Quiet!

Is Stock Market Noise Reduction Profitable?

BY

ADRIAN LETCHFORD BCompSci(Hons)

A THESIS SUBMITTED FOR THE DEGREE OF:

DOCTOR OF PHILOSOPHY

AT

CHARLES STURT UNIVERSITY

October 7, 2014
Contents

Certificate of Authorship vii
Acknowledgements viii
Abstract ix
List of Publications x
Mathematical & Literary Standards xi
Preface xii
Introduction xvi

1 Background 1
  1.1 Can I Make Money? 3
  1.1.1 Weak Form 5
  1.1.2 Semi-strong Form 6
  1.1.3 Strong Form 8
  1.1.4 Concluding Remarks 9
  1.2 Number Crunching 10
  1.2.1 Performance Measures 11
  1.2.2 Time Series Prediction 14
  1.2.3 Technical Analysis 18
  1.2.4 Final Remarks 21
  1.3 Noise Reduction 23
    1.3.1 Quantifying Noise & Noise Reduction 24
    1.3.2 Models 27
1.3.3 Moving Averages as Technical Analysis 38

1.4 Summary 41

2 Smoothing Stock Prices 43

2.1 Filter Purpose & Goal 45

2.2 Datasets & Data Model 47

2.3 Metrics 51

2.3.1 Smoothness 51

2.3.2 Lag 59

2.3.3 Smoothness vs Lag 62

2.3.4 Closeness of Fit 63

2.4 Filter Derivation 66

2.4.1 Data Driven Derivation 67

2.4.2 Expected Filter Derivation 69

2.5 Filter Family 78

2.5.1 Linear Gaussian Smoother (LGS) 78

2.5.2 Symmetric Linear Gaussian Smoother (SLGS) 82

2.5.3 Asymmetric Linear Gaussian Smoother (ALGS) 86

2.6 Filter Comparison 87

2.6.1 Filter Parameter Optimization 88

2.6.2 Comparison Methodology & Results 91

2.7 Conclusions 96

3 Forecasting the Smoothed Prices 98

3.1 Model 99

3.2 Measuring the Model Error 101

3.2.1 Expected Error when Forecasting the Filter 101

3.2.2 Expected Error when Forecasting Change 104

3.3 Filter Comparison 107

3.4 Alternative Predictive Algorithms 112

3.5 Conclusions 113

4 Effects of Smoothing on Profit 114

4.1 Trading System 115

4.1.1 Similarity Between the SFR and the DFR 117

4.2 Effect of Smoothing 121

4.2.1 Methodology & Results 121

4.2.2 What about Forecasting? 129
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3</td>
<td>Other Filters</td>
<td>130</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Asymmetric Filters</td>
<td>131</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Single Parameter Filters</td>
<td>133</td>
</tr>
<tr>
<td>4.4</td>
<td>Conclusions</td>
<td>135</td>
</tr>
<tr>
<td>5</td>
<td>Summary &amp; Conclusions</td>
<td>136</td>
</tr>
<tr>
<td>5.1</td>
<td>Conclusions &amp; Contributions</td>
<td>139</td>
</tr>
<tr>
<td>5.2</td>
<td>Future Research</td>
<td>140</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>All the datasets</td>
<td>49</td>
</tr>
<tr>
<td>2.2</td>
<td>Testing properties of each time series</td>
<td>50</td>
</tr>
</tbody>
</table>
List of Figures

1 Example of filtering stock prices. ........................................ xvii
1.1 Head and Shoulders Example. ................................. 20
1.2 Dual Filter Rule Example. ........................................ 39

2.1 Example of smooth filters. ......................................... 46
2.2 Smoothness ratio vs smoothness index. ..................... 54
2.3 Correlation between the SI and its expected value. ....... 59
2.4 Correlation between lag and its expected value. .......... 62
2.5 Smoothness vs Lag of three basic filters. ................. 63
2.6 Correlation of fit and its expected value. .................. 66
2.7 Filter expected fit vs expected lag and smoothness index. 66
2.8 Correlation of the filter weights and their expected value. 77
2.9 Filter weights with their expected value. ................... 78
2.10 Differencing parameter. ......................................... 79
2.11 The expected SI and fitness surfaces of the LGS. ........ 80
2.12 Examples of the LGS weights. ................................. 81
2.13 The expected SI surfaces of the SLGS. ...................... 83
2.14 The expected fit surfaces for the SLGS. .................... 83
2.15 Examples of the SLGS weights. ............................... 85
2.16 SI Surface of the ALGS. ......................................... 86
2.17 The lag and fitness surfaces of the ALGS. ................ 87
2.18 Examples of the ALGS weights. ............................... 87
2.19 Smoothness Index vs Lag. ....................................... 91
2.20 Expected lags with the highest expected SI. ............. 92
2.21 Data-driven lags with the highest SI. ...................... 92
2.22 Lag vs the Smoothness Index. ............................... 93
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure No.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.23</td>
<td>Expected SIs with the lowest expected lag.</td>
<td>94</td>
</tr>
<tr>
<td>2.24</td>
<td>Data-driven SIs with lowest lag.</td>
<td>94</td>
</tr>
<tr>
<td>2.25</td>
<td>Fit vs Lag.</td>
<td>95</td>
</tr>
<tr>
<td>2.26</td>
<td>Expected lags with the lowest expected fit.</td>
<td>96</td>
</tr>
<tr>
<td>2.27</td>
<td>Data-driven lags with lowest fitness.</td>
<td>96</td>
</tr>
<tr>
<td>3.1</td>
<td>Forecasting Demo.</td>
<td>100</td>
</tr>
<tr>
<td>3.2</td>
<td>Correlation of the forecast error and its expected value.</td>
<td>104</td>
</tr>
<tr>
<td>3.3</td>
<td>Accuracy of the forecast error of the rate of change and its expected value.</td>
<td>106</td>
</tr>
<tr>
<td>3.4</td>
<td>Expected Errors for each Filter and Horizon.</td>
<td>108</td>
</tr>
<tr>
<td>3.5</td>
<td>Best Filter for each Horizon.</td>
<td>109</td>
</tr>
<tr>
<td>3.6</td>
<td>Expected Errors when Extrapolating out the Filter Lag.</td>
<td>110</td>
</tr>
<tr>
<td>3.7</td>
<td>Expected Errors when Extrapolating out the Rate of Change vs Lag.</td>
<td>111</td>
</tr>
<tr>
<td>4.1</td>
<td>Single Moving Average System.</td>
<td>116</td>
</tr>
<tr>
<td>4.2</td>
<td>Dual Filter Rule Example.</td>
<td>117</td>
</tr>
<tr>
<td>4.3</td>
<td>Step 1 Converting Two Filters into One.</td>
<td>118</td>
</tr>
<tr>
<td>4.4</td>
<td>Step 2 Converting Two Filters into One.</td>
<td>119</td>
</tr>
<tr>
<td>4.5</td>
<td>Dual MA Converted to Single MA.</td>
<td>120</td>
</tr>
<tr>
<td>4.6</td>
<td>3D Plot of Sharpe Ratio with Fees.</td>
<td>124</td>
</tr>
<tr>
<td>4.7</td>
<td>3D Plot of Sharpe Ratio without Fees.</td>
<td>125</td>
</tr>
<tr>
<td>4.8</td>
<td>2D Plot of Sharpe Ratios With and Without Fees.</td>
<td>126</td>
</tr>
<tr>
<td>4.9</td>
<td>Percentage of Transactions.</td>
<td>126</td>
</tr>
<tr>
<td>4.10</td>
<td>Percent Smoothing is Better.</td>
<td>126</td>
</tr>
<tr>
<td>4.11</td>
<td>Percent of Positive Sharpe Ratio Slope.</td>
<td>127</td>
</tr>
<tr>
<td>4.12</td>
<td>PPS vs Price Change Volatility.</td>
<td>128</td>
</tr>
<tr>
<td>4.13</td>
<td>Effect of Lag on Profits.</td>
<td>129</td>
</tr>
<tr>
<td>4.14</td>
<td>Overall Forecasting Performance.</td>
<td>130</td>
</tr>
<tr>
<td>4.15</td>
<td>Percentage of Forecast Surface Above Original Surface.</td>
<td>130</td>
</tr>
<tr>
<td>4.16</td>
<td>ALGS Filter Performance.</td>
<td>131</td>
</tr>
<tr>
<td>4.17</td>
<td>ALGS Percent Smoothing is Better.</td>
<td>132</td>
</tr>
<tr>
<td>4.18</td>
<td>ALMA Filter Performance.</td>
<td>132</td>
</tr>
<tr>
<td>4.19</td>
<td>ALMA Percent Smoothing is Better.</td>
<td>132</td>
</tr>
<tr>
<td>4.20</td>
<td>Basic Filter Results.</td>
<td>133</td>
</tr>
<tr>
<td>4.21</td>
<td>Basic Filter Fallacy.</td>
<td>134</td>
</tr>
</tbody>
</table>
Certificate of Authorship

