PDFs for both possible transmitted values. As the timing error increases, we note that the PDF spreads out, but the mean remains steady.

A.2.3 *09/10/13 - Setting up Digital Comms Lab PC*

With Ger's help, I set up an account on `Digital Comms Lab 1` & `Digital Comms Lab 2` and got the internet working. Mathematica 8 is installed and working on both machines, we will have to consider whether

3 The ISI components are now calculated using:

$$\left[\sum_{k=0}^{k=40}\left(g_k\omega_k^j + g_{-k}\omega_{-k}^j\right)\cdots\right]^{j=\{1..m\}} = [g_{-40}\cdots g_{-1}g_1\cdots g_{40}]\bullet\begin{bmatrix}\omega_{-40}^j\\ \vdots\\ \omega_{-1}^j\\ \omega_1^j\\ \vdots\\ \omega_{40}^j\end{bmatrix}^{j=\{1..m\}}$$

where ω_k^j is the k'th ISI with the j'th timing offset.

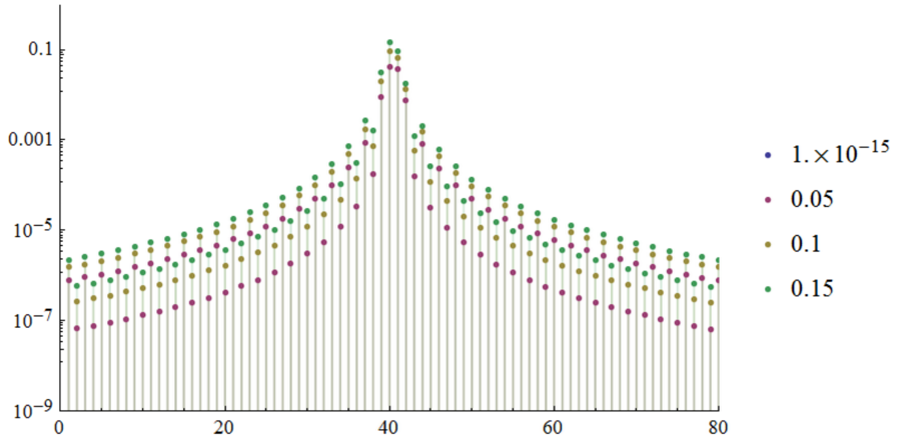

Figure 8: g_k log plot

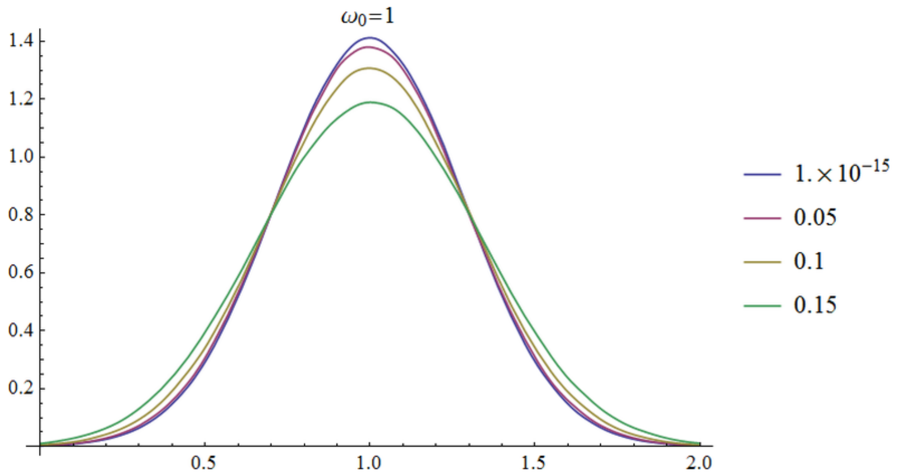

Figure 9: kernel density estimation $\omega_0 = 1$

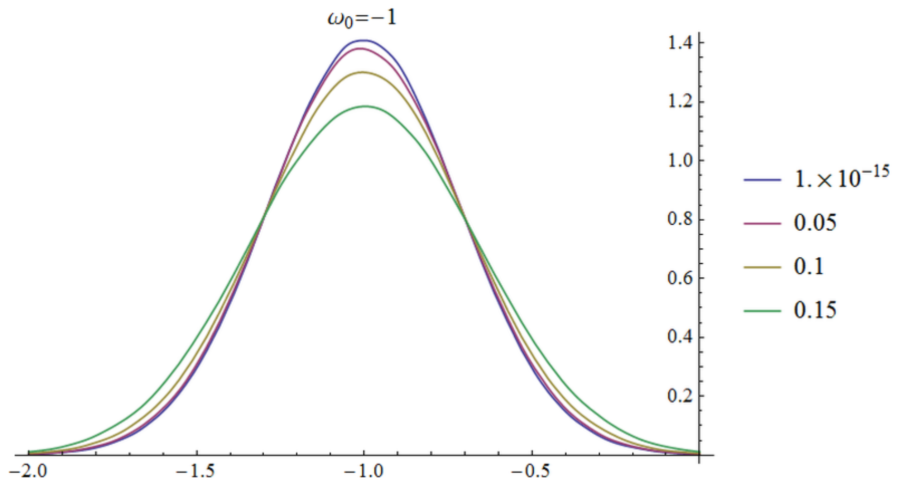

Figure 10: kernel density estimation $\omega_0 = -1$

an upgrade to Mathematica 9 would be useful or not. Git and VNC or similar have to be installed next. A request was made to the Boole cluster for access for this machine, however the email given (`bcrisupport@bcri.ucc.ie`) was invalid.

A.2.4 *10/10/13 - Probing the Elec Eng network*

After finding out the Boole cluster was no more, I used today to examine what hardware I had available to me. I got access from Ger to the public `UEPC004` server, and from there I am able to access machines on the elec eng network. I set up a *Remote Desktop Protocol* link to `Digital Comms Lab 1` through this server, allowing me to control the machine from any location. I am also able to log remotely into EDA lab machines, and run Mathematica 6 on those machines. Ger has been known to tweak machines in response to personal requests, so if asked nicely he may let me use two or three of these machines concurrently.

The GUI does not work when using `ssh` to access the EDA Lab machines, but using the command `math` to start and operate Mathematica kernels does.

Given these resources, I feel there are three ways I could continue:

- I could upgrade to the latest version of Mathematica on all machines, and set up a Mathematica cluster with `Digital Comms Lab 1` as the front end and the EDA Lab PCs as remote nodes. With this setup, all machines would act as one (as in a traditional cluster). This would be the easiest to use, but would require considerate work to set up.