I hereby declare that this submission is my own work and to the best of my knowledge and belief, understand that it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at Charles Sturt University or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by colleagues with whom I have worked at Charles Sturt University or elsewhere during my candidature is fully acknowledged.

I agree that this thesis be accessible for the purpose of study and research in accordance with normal conditions established by the Executive Director, Library Services, Charles Sturt University or nominee, for the care, loan and reproduction of thesis, subject to confidentiality provisions as approved by the University.

Adrian Letchford

Signature

Date
Acknowledgements

I specially thank my principle supervisor, Professor Junbin Gao, who patiently allowed me to learn in my own way.

I thank my co-supervisor, Dr. Lihong Zheng, with whom I spent the most time talking and who sat and listened to all of my crazy ideas.

I thank the Spatial Data Analysis Network (SPAN) here at Charles Sturt University. They provided me with huge computing power and expensive software that allowed me to complete my work.

I also want to thank Dr. Philip Charlton, Professor Ken Russell, Professor Terry Bossomaier, Dr Xiaodi Huang, and Dr Zahidul Islam for reading my thesis and returning invaluable feedback.
Abstract

If you ever talk finance, in a blink of an eye your audience’s face feels as if they have been hit by a truck and slammed into a solid concrete bunker. Exaggerations aside, one can hardly blame them. The desire to line pockets with little green pieces of paper has lead to complicated monetary philosophies. In one camp there are technical analysts, in another value investors, then financial engineer or “quants”, and finally astrologists are thrown into the mix. Everyone has a different idea on how to conduct this business.

In this thesis, we are going to scientifically look at thoroughly used tools from technical analysis – the moving averages. These are filters which smooth noisy fluctuations from asset prices highlighting trends; they show the bigger picture. We will pull these models apart and explore them. Our aim is to discover what the relationship between smoothing and return actually looks like, and to answer the question: does smoothing really make us more competitive? We need to know this before we start investing real dollars.

We will answer this question with a three step process. In the first step I will show you how to measure smoothness and other properties of filters. In the second step, I will show you new filters which optimally smooth and allow you to control the amount of smoothing. Lastly, using our new ability to measure and control smoothing, we will look at the affect on returns. All contributions are listed in Chapter 5 on page 139.

Ultimately, we will discover that filtering offers no “edge” to investing. However, as smoothing increases, transactions and the money spent on them decrease.
List of Publications

I have published the following papers from this work:


The following paper has been submitted and has been accepted for publication:

Mathematical & Literary Standards

Mathematical Standards

- A lower case italic letter or symbol indicates a scalar.
- A bold lower case letter or symbol indicates a column vector.
- A bold upper case letter or symbol indicates a matrix.
- Unless otherwise stated, a subscript indicates that the vector or scalar is the $i^{th}$ value in a series.
- Indicates the Euclidean norm. It can alternatively be written as $||x||^2 = x^T x$.
- Denotes the identity matrix.
- Denotes a matrix of all ones.
- The dot (·) denotes multiplication.

Literary Standards

The word “T” refers to the author and the word “we” refers to both you (the reader) and me (the author).
HEATHER Joy Lawrence, a young nursing student, boarded with a dear elderly couple in Launceston, Australia; Mr & Mrs Letchford. Every day, Heather rushed through her work and, with the determination of an Olympian, flew home on a bicycle. Now, women may race for fatherly approval, long for a sweetheart’s attention or fight for feminine liberty; but not Heather Joy. The BBC tales of a tall disorganised man, possessing an unrivalled intellect and shaking a peculiar screwdriver, were enough to steal Heather’s drive.

Bursting through the Letchford’s front door, the sound of the TV scared Heather with the thought she may miss that night’s episode of Dr. Who. Charging into the living room she was surprised to see a young bearded man settling down to watch her favourite show. Without a second thought and nothing more than a distracted “Hi,” Heather jumped into a comfy chair. Roderick, the young bearded man, is the Letchford’s oldest son.

The next few years saw many outings, a spark of romance, a wedding and in 1989, I was born. I must have been a marvellous baby or a sheer embarrassment. My parents manufactured eight more of me – I am still unsure if they actually wanted copies of me, or were self-inflicting disappointment at each failed attempt of a perfect child.

After Justin and Benjamin, my next two siblings, were born, Dad and Mum executed a monumental idea that changed my family forever. With criticism shooting from every direction, my parents set out to school their progeny themselves. At five years old, I was withdrawn from kindergarten. If ever there is a list of Australian non-conformists, we ought to
be on it.

Dad holds a degree in science and a Ph.D in history while Mum is a medical practitioner, a registered nurse. Together, they made fantastic teachers. Their secret: they taught us to teach ourselves. Once we could read and write, they focused on independence. Each and every one of us taught ourselves through most of our education.

By 2005 Teresa, Monica, Catherine and Gerard had been welcomed into this world.

Home-schooling may be considered non-conformist but the best was yet to come. In 2005, I was 15 years old and the reality of not attending a “proper” school was heavy on our minds. However, my parents, resourceful and brilliant, found a solution. A loop-hole exists that is so far removed from institutional thinking my family could not resist its lure. To put it bluntly, the loop-hole makes a School Certificate useless and allows anyone willing to attend university with little effort.

I wanted to be a paramedic. So, just after my 16th birthday, I began studying medicine. The same year, Felicity was born. My performance was less than satisfactory. You see, there was a problem; I found medicine to be intellectually boring. That is not to say primates can make it through or that it is not useful. Indeed, it is the noblest and most demanding of professions. Only, spending every waking hour studying massive quantities of information that I could not use for another seven years destroyed my spirit. I wanted to do something.

My parents never forced a career on any of us, we were always told, “Do whatever you want to do!” Offering me their hand, I climbed out of medicine and dived into a different river.

In 2007 I started working towards a Bachelor of Computer Science and my youngest sibling, Dominic, was born. Suddenly, the world opened up. Every day was filled with a new discovery, a new way of thinking, or a new ability. I could explore database design and that evening build a working application. Perhaps a lesson on a cleverly designed algorithm would allow me to create a life-like game character. Even mathematics — I promised myself in school never to study this subject — taught me new abilities and paradigms that made the world ever more magnificent. Rising every day, I said to myself, “This is what I want to do for the rest
of my life!"

Because I was a distance education student, University was just like home schooling, only better! I studied the subjects I loved the most and when I wanted to do them. Soon, third year undergraduate crept up. Developing an obsession for finance and investing, I worked out an honours project to analyse currency prices with artificial neural networks (artificial brains).

More excited than opening gifts on my 10\textsuperscript{th} birthday, I searched for a supervisor. Luck! Someone wanted me. “Don’t do neural nets,” they said; a heated discussion followed suit. I was determined, nothing would change my mind. In frustration, the distinguished academic slammed down the phone wishing never to speak to me again. Leaving the situation subside for a few days, I phoned back. Another failed attempt to change my mind left him wanting to shake me off. He passed me to a man who became the third most important person in my education – my parents tie for first place.

I paced up and down the room convincing myself not to change my position. Ready for another fight, I rang Prof. Junbin Gao at Charles Sturt University. “Hello, I’m looking for an honours supervisor. I \textit{will} be researching neural networks.” “That’s great!” Came the unexpected response, “when can you start?”

Besides from his own inherent brilliance, Junbin Gao is a superb supervisor. He loves independence; a quality perfect for a self-taught young man. Junbin gave me the space I needed to learn and stretch my wings in the world of research. Half way through 2010, Junbin called me into his office. “I need to convince you to do a Ph.D.” he says. “Don’t bother,” I smiled back, “I already want to do it!” Following the usual procedures, he offered me one of several projects he had ready. “No,” I said to Junbin’s delight, “I have my own project.” He accepted the proposal for this thesis. I started in 2011.

Halfway through candidature I, and fellow Ph.D. students, asked Prof. Gao why he stands back and allows us to find our own path. He told us a story. As a young man, Junbin studied pure mathematics, yet, he thought big and was drawn to the exploding field of artificial intelligence. I must make one thing clear. Changing from mathematics
to machine learning is only slightly less dramatic than an astronomer becoming an astrologer; a move from solid equations to "if it works, it works." His Ph.D. supervisor did not mind the drastic turn and gave him the room needed to explore and discover. Junbin’s recollection shocked and inspired us, not to just finish our work, but, to aim further than we could imagine. Thank you, Prof. Junbin Gao.

Every week for the past two and a half years, I have visited Dr. Lihong Zheng, my co-supervisor. She is a thoughtful, fresh set of eyes with a different background to my own. I will remember Lihong for her patience and her joyful personality. Thank you, Dr. Lihong Zheng.

My fellow students were full of entertainment. Thank you to Jason Hambly, Anisur Rahman, Geaur Rahman, Lis Sar, Væntan Thinuvarudchelvan, Jason Traish, James Bekkema, Bin Liang, Saman Safigh, Nesa Mouzehkesh and Sabih-Ur Rehman, for the fun times together.

This thesis is a testament to my parents and their decision to home school their nine children. With only each other, they embarked on a perilous journey not knowing if they would succeed. They enlightened us to understand rather than know. They nourished us to think rather than follow. They formed us to act rather than wish. Dad and Mum, thank you.

Adrian Letchford
Wagga Wagga, Australia
May 10, 2013
http://www.dradrian.com
Introduction

We can't be afraid of change. You may feel very secure in the pond that you are in, but if you never venture out of it, you will never know that there is such a thing as an ocean, a sea. Holding onto something that is good for you now, may be the very reason why you don't have something better.

- Spoken by Poet, Philosopher, & Novelist C. JoyBell C.

Standing on the streets of New York city you see chaos everywhere, especially on Wall Street. The hustle and bustle of Americans finding their way around creates tremendous noise and confusion. Stock markets are no different. Investors negotiate and fight the traffic with more drama and energy than a Hollywood police chase. Conflicting opinions congest highways causing prices to wander randomly like a lost pedestrian. Mathematicians long ago invented noise filters, offering a road map to calmly point the road to riches.

Asset prices fluctuate, racing up and down unpredictably. Traders assume there are predictable trends hidden by chaotic behaviour. This noise is cleaned from asset prices using filters. They scrub away the chaos leaving a smooth curve that reveals when prices change direction. Markets are analysed in this manner in banking [132], economics [233], accounting [106], and computer science & mathematics [150].

The financial arenas are becoming more and more competitive every day. The largest syndicate in the world [248], and the most important [153], is the foreign exchange (currency) market. Four trillion dollars ($4,000,000,000,000) exchange hands every single day [184]. The sheer
size is caused by multi-billion dollar global companies who participate in substantial transactions for business [248]; as well as computerised dealing [184]. In 2008 75% of currency trading was conducted by interbank transactions [122].

The stock market easily forms the next largest monetary giant with a global capital of $36.6 trillion US dollars in October 2008 [27]. Shares are the most popular form of investment second to high interest savings accounts. Approximately 35% of Australian adults have holdings in stocks, this is not counting indirect holdings through managed funds [20]. Shareholders assume ownership of part of the wealth of the underlying company. Unlike currencies, where investors can earn or pay interest, stocks receive income through dividends, only suffering capital losses when prices decline [115].

![Figure 1: Example of filtering stock prices. As smoothing is increased, the lag also increases.](image)

Investors and traders use two types of filters to reduce price fluctuations in share and currency markets, offline and online. Offline models, also known as graduation, require future information to accurately remove noise [79, 132, 243, 255, 269, 274], while online filters only use present information to smooth prices [111, 173, 247, 250, 256]. Among the financial literature, online models are commonly called moving averages [45, 129, 233]. These moving averages do have one huge drawback: they lag behind the price. Figure 1 shows an example of the AUD/USD currency pair with two filters. When a stock peaks and then plummets, the filter’s smooth curve does not show this for some time. The delay between a change in price and the corresponding change in a filter is called the lag. More smoothing means more accuracy when gauging the price
direction. However, as smoothing increases, so does lag.

So far, researchers have focused on using moving averages for trading based on a simple set of rules \([45, 53, 91, 94, 129, 150, 155, 221, 233]\). Their conclusions give a “yes” or “no” verdict to use them for business. Astonishingly, to the best of my knowledge, nobody has published an exploration on the effect of smoothing on profits. We do not actually know how filtering affects profits. This is the motivation behind this thesis. What does the relationship between smoothing and return look like? Is smoothing useful? Does smoothing really give us an “edge,” that is, exploitable information?