- I could use the `MathLink` interface to acheive a similar, lower-level version of the former, with the EDA Lab machines as independent, remote slaves and `Digital Comms Lab 1` sending commands to these slaves and collating the replies. This setup is distributed computing with a star topology, and would be easier to setup. The downside is that the code needs to manually divide the task between each of the nodes, and needs to be well designed to minimise network delays.
- I could simply run the code in parralel on each of the machines available to me, dumping the results to text files, and collate the data at the end. This would require no setup, and code written on any machine would only require porting to another version of Mathematica. Additionally this seems like it would deal best with hiccups such as machines going down and it does not depend on a connection between the machines. The downside is there would be some overhead with collecting the results afterwards.

A.2.5 *Week 2 Summary*

I fixed the code written last week and began setting up my simulation environment.

A.2.6 *Goals for Week 3*

- Work out a setup that will allow me to carry out largeer-scale simulations.
- Adapt the previous code to run in parallel and produce useful machine-readable output.

A.3 WEEK 3

A.3.1 *14/10/13 - Running longer scripts on the EDA machines*

Today I spent some time figuring out how to build and run scripts on the EDA machines. I found that defining a module in a text file and copying the Mathematica code into the module allows the code to be called with input arguments, and writing the output to a text file and placing the module in a loop allows each pass to be recorded for later parsing. After running the code overnight, this system appears to work, and is scaleable over multiple machines. The main disadvantage is the size of these files (7.7MB per 400,000 values), so I must either figure out how to transfer them over the network or see whether reducing the precision of the output values will reduce the file sizes.

Using the `Get` and `Put` methods. The `DumpSave` method is supposed to be more efficient, but was added after Mathematica 6.

A.3.2 *15/10/13 - Reducing output size*

Given the 15GB of samples produced the night before was far too much to pull off the machine, I copied 20 million of the samples and plotted them to make sure the script had worked in practice. I then looked into how I could reduce the size of the output produced, and decided to replace the `SmoothKernelDistribution` function (which came in after Mathematica 6.0 and therefore couldn't be used on the EDA machines) with a fine-grained histogram function. This allowed me to add the probabilities generated in each sweep to those generated before and keep the output to a handful of 1kB files. I ran the simulation overnight to check it.