In this thesis we will work out the relationship between smoothing and profit to tackle one simple question: is stock market noise reduction profitable? However, as with all worthwhile endeavours, this simple task carries some difficulties:

1. There is no method to measure the smoothness of an online filter in a single value when the original clean time series is not known and there is lag present.
2. There are no filters that have been designed to smooth asset prices.
3. Some methods of measuring smoothness exist \([28, 36, 44, 131, 138, 140, 167, 183, 204, 227, 228, 274]\). However, they do not work well on stock prices for reasons including, requiring large amounts of data, needing to know the data-generation process, and not being able to compare models or datasets.

The objectives of this thesis are:

1. To develop a way to measure the smoothness and lag of an arbitrary linear filter (Chapter 2).
2. To develop a new linear filter that maximises the amount of smoothing (Chapter 2). This is the main technical work of this thesis.
3. To develop a means of extrapolating a filter forward to compensate for lag (Chapter 3).
4. To conduct experiments on real world data to discover the relationship between smoothing and profit (Chapter 4). This is the
ultimate purpose of this thesis.

Only linear online filters will be refined for analysing financial markets. While the literature review (Chapter 1) will show you how linear filters relate to non-linear and offline filters, they are not considered in the scope of this research. Also, we will not develop a profitable strategy for investment. The one and only focus is on determining if smoothing is better than not smoothing. Even if a loss is incurred, if it is smaller than the losses of our non-smoothing competitors, filtering is successful.
Chapter 1

Background

Whenever you find that you are on the side of the majority, it’s time to reform – (or pause and reflect).
- Mark Twain (1904) in [220]

Look past the age of bartering and peer into the first world of currency. Five thousand years ago “I owe you” slowly developed and refined into “I owe you one unit of something.” According to anthropologist David Graeber [123], this is when credit and debt became formal. A “something” became crafted pieces of precious metal at the dawn of the coin phenomenon forged in ancient Greece. The earliest coin artefacts have been dated at 560 BC [149]. Over one and a half thousand years later in the 13th century, European culture was introduced to paper money from Asia by such renowned explorers as Marco Polo [117]. The first banknotes were issued in 1661 by a bank in Stockholm, Sweden [113]. Now, fast-forward to present day Wall Street; today’s monetary realm is a whole new planet.

Financial markets are vicious territories, fortunes amass and crumble, economies flourish while world super-powers are torn down. The 20th century saw talented investors such as Warren Buffett build incredible wealth. Yet, many financial players succumbed to suicide around the events of the Great Depression [186]. While some players find success,
others find failure.

General society is ignorant of the driving forces behind the entire financial arena. You only need to look at the incompatible differences between the investing “philosophies.” Benjamin Graham famously authored *The Intelligent Investor* [124], a book on value investing. This involves purchasing healthy companies at cheap prices. Graham’s advocates include 54 billion dollar Warren Buffett, chairman and CEO of Birkshire Hathaway [107], Walter Schloss [14] and Irving Kahn, chairman of Kahn Brothers [288]. An alternative school of thought examines historical prices for predictive patterns. Knights of price analysis include Hull [159] and Pesavento [234]. This field, also known as technical analysis, is even popular with fund managers [218]. A still relatively new breed of analysts come in the form of mathematical geniuses, financial engineers – colloquially called “quants.” In a nutshell, their premise is that markets are random. The goal is to watch risk like a hawk and take on investments that are mathematically or statistically in your favour. Paul Wilmott [276] and Emanuel Derman [83], head of risk at Primsa Capital Partners, are two proponents and personalities of financial engineering. Some people seek another avenue in astrology. Perry Kaufman, author of *Trading Systems and Methods* [176] now in its fifth edition, advocates astrology in his book, receiving critical acclaim from respected institutions including the Federal Reserve Bank of Chicago.

These ideas are not compatible because each one believes a different force drives the market. Value investors assume business drives the market; price analysts assume human behaviour, reflected in asset prices, is the motivating factor; financial engineers think everything is random and all is fair game; while astrologers follow the stars.

The great statistician George Box wrote that “Essentially, all models are wrong, but some are useful” [39]. Which begs the question, which are useful; value-investing, price analysis, financial engineering or astrology? The hugely conflicting opinions on investing lead to one conclusion. Financial analysis is still in the alchemy stage – or, perhaps less confronting, a pseudo-science.

In the spirit of Mark Twain, this chapter will pause and reflect on the literature surrounding technical (price) analysis. The review will demonstrate:
1. Trading model results are not conclusive or complete, and are subject to reasonable contradictions.

2. There are no filters for asset prices that optimally reduce noise.

3. A standard method of measuring and comparing the level of noise reduction on asset prices does not exist. As a result, the effects of noise reduction on profits are not known.

These points will be used to support further chapters which refine a popular trading strategy and succinctly expose its advantages and disadvantages.

The audit begins by investigating the theory and evidence of profitable investing in Section 1.1. Then, time series and technical analysis are explored in Section 1.2. Finally, noise reduction models are discussed in Section 1.3.

1.1 Can I Make Money?

Between academics, investors and any other professionals, opinions about profiting in the marketplace are divided. The major work surrounding this topic falls under the efficient market hypothesis (EMH). Eugene Fama [97] gave the first formal definition of an “efficient” market as reflecting all available information, at any point in time, in the price. This includes inferable information [169]. Prices adjust immediately in an efficient market, leaving no time to benefit. Inefficient markets change prices slowly, offering lucrative opportunities [169].

The efficient market hypothesis directly affects the research in this thesis. I want to carefully examine the mechanics and potential profit offered by moving average filters. If markets are efficient then such techniques are useless.

The concept of an efficient market was demonstrated in 1889 [114] where share prices were described as the best intelligent price. Following Fama’s 1965 definition of an efficient market, the first use of the term “efficient market hypothesis” was by Roberts in 1967 [237]. Fama then wrote the definitive paper on the EMH in 1970 [99]. Here, Fama conducted a
literary and empirical review and concluded that the EMH holds. The exception of slight positive correlation between prices changes of successive days (confirmed by [7, 97]) was dismissed because transaction costs absorb all profits.

The idea of excess returns is simply earning more than what is expected after risk has been considered. A consequence of the EMH is that investors are not rewarded for their ability to predict the future, rather, they are rewarded for the risk they take when purchasing risky securities [26,169,178]. The risk free return may be estimated from a Treasury Bond and then the minimal amount of return required to compensate for the risk of investing elsewhere is added to produce an expected rate of return. Returns above this expected rate are known as excess or abnormal returns [169] and are not allowed under the EMH.

Market efficiency does not require prices to change perfectly when receiving new information. An unbiased change is required. The final value occurring after market participants have fully assessed all relevant information is then an unbiased estimate. Such an efficient market may have prices greater than or less than the real intrinsic value of the security. However, investors will not be able to exploit this discrepancy for excess returns after all fees are accounted for [169].

There is significant controversy surrounding the EMH, with many market participants rejecting it completely. We know that stock prices are determined by investors' expectations of cash flows and risk [169]. Information is key to this valuation and is thus the central issue of the EMH [99]. Fama [99] organized the efficient market hypothesis (EMH) into three degrees of increasing information. These three forms are; weak, semi-strong and strong form efficient. The weak form states that markets incorporate all price and volume information. The semi-strong form adds to this all publicly available information. The strong form adds all private information thereby assuming markets reflect all available information. We say that a market is “weak-form efficient” if the weak form of market efficiency cannot be proven false in that market. The same can be said for the other two forms. Because the semi-strong and strong form build on the weak-form EMH, proving that the weak-form is false also proves the entire EMH to be false. The next three sections cover the details of each form and various attempts to prove or falsify each one.
1.1.1 Weak Form

A weak-form efficient market reflects all past prices and volume of any security. Extra information such as balance sheets, company announcements and Steve Jobs style keynote presentations are not incorporated in the price. Proving that knowledge of historical prices delivers excess returns also proves that the market is not weak-form efficient.

Using moving averages to remove noise from asset prices fits into the weak-form of market efficiency as only past price data is being used. Even if the semi-strong and strong forms can be proven true, the research in this thesis has a place if the weak-form cannot be verified.

The returns on securities ought to change randomly across time according to Samuelson [244], who used a martingale model. Indeed, many other researchers have supported this view, such as Fama [98], who examined the daily price changes on thirty stocks and found that prices follow a random walk with only a small percentage of movement explainable by previous changes. The idea that prices change randomly is called the Random Walk Hypothesis. This contradicted previous research, such as Alexander [8], who found that prices move in trends, and not random walks. These trends were identified by ignoring uniform periods of time, instead, using price movements as the unit of study.

Trading rules based on price movements were examined on the Melbourne Stock Exchange during 1958-70 for profitability [21]. A buy (sell) signal was flagged when the price advanced (decreased) by a certain amount from the last low (high) price. This trading system is known as a filter rule. The filter size relates to the amount of change needed to signal a position. Ball [21] concluded that these rules were not profitable on the Melbourne Stock Exchange during the test period. Other researchers have been able to cast doubt on the EMH by analyzing technical trading rules. Houthakker [148] discovered patterns in prices after analyzing stop-loss sell orders on commodities.

One interesting discovery in technical trading is seasonality. Monthly patterns in share prices were first shown by Rozeff & Kinney [239] in the US share market. The research showed that the shares returned the highest amount in January. Similarly, in Australia, shares are peaking in
January and July while falling in June [48]. New Zealand also exhibits this behaviour with January and April being the most profitable months [41]. Keim [177] related this seasonal effect in the US to the size of the companies. He showed that the small-size companies carried excess returns with half of these profits in January. More than half the abnormal January profits occurred in the first week of the month. Of every year under study, the first trading day carried a large return for the small firms.

DeBondt & Thaler [33] examined a hypothesis which implies that investors overreact to unexpected and severe events. Applying this hypothesis to the stock market, investors can expect that portfolios declining in value will soon outperform the market. By their experiments, they found that these loser portfolios earned excess returns of approximately 20% while the portfolios that were increasing in value soon returned 5% below the market. They interpreted this as overreaction. As the information they used to distinguish the appreciating assets (winners) from declining assets (losers) is only price data, it indicates that markets are predictable, discounting weak-form efficiency. This work has been supported in other research such as Chopra, Lakonishok, & Ritter [67] who adjusted stocks for risk and firm size and found that extreme prior losers returned more than the winners. The smaller firms returned more than the larger companies. This situation provides the best example of contradiction. Jegadeesh & Titman [168] showed that buying stocks that have been appreciating and selling under-performing stocks results in abnormal returns. This completely contradicts the work of DeBondt & Thaler [33].

Some authors indicate acceptance of week form efficiency while others reject it even when conducting the same study. As results are contradictory and unexplained, we must conclude that weak-form efficiency has neither been proven nor dis-proven.

1.1.2 Semi-strong Form

A market incorporating all publicly available information is semi-strong form efficient. Tests of the semi-strong form (SSF) of market efficiency involve testing the speed with which new information influences the mar-
Chapter 1. Background

This new information comes in the form of events such as share splits and earnings reports.

Eugene Fama, along with Lawrence Fisher, Michael Jensen, and Richard Roll wrote a largely cited paper [101] studying the results of share splits on returns. Their work is considered to be the original event study [169]. They support SSF market efficiency by arguing that a share split adds nothing to the market value of a company. Thus, a share split should have no effect on the price. Their results showed that any effects of a share split are reflected in the price immediately after the announcement. They did, however, make an important observation. Stock splits, in the investor’s mind, precede increases in dividends. This changes the expected cash flow and the value of the security. They noted that prices do make this adjustment before the announcement of a share split. While this may not support SSF, it implies one could profit from insider information on an upcoming split [101]; thus rejecting strong-form efficiency.

In 2001, Frino, Cusack & Wilson examined earnings announcements on the Australian Stock Exchange for abnormal returns [109]. They used seventy-eight of the largest firms and seventy-eight of the smallest. The study demonstrated that the value of the large companies adjusted within 30 minutes of the report. For the smaller companies, however, it would take up to six hours for the full adjustment. The researches concluded that the smaller companies are less SSF efficient than the larger. They suspected this is a result of the illiquid market\(^1\) for these smaller firms.

A convincing argument by Jones, Shamsuddin, & Naumann [169] says that US stock prices go through a substantial move prior to the earnings announcement but continue to follow a larger move in the same direction after the announcement for approximately 60 days. The moves are proportional to the unexpected earnings. This suggests that unexpected earnings are predictable from the prices alone, and the confirmation of the earnings results in a massive long term movement. These findings appear to be robust to different time periods and methodologies [22, 58, 100] and corroborate the findings of Frino, Cusack & Wilson [109].

A different approach to firm earnings is to invest based on the ratio between the market price of the company and the earnings, the P/E

\(^1\)An illiquid market or asset has very few people trading. Alternatively, a liquid market or asset has a lot of people trading.
ratio. This ratio has been interpreted differently through the decades. For example, Jaffee, Keim & Westerfield saw it as an indicator of mispriced companies [165] while Cragg & Malkiel [73] viewed it as reflecting earnings growth. Essentially, this ratio informs market participants of how much investors are willing to pay per dollar of earnings. The first major research was published in 1960 by Nicholson [226] who indicated that higher P/E stocks are outperformed by the lowest P/E stocks. McWilliams [216] investigated 390 companies and concluded that the worst performers began with a high P/E and the best performers could be found with any P/E ratio. However, the most profitable portfolio had only low P/E stocks. In 1975, Basu [23] pointed out that these studies did not account for fees, tax, and survivorship-bias\(^2\). After accounting for these, Basu drew the same conclusions, but indicated that those trading at a high frequency would be hindered by fees and tax. In 1977, Basu [24] divided U.S. stocks in two, high and low P/E groups. As the P/E ratio is publicly available, the SSF states it should be reflected in the price. Basu compared the two groups at one year after purchase. The final result was that low P/E stocks performed better than the high P/E stocks. In a later study [25], Basu investigated this P/E ratio effect along with the sizes of firms on the New York Stock Exchange. The results were the same as the previous study even after adjusting for the size of the companies. The P/E effect was also confirmed by Goodman in 1985 [118] who also showed that the level of risk does not affect this strategy, and again in 1986 by Peavy [119]. Finally, while Shen [254] also confirmed this phenomenon, the study found that the P/E ratio has no effect on real earnings growth.