For the record, I could only use a fraction of them, as loading all 20 million samples crashed the machine for over an hour.

I am assuming that both the smooth kernel distribution and the histogram approach the true PDF as $N \to \infty$

A.3.3 *16/10/13 - Moving onto 4-PAM*

Checking the output from the night before, I get a similar PDF plot as with the `SmoothKernelDistribution` function. I therefore modified the code to examine all 3 decision region boundaries in a 4-PAM system and ran the simulation for 100 million samples per condition. The resulting distributions shown below show increased probability of error with timing error, as expected, but decision region boundaries in this case remain the same.

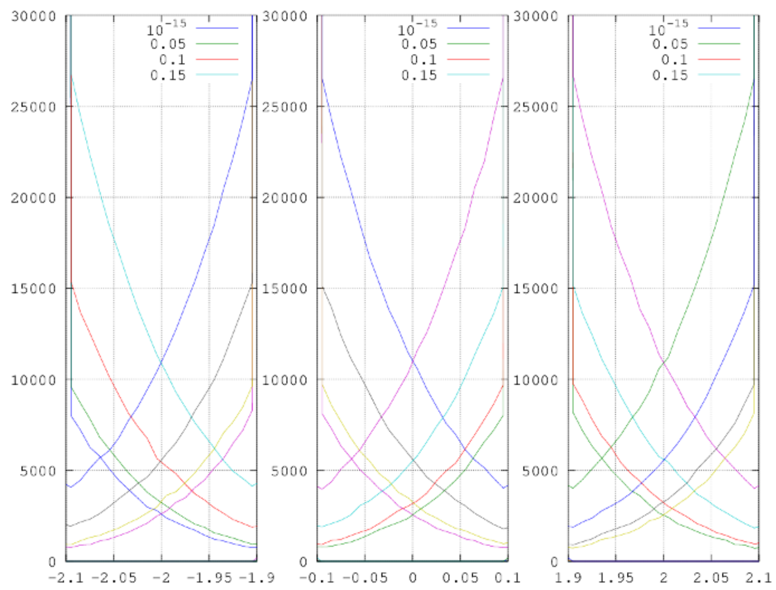

Figure 11: PDF for 4-PAM, $\omega_0 \in -3, -1, 1, 2$, 10^8 samples

I could imagine finding a value for the probability of error and moving onto PSK systems as the next steps in the process.

A.3.4 *Week 3 Summary*

Code was written that could be executed in parallel on multiple machines, and this was demonstrated in practice. The code was extended to the 4-PAM case, and showed no change in optimum decision region boundaries. Upon later consultation with Dave, it seems this is because the decision region boundaries shift due to a change in the g_0 term, and not the appearance of ISI components due to the g_k terms; the latter was believed to be the expected cause, and so the g_0 term was assumed to be 1 in the code.

A.3.5 *Goals for Week 4*

- Re-run the simulations to see if implementing the change in g_0 with timing error changes the location of the optimum decision region boundaries.

- If so, it would be interesting to see if the Gram-Charlier approximation produces the same boundary locations.

A.4 WEEK 4

A.4.1 *21/10/13 - Implementing the Gram-Charlier series*

Over the weekend, I implemented the g_0 term fix discussed in our Friday weekly meeting and re-ran the simulation, this time across two machines. Results showed that the Decision Region Boundary is displaced towards the origin as the timing offset increases.

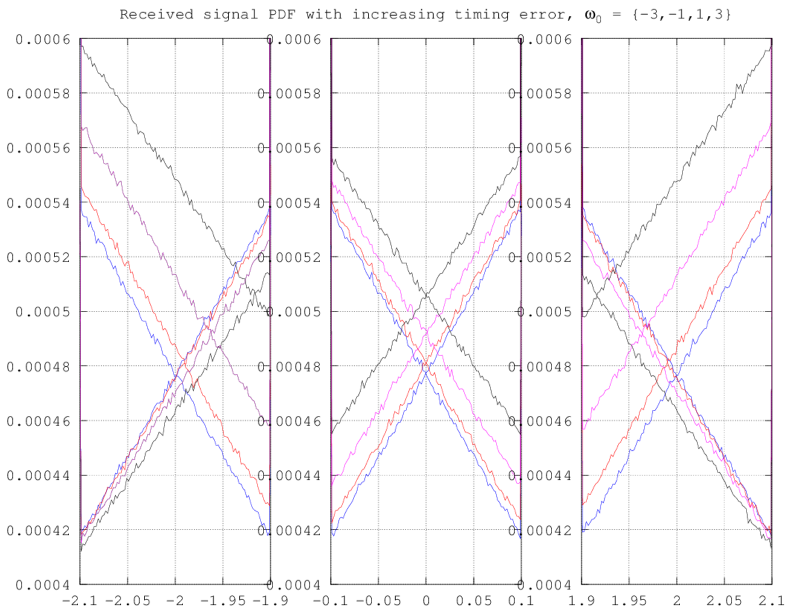

Figure 12: PDF for 4-PAM, $\omega_0 \in -3, -1, 1, 2$

I spent Monday carrying out two tasks:

1. I re-wrote Dave's Gram-Charlier equations[4] for Mathematica, and should be ready to try them out tomorrow.
2. I modified the PAM simulation with a coarser-grained histogram, but more timing offset values, in order to see how the decision variate varies with timing offset. The results should be available in the morning.

A.4.2 *22/10/13 - More Gram-Charlier series*

The simulation results showed that the decision region boundaries did decrease with timing error, however the histogram was not fine-

4 See Dave's notes

grained enough to accurately determine the exact boundary locations, so the simulation was re-run with more bins.

I fixed some bugs in my implementation of the Gram-Charlier series and was able to generate a few plots, which were very similar to those generated by the simulator, albeit with half the amplitude. A goal for tomorrow is to generate the plots with identical timing offsets to the simulation and compare both plots.

A.4.3 *23/10/13 - Proper Gram-Charlier plots*

The simulation results had been appended to the previous set of results by accident, so the whole thing had to be run again for tomorrow. On a more positive note, I noticed a missing power in my implementation of the Gram-Charlier series, and the plots are now a lot closer to those generated previously.

A.4.4 *Week 4 Summary*

I implemented the Gram-Charlier series and was able to compare the results from the simulationa dn the Gram-Charlier series. These are close, but not exact, so we will have to look closely at where the differences may be coming from.

A.4.5 *Goals for Week 5*

- I will make use of the long weekend to run some extra-long simulations and compare these to the Gram-Charlier series.

A.5 WEEK 5

A.5.1 *29/10/13 - Comparing Gram-Charlier to Simulation*

The simulations ended, and I was able to compare simulated and gram-charlier PDF plots. I extracted a rough estimate of the decision region boundaries given by both methods and compared them to the corresponding values of $2g(\Delta)$, and found very close correlation.

A.5.2 *30/10/13 - Applying the Tikhonov Distribution*

I was able to implement the Tikhonov Distribution using the equation provided in *PAMTikhonov.pdf*:

{1.99962, 1.99062, 1.96267, 1.91668, 1.85355}

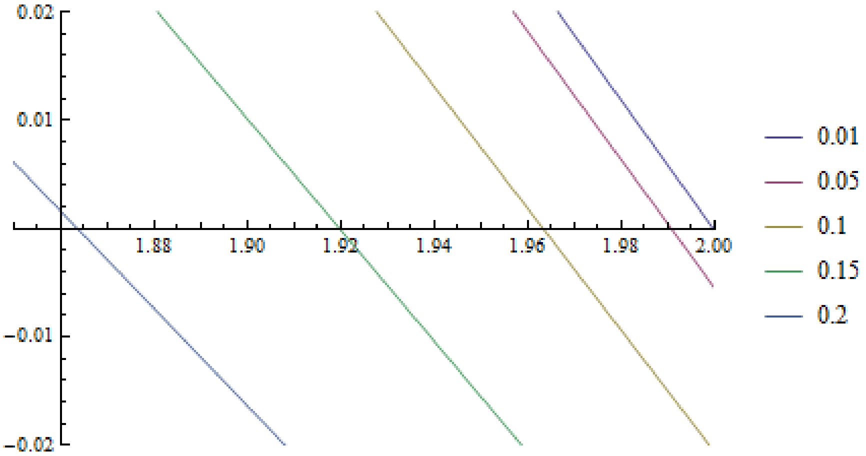

Figure 13: Gram Charlier approximation of $P(\omega_0 = 1, R) - P(\omega_0 = 3, R)$