The idea behind the P/E effect appears robust and the relationship has been confirmed across many studies. Though this is far from an exhaustive review, all the research presented here that discusses the P/E phenomenon easily rejects the semi-strong form of market efficiency.

1.1.3 Strong Form

A strong form efficient market reflects all information, public and non-public. The strong form (SF) of market efficiency is hard to test because

\(^2\)A dataset of stocks with survivorship-bias does not include companies that failed.
it requires private information. For example, confidential information on international negotiations known only to the CEO. The discussion on semi-strong form around the P/E ratio implies rejection of strong form as insider corporate investors may obtain this information before it is release to the public. There is only one notable study that does not rely on implications. Rozeff & Zaman [240] researched US stocks and found that anyone can make returns by trading with publicly available information on insider transactions. However, when they consider transaction costs, these profits disappear, suggesting that the corporate insider traders are also unable to make excess returns which then supports SF efficiency.

1.1.4 Concluding Remarks

The consistency with which market participants can produce abnormal returns is the key to testing the validity of the three forms [169].

Supporters of the EMH argue that, if it were false, then professional investors would be able to generate excess returns. Cowles [71] tested the returns of professional investors, concluding that they cannot forecast. Ten years later, Cowles again reported no ability for professionals to produce excess returns [72]. These finding have been repeatedly re-examined on many different time periods and the results are the same; stock portfolios compiled by professional investors do not produce returns above their benchmarks [212, 219]. However, this evidence is against investment professionals rather than in support of the EMH.

The researchers of [127] argued that perfect market efficiency is impossible. According to their reasoning, obtaining information involves a cost to the investor. If information were fully reflected in the prices, then investors would receive no compensation for purchasing the information. Therefore, a reasonable market model must allow for the incentive to gather information. Additionally, Jones, Shamsuddin, & Naumann [169] suggested an acceptable conclusion that some information may not be reflected in the market or that there is some lag in the distribution of information. Thus, markets may not be perfectly efficient.

Sewell [252] examined approximately 164 papers tackling the EMH and found that only just under half supported market efficiency. Accord-
ing to Sewell, the EMH cannot be proven nor dis-proven but remains the best hypothesis to describe markets.

One must be careful with the interpretation of the EMH. It does not say that profiting is impossible or improbable. One can consistently profit under the EMH even if only using price analysis (weak-form). The EMH simply states that it is not possible to earn excess returns.

Based on this review, we can the semi-strong and strong form of market efficiency solely on the price to earnings research. However, weak-form markets remain a possibility because the evidence for and against them is inconsistent. The best example of this inconsistency is DeBondt’s & Thaler’s [33] positive study on buying under-performing stocks against Jegadeesh & Titman [168] who sold the under-performing stocks with affirmative results.

This thesis investigates technical analysis. To my knowledge, there is no proof or even a falsifiable statement for the profitability of price analysis. The only way to prove it is to find a strategy that consistently generates returns. The discussion in this section has been light on various ideas to accomplish this. The following section discusses technical analysis in greater detail.

1.2 Number Crunching

Number crunching as an investing scheme, otherwise known as price analysis, is an act of forecasting no matter how simple or exotic. If you trade to close the transaction at a profit, you are forecasting. When successful, you’re a prophet with profit. Whether you gamble, or act with intelligence, you form an opinion on future events and act on that prediction. Other reasons for purchasing and selling assets include; capital preservation, interest or dividend earning, ownership, and need for cash. None of these involve forecasting. This work only considers the price-driven pursuit of capital gains, forecasting.

The problem of forecasting a time series is presented with a set of
historical input data with corresponding output data:

\[
\{ \{x_1, y_1\}, \{x_2, y_2\}, ..., \{x_T, y_t\} \},
\]

where \(x_t\) is the independent data at time \(t\) and \(y_t\) is the dependent variable to be predicted at time \(t\). To even consider forecasting, it must first be assumed that there exists a function that converts the data into the output: \(f: x_t \rightarrow y_t\). Only then can an approximate function be sought [287]. If \(y\) is a category, the task of finding \(f\) is called classification. If \(y\) is a numerical value, the task of finding \(f\) is called regression. There is an ever-increasing array of methods for this task, with many areas of applications including, agriculture [258], criminology [80, 120], ecology [171], medical management [263], hydrology [166], economics [69], and sales [64].

Once you have made a prediction about the future of a stock, currency or entire market, you can then trade. Before you actually start trading with your forecasting model and trading strategy, you need to know the performance of your system. You need to know the accuracy of your forecasting model and/or the estimated earning potential of your trading strategy using your forecasting model.

### 1.2.1 Performance Measures

There are many different performance measures for time series forecasting and stock market analysis. Some examples are: mean squared error, return on investment [150], Sharpe ratio [253], normalised mean squared error [136], modified Diebold-Marino statistic [136, 284] and maximum draw down [132]. There is no need to look at all of them, only those widely accepted and relevant to this thesis.

#### 1.2.1.1 Model Accuracy

The most common error metric for time series forecasting performance is the mean squared error [46] (MSE) which is defined as:

\[
\text{MSE} = \frac{1}{n} \sum_{t=0}^{n} (\hat{y}_t - y_t)^2,
\] (1.1)
where $\hat{y}_t$ is the prediction of $y_t$. This is often expressed as the root mean squared error (RMSE) [62]:

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n} \sum_{t=0}^{n} (\hat{y}_t - y_t)^2}.$$

The main advantage of the MSE is its mathematical elegance; it is continuously differentiable which allows us to find solutions to problems that minimise the MSE. For this reason, we shall use this measure later in this thesis. However, while the MSE measures time series forecasting performance, Gradojevic & Yang [122] suggest that the RMSE (thus also implicating the MSE) is not the best indicator of forecasting ability for a trading system. In a trading situation, absolute price accuracy is not necessary but accurate market direction forecasting is necessary. Gradojevic & Yang [122] do have a valid point here. In this thesis, we will use the MSE when building time series forecasting models. However, their trading performance will be measured by other means. Other users include [40, 200].

The hit ratio or rate (HR) measures the percentage of correctly predicted market directions. If the forecast is up and the market then moves up, that is a successful hit, and vice versa. While this may not be important in other areas, it is important in the financial sector [190]. Correctly predicting direction is easy money. The HR is defined in [105] as:

$$\text{HR} = \frac{1}{n} \sum_{t=0}^{n} h_t , \quad h_t = \begin{cases} 1 & \text{if } (y_t - y_{t-1})(\hat{y}_t - y_{t-1}) > 0 \\ 0 & \text{otherwise} \end{cases},$$

where $h_t$ is the “hit” at time $t$, $y_t$ is the time series we are forecasting and $\hat{y}_t$ is the forecast of $y_t$. A correct hit is 1 and a mistake is 0. The HR is written as a percentage. Other users include Hassani, Soofi, & Zhigljavsky [136].