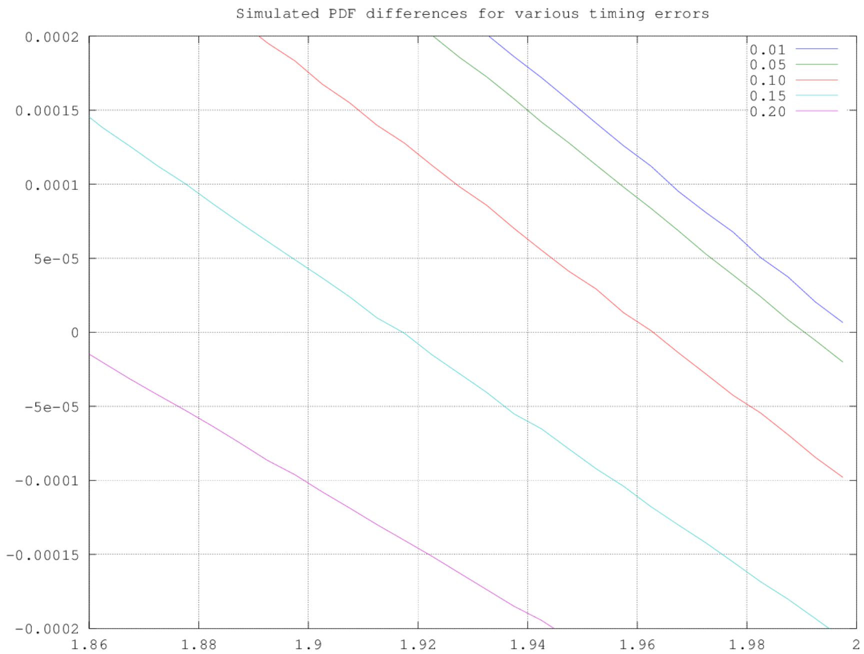

Figure 14: Simulation of $P(\omega_0 = 1, R) - P(\omega_0 = 3, R)$, N=$3 \times 10^6$

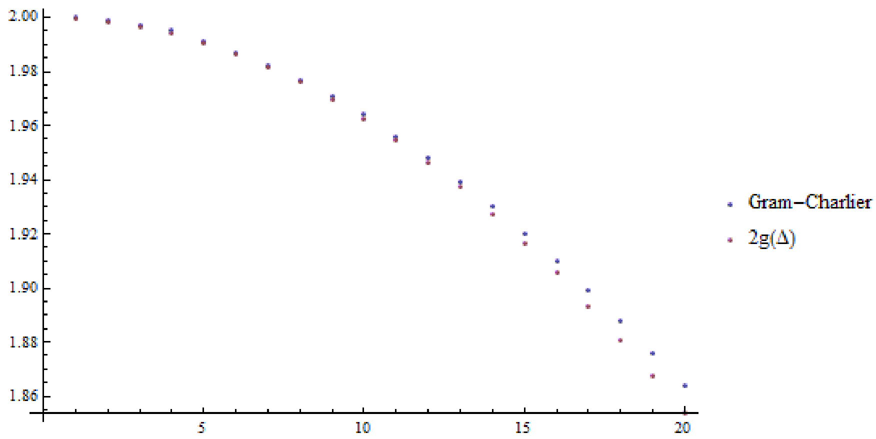

Figure 15: Comparison of Gram-Charlier Decision Region Boundaries and $2g(\Delta)$ estimation $(0.01 \geqslant \Delta \geqslant 0.2)$

$$F_\Delta(y) = \frac{\mathrm{Exp}\left[\frac{\cos(2\pi y)}{(2\pi\sigma_\Delta)^2}\right]}{I_0\left(\frac{1}{(2\pi\sigma_\Delta)^2}\right)} \text{ where } -\frac{1}{2} \leqslant y \leqslant \frac{1}{2}$$

Given these timing error probabilities and the optimum decision region boundaries for each timing error, I calculated the overall optimum decision region boundary for each timing error probability distribution using

$$B_{OPT} \sim \sum_\Delta P(\Delta) B_{OPT,\Delta}$$

It is important to note that with increasing variance, the probability density function places more weight on larger timing errors outside the range simulated, so these results are less accurate for higher variances.

A.5.3 *Week 5 Summary*

In week 5, I calculated the optimum decision region boundary for a range of timing offsets, through simulation and the Gram-Charlier approximation. I demonstrated a close correlation between these boundaries and the $2g_k$ term. A slight difference between the Gram-Charlier approximation was found to be due to a typo in its implementation. I applied the Tikhonov distribution to the calculated optimum decision region boundaries for each timing offset, in order to calculate an optimum decision region boundary for a given Tikhonov distribution of timing offsets

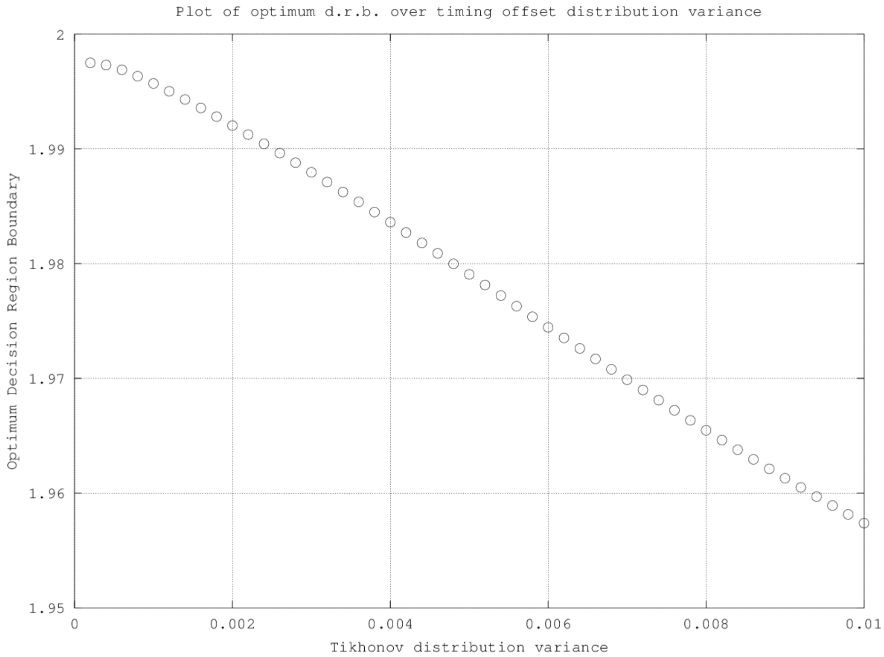

Figure 16: Optimum Decision Region Boundary for various timing error probability distributions

A.5.4 *Goals for Week 6*

- On the simulation side, a key goal for week 6 is to randomly generate timing offsets according to the Tikhonov distribution and apply these to the simulation as timing offsets, in order to verify correlation with the Gram-Charlier and $2g_k$ approximations.
- A typo in the Gram-Charlier implementation has been found and corrected, and it would be interesting to see if this approximation matches $2g_k$.

A.6 WEEK 6

A.6.1 *04/11/13 - Fixing errors*

Dave took a look at my code and spotted errors which I fixed. The fixed Gram-Charlier implementation was found to match $2g_k$ very closely. The fixed simulation was left to run overnight; unfortunately Mathematica 6.0 running on the Unix machines was unable to run it, so the number of points had to be reduced.

A.6.2 *05/11/13 - Corrected simulation results*

The produced PDFs were too inaccurate to properly calculate the zero crossing points, so the simulation will have to be run over several days.

A.6.3 *Week 6 Summary*

A simulation was constructed that generated timing error offsets according to a Tikhonov distribution of predetermined variance, and used to produce received symbol PDFs. The simulation was found to run very slowly, and could only be run on Mathematica 9. Ger has been asked whether it would be possible to upgrade the Unix machines to this version and he will look into it.

A.6.4 *Goals for Week 7*

- Continue running the simulation, trying to speed it up if at all possible.

A.7 WEEK 7

A.7.1 *11/11/13 - Returning to the UNIX machines*

The UNIX machines were upgraded to Mathematica 9 over the weekend, so I was able to port the code in order to run off these. In addition, Dave suggested that I look into paralllelising the code. Since these were dual-core machines I was able to make use of Mathematica's `ParallelTable` function to reduce run times a little. Th simulation will have to run over several days, however, as the expected deviation in optimum decision region boundary is very small.