While the Hit Ratio may overcome the problem of trading performance described by Gradojevic & Yang [122], minimising the MSE and the HR is not a mathematically simple task as the HR is not differentiable. Instead, we will measure investment performance independently with more suitable metrics discussed in the next section.
1.2.1.2 Investment Performance

When measuring the performance by monetary income, we want to know how profitable our forecasting model and our trading system are. That is, how much money would they have made us on our testing data? We could simply just calculate how much money we could have made, however, different investors and portfolios invest different amounts of money. Instead, return on investment is the primary (ROI) measure. This is the percentage gain that can be achieved from an investment strategy over a price series. The return or ROI is calculated as [150]:

\[
ROI = \frac{G - C}{C},
\]

where \(G\) is the gain from the asset and \(C\) is the cost. Literally, it is the amount you earned as a percentage of the amount you spent. For example, purchasing an asset for \$100 at a transaction cost of \$1 and selling for \$110 gives:

\[
G = 110, \quad C = 102 \quad \text{price of asset, one buy and one sell transaction,}
\]

\[
ROI = \frac{G - C}{C} = \frac{110 - 102}{102} = \frac{8}{102} = 7.84\%.
\]

The ROI handles differences between the amount invested, however, there are differences between assets and trading strategies. Notable, they fluctuate by different amounts. For example, two stocks may have the same expected ROI, but the price of one may fluctuate up and down by 20% while the other merely 5%. Sharpe’s ratio [253] handles this difference.

William Sharpe received the 1990 Nobel prize for economics along side Harry Markowitz and Merton Miller [266]. In 1966 Sharpe published a landmark paper, Mutual Fund Performance, where he developed what is now known as the Sharpe ratio [253]. The measure values excess returns as units of risk:

\[
\text{Sharpe} = \frac{E[r_a] - E[r_f]}{\sigma_a}, \quad (1.2)
\]

where \(E[r_a]\) is the expected return of the asset, \(E[r_f]\) is the expected
return of a risk free asset and $\sigma_a$ is the standard deviation of asset returns. This is generally read as the ratio between profit and risk.

Be aware that volatility ($\sigma_a$) does not quantify all the risk. It is merely a proxy for risk. There are other risks in the life of a company (or other asset) that are not reflected in the volatility. For example, there is a risk that a bad manager will take over and cause a significant drop in the price. This is a real risk that is not reflected in volatility until it has actually happened. The Sharpe ratio measures past risk and underestimates the real future risk. However, if the Sharpe ratio is used for tests that explore past data and is not used as the final decision variable for future investment activities, the limitations are irrelevant.

Andrew Lo, co-author of *A Non-Random Walk Down Wall Street* [208], heavily criticised the ratio in 2001 because it is based on sample estimation [206]. He illustrates that the Sharpe ratio can be incorrect by up to 65%. Other researchers claim that the Sharpe ratio is not consistent across different time scales [141]. A quick remedy is to annualise the measure. However, Lo also dismisses this approach [206] as it is only valid for independently and identically distributed returns. A few years later, in 2004, McLeod & Vuuren [214] praised the metric establishing that maximising the ratio does not maximise excess return per unit of risk but rather maximises the probability of outperforming the risk-free rate.

Saying that the Sharpe ratio is an estimate of profit per unit of risk is naive and likely to be incorrect. However, we know that the ratio is an excellent proxy for the likelihood of outperforming the risk free rate. That is, the greater the Sharpe ratio, the more likely an investment will earn excess returns.

We shall use the Sharpe ratio when examining trading performance later in this thesis.

### 1.2.2 Time Series Prediction

The financial forecasting literature is flooded with many different models, each with many variations. As Chapter 3 deals with forecasting a transformation of prices, this section very briefly reviews some of the most
frequent models used in the literature. This is by no means deep or exhaustive on any level, but only serves as an introduction to the various models and indicates sources for further information. The models covered are the autoregressive moving average, autoregressive conditional heteroscedastic model, artificial neural networks, recursive least squares filter and kernel machines.

1.2.2.1 Autoregressive Moving Average

The autoregressive (AR) model is an old model where the future values of a time series are determined from past values via regression. Recently, the AR model has been used for analyzing emerging Asian markets such as the Malaysian stock market [163] and the Taiwan market [60].

A similar model to the AR is the moving average (MA) model. Here, the time series is assumed to be a moving weighted average of a white noise series. These two models, AR & MA, have their origins in the early 1900’s [78]. Later, these two models were combined to form the autoregressive moving average (ARMA) model. Box & Jenkins [38] popularized the ARMA with their work on an iterative methodology to estimate the model parameters. The ARMA, or the generalized autoregressive integrated moving average (ARIMA), is used in such financial forecasting applications as; Amman stock exchange analysis [6] which supports the weak-form EMH; testing the efficiency of futures contract predictions [192] showing inconsistencies and resulting in an ambiguous conclusion of “approximate” efficiency; and hybrid stock forecasting models [229] where the ARMA based model was the worst performer. As these ARMA models have existed in the literature for quite some time, they are very often used as benchmarks for the development of more sophisticated time series models.

1.2.2.2 Autoregressive Conditional Heteroscedastic

A notable characteristic of financial time series is the apparent trends. Gains are followed by more gains, and losses are followed by more losses, or periods of no or little gains/losses are followed by continuing absence or small changes. In other words, there are times of high and
low volatility. This is termed volatility clustering in the econometric literature [78]. Engle [96] derived the autoregressive conditional heteroscedastic (ARCH) model which treats future returns as conditional on past returns. This model was further developed into the generalized ARCH (GARCH) model which include additional dependencies on the lagged conditional variance of returns [265]. The GARCH and ARMA have similar properties due to a similar representation. The (G)ARCH models are thoroughly used in econometrics and finance. For example see Sabbatini & Linton [242] who used option prices to estimate the implied volatility of the Swiss market index with superior performance to the Black-Scholes option pricing model [31], but, with a larger standard deviation of errors. More recently, Chortareas, Jiang, & Nankervis [68] examined high frequency data on different Euro exchange rates with variations on the GARCH model showing them to rival the ARIMA models.

1.2.2.3 Recursive least squares filter

The recursive least squares (RLS) filter is an adaptive linear filter which updates at each time step. The RLS filter updates in a recursive manner to be optimal over all the known input. If the input vector at time $t$ is $x_t$ then the matrix that represents all known inputs at time $t$ is:

$$X_t = [x_0, x_1, \ldots, x_t].$$

Then, the weights $w$ must be found such that $X_t^T w = y_t$, where $y_t$ holds all of the known outputs up to and including time $t$. Each row of $y_t$ corresponds to a column of $X_t$. The minimisation problem and solution is then:

$$w_t = \arg\min_{w_t} \|y_t - X_t^T w_t\|^2 = (X_t X_t^T)^{-1} X_t y_t.$$

This is the standard least squares problem. Recursive least squares updates $w$ without the expensive calculation of $(X_t X_t^T)^{-1}$ [203]:

$$P_t = (X_t X_t^T)^{-1}, \quad \text{then:}$$

$$P_t = P_{t-1} - \frac{P_{t-1} x_t x_t^T P_{t-1}}{1 + x_t^T P_{t-1} x_t}.$$
\[ w_t = w_{t-1} + \frac{P_{t-1}x_t}{1 + x_t^TP_{t-1}x_t} (y_t - x_t^Tw_{t-1}) \]

The above formula is used throughout the signal processing and machine learning discipline because it converts a problem involving a huge amount of data to a quick problem looking at one piece of data at a time.