A.7.2 *19/11/13 - Day 9 of Week 7*

After several days of running the simulations, we found the optimum decision region boundaries given by the simulations, in red, converged to roughly those predicted by averaging the optimum decision region boundary of a timing offset over the tikhonov distribution of timing offsets, in blue, given by the equation:

$$B_{OPT} \sim \sum_{\Delta} P(\Delta) B_{OPT,\Delta}$$

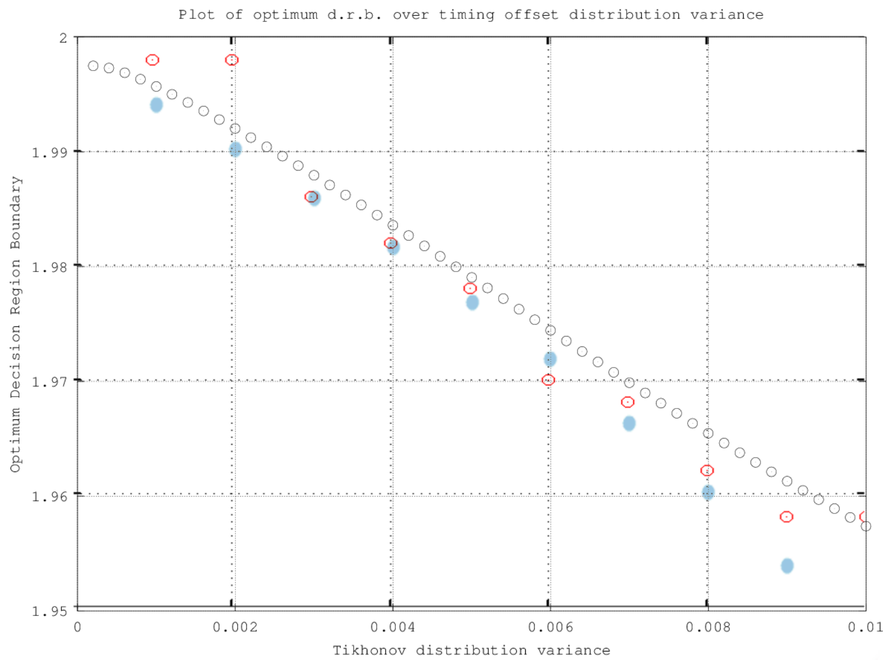

Figure 17: Optimum Decision Region Boundary for various timing error probability distributions. Black: estimate found by averaging fixed timing errors over Tikhonov distribution; Blue: analytical result determined by Dave; Red: simulation results.

A.7.3 *Week 7 Summary*

Simulations supported the theory that the optimum decision region boundary in the presence of statistically distributed receiver timing errors will decrease from the expected value. Additionally, it was shown through simulation that the new optimum decision region boundary can be approximated, assuming a known distribution of these timing errors, by averaging the optimum decision region boundary given each timing offset over the distribution of timing offsets.

A.7.4 *Goals for Week 8 onwards*

- Verify that the Gram-Charlier series provides an adequate approximation to the received symbol PDF in the presence of timing errors.
- Provide numerical values for the probability of error P_e in the presence of a distribution of timing errors.

A.8 WEEK 8

Additional simulation was carried out in order to provide a more accurate representation of the received signal PDF. Work on the preliminary report was also started.

B

EXAMPLE SOURCE CODE LISTING

The code below simulates transmitting a random array of "interference" symbols surrounding a given symbol through a memoryless communications channel with AWGN, and receiving it through a root-raised cosine filter with a random timing offset following a Tikhonov PDF. The algorithm simulates 100,000 transmissions for a single Tikhonov distribution, and is run 100 times for each of 10 Tikhonov distributions, for symbols 1 and 3.

```
l = 4;
be = 0.5;
th = 0;
de = 0.1;
nISI = 20;
snrDB = 8;
ts = 1;
npts = 100000;

(* functions *)
h[t_] := (Sinc[(Pi*t)/ts]*Cos[(Pi*be*t)/ts])/(1 - ((2*be*t)/ts)
    ^2);
g[kk_, de_] := h[ts*(de + kk)];
thikfactor[var_] := thikfactor[var] = 1/((2 \[Pi])^2 var);
thik[var_, y_] := thik[var, y] = Exp[thikfactor[var] Cos[2 \[Pi]
    y]]/BesselI[0, thikfactor[var]];

(* Algorithm (runs 'count' times) *)
f[count_, omega0_, tvar_] := Module[{c = count, w0 = omega0, tv =
     tvar},

   Print[{w0,tv,c}];
   thikdist = ProbabilityDistribution[thik[tv, y], {y, -0.5,
       0.5}];

   (* received symbols *)
   w = Join[RandomChoice[{-3, -1, 1, 3}, nISI/2], {w0},
       RandomChoice[{-3, -1, 1, 3}, nISI/2]];
   k = Range[-nISI/2, nISI/2];
   dtiming = N[RandomVariate[thikdist, {npts}], 8];
   ndtiming = Length[dtiming];
   gk = N[ParallelTable[g[a, b], {b, dtiming}, {a, k}]];
   rk = gk.w;
   snr = 10^(snrDB/20);
   var = N[Sqrt[(l^2 - 1)/(6*Log[2, l]*10^(snrDB/10))]];
   nu = RandomReal[NormalDistribution[0, var], npts];
   r = rk + nu;
```

```
   (* histogram *)
   histpts = Range[1.5, 2, 0.002];
   newprob = BinCounts[r, {histpts}];
   filename = "1/output_" <> ToString[tv] <> "_" <> ToString[w0]
       <> ".txt";
   If[c == 0, oldprob = Array[0 &, Dimensions[newprob][[1]]],
       oldprob = Get[filename]];
   prob = oldprob + newprob;
   Put[prob, filename];];

(* Loop (runs algorithm 100 times with various tikhonov variances
     (tv)) *)
Do[Do[f[ic, 3, tv], {ic, 0, 100}], {tv, 0.001, 0.01, 0.001}]
Do[Do[f[ic, 1, tv], {ic, 0, 100}], {tv, 0.001, 0.01, 0.001}]
```

C

BIBLIOGRAPHY

[1] D. D. Murphy and C. C. Murphy, "A Gram-Charlier series method for calculating general signal constellation error probabilities", *IEEE Transactions on Communications*, Vol. 60, Number 2, p. 300 - 305, 2012.

[2] Y. Yin, J. P. Fonseka and I. Korn, "Sensitivity to timing errors in EGC and MRC techniques", *IEEE Transactions on Communications*, Vol. 51, Issue 4, p. 530 - 534, 2003.

[3] M. K. Simon and M.-S. Alouini, "Simplified noisy reference loss evaluation for digital communication in the presence of slow fading and carrier phase error", *IEEE Transactions on Vehicular Technology*, Vol. 50, Issue 2, p. 480 - 486, 2001.

[4] N. A. Najib and V. K. Prabhu, "Analysis of equal-gain diversity with partially coherent fading signals", *IEEE Transactions on Vehicular Technology*, Vol. 49, Issue 3, p. 783 - 791, 2000.

[5] X Tang, M.-S. Alouini and A. Goldsmith, "Effect of channel estimation error on M-QAM BER performance in Rayleigh fading", *IEEE 49th Vehicular Technology Conference*, Vol. 2, p. 1111 - 1115, 1999.

[6] M. O. Sunay and P. J. McLane, "Probability of error for diversity combining in DS CDMA systems with synchronization errors", *European Transactions on Telecommunications*, Vol. 9, Issue 5, p. 449 - 463, 1998.

[7] M. K. Simon, "A simple evaluation of DPSK error probability performance in the presence of bit timing error", *IEEE Transactions on Communications*, Vol. 42, Issue 234, p. 263 - 267, 1994.

[8] J. Proakis and M. Salehi, *Digital Communications*, 5th ed., New York: McGraw-Hill, 2008.

[9] R. Michael, L. Joseph and S. Josef, "Applying relay attacks to Google Wallet", *5th Int. Conf. on Near Field Communications*, Zurich, pp. 1 - 6, 2013.

[10] Kopp C., *Technology of Improvised Explosive Devices*, Defence Today, Vol.6, No.3, 2007.

[11] D. R. Hopfstadter, *Gödel, Escher, Bach: an Eternel Golden Braid*, London: Penguin Books Ltd., 2000.

[12] E. Cubukcu. (2012, May). *Root Raised Cosine (RRC) Filters and Pulse Shaping in Communication Systems* [Online]. Available: `http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/` ↦ `20120008631_2012008365.pdf`

[13] D. Graffox. (2009, Sep). *IEEE Citation Reference* [Online]. Available: `http://www.ieee.org/documents/ieeecitationref.pdf`