A simpler variation is called the least mean squares (LMS) algorithm while a more advanced variation is known as the affine projection algorithm. They are both used in such applications as; equalization, signal modeling, beamforming, control, echo cancellation, noise cancellation [81,93,121,209,275], and speech processing [281]. The noise cancellation revolves around removing noise from audio signals and requires some means of estimating the noise (i.e. a second microphone closer to the noise source), as there are no known values for the desired system output \( d \) [1,121,275]. This model can easily be used for forecasting time series [42] in finance [92].

### 1.2.2.4 Artificial Neural Networks

Artificial neural networks (ANN) are essentially models of biological brains where artificial neurons are constructed and connected together in layers. Data is passed from one end of the network to the other, and the neurons are adjusted so that the network output is as desired. The major advantage of ANNs is their ability to approximate nonlinear functions [110]. Notably, it has been proven they can approximate any function [74,147]. ANNs are used for a huge range of time series modeling tasks. Examples of exchange rate forecasting include [161,231]. ANNs were applied to gold prices in [232] while stock prices were the issue in [3]. Neural networks are also combined in many different ways, such as the method of Chen, Peng, & Abraham [65] who stacked ANNs together. For a review and extensive list of literature, see [19].

### 1.2.2.5 Kernel Machines

Kernel methods are brilliant little models with powerful abilities. The idea behind them is beautifully simple; transform your data into a higher dimensional space and then use a linear model [203]. The only real
challenge is choosing a higher dimension that can represent your problem linearly.

Amazingly, data can be transformed into a space with an infinite number of dimensions. While this is actually impossible to do in practice, a mathematical trick, known as the kernel trick, is used to make this concept a reality. Kernel models transform the input vector $x$ into an infinite vector $\varphi(x)$. The equations for many linear models can be rearranged so that they only involve inner products, $\varphi(x_a)^T\varphi(x_b)$, which are scalars. By the Mercer theorem [203], a function $\kappa$ can be found such that $\kappa(x_a,x_b) = \varphi(x_a)^T\varphi(x_b)$. This solution allows for simple manipulation of data in an infinite space as $\varphi$ does not have to be explicitly calculated, only the kernel function $\kappa$.

The most common kernel method in the financial literature is the support vector machine (SVM). After the transformation of input vectors, the SVM finds a hyperplane to separate the data classes. The hyperplane is defined with the closest data points – the support vectors. One of the best introduction to SVMs was written by Burges [52] in 1998. For an example of their use see Kim [179] who used SVMs to predict the future stock price direction. Elsewhere, SVMs have been used for exchange rate prediction [152], multiple-kernel SVMs were used in [280], kernelized regression was combined with wavelet analysis in [151], and kernel methods were used for portfolio optimization in [164]. A good example of kernel method forecasting is a study on the Korean composite stock price index by [179].

As an alternative to time series forecasting, analysts calculate features of an asset price series and use these features to decide whether to buy or sell. This area is called technical analysis.

### 1.2.3 Technical Analysis

Technical analysis is defined as the study of security prices and their movements [55]. Technical analysis manipulates security prices into indicators displayed on a price chart. Indicators show if the market is trending, moving sideways, and if the price is high or low. Decisions to buy or sell assets are made using technical analysis.
CHAPTER 1. BACKGROUND

TA methods do not explicitly forecast asset price. Instead, they calculate properties about the present or, more often, the past. A decision to buy or sell is then made based on this information. Because of this, technical analysis has been described as understanding how humans react to market situations [205, 213, 249].

There is a massive array of price indicators; a short list includes: average true range [55, 156, 215], support and resistance [47, 55, 156, 215, 267], turning points (fractals) [55, 215, 267], Douchian channel [55], moving averages (simple, weighted, exponential) [29, 47, 55, 156, 158, 215], moving average convergence divergence [47, 55, 156, 215], candlesticks [29, 47, 156, 194, 195, 215], gaps [47, 267], double tops and bottoms [47, 156, 215, 267], relative strength index [47, 156, 215], volume [158, 215], on balance volume [215], directional movement index [215], stochastic oscillator [29, 47, 156, 215], linear regression [47, 156], Fibonacci [47, 156], Keltner Channel [47], Ichimoku Kinko Hyo [47], Bollinger Bands [47], and head & shoulders [47, 76, 156, 267].

Technical indicators are far too numerous to detail each one. Three methods will be discussed to paint a picture of the TA field, the relative strength index, the head and shoulders pattern and Bollinger bands. The relative strength index is an example of converting the non-stationary price series into a stationary oscillating signal; the head and shoulders pattern is just one of the many price formations traders look for; while the Bollinger bands are an example of removing price fluctuations and estimating an upper and lower price boundary.

Relative strength index (RSI), and other momentum indicators, measure price direction and speed [202]. Relative strength (RS) is the average of the past \( m \) price increases divided by the average of the past \( m \) price decreases [215]:

\[
\begin{align*}
    u_t &= \begin{cases} 
    p_t - p_{t-1}, & \text{if } p_t > p_{t-1} \\
    0, & \text{otherwise}
    \end{cases} \\
    d_t &= \begin{cases} 
    p_{t-1} - p_t, & \text{if } p_{t-1} > p_t \\
    0, & \text{otherwise}
    \end{cases}
\end{align*}
\]
The RSI is then:

\[ \text{RSI}_{t,m} = 100 - \frac{100}{1 + \text{RS}_{t,m}}. \]

When the RSI is above 70, this signals a market top and may invoke a decision to sell. When the RSI is below 30, this signals a market bottom and may invoke a decision to buy [202]. Another rule buys when the RSI is above 50 and sells below. Using this 50-50 rule, Chong & Ng [66] concluded that the strategy beats the buy-and-hold method on the FT30 index over a 60 year (daily) period. However, they did not include transaction costs. Wong, Manzur, & Chew [278] examined both the RSI methods on the Singapore Stock Exchange and found mixed results even with no transaction costs.

**Head and Shoulders pattern** is just one of many patterns analysts claim to have predictive power. The prices moves up and down forming a pattern that looks like a human head with a shoulder on either side. Figure 1.1 shows an example of this pattern.

![Figure 1.1: Head and Shoulders Example](image)

Figure 1.1: Head and Shoulders Example. The pattern is characterised by a high middle peak, two equally high “shoulder” peaks with an equal base or “neck line.” The idea is to sell when the price breaks below the neck line after the second shoulder. When upside down, the signal is to buy.

The idea is to buy or sell once the head and shoulder pattern has been identified. Originally, this process was performed visually and was thus entirely subjective. However, now researchers use computer models to find the patterns and check for any significant profitability. Savin, Weller
& Zvingelis [246] tested this strategy on the S&P 500 and Russell 2000 from 1990 to 1999 and found no profitability. However, they did discover that including the pattern in an indexing strategy does produce excess returns. Alexeev & Tapon [9] tested the head and shoulders pattern as well as nine other patterns on the Toronto Stock Exchange and failed to reject the weak-form hypothesis.

**Bollinger Bands** were invented by John Bollinger in the early 1980s [32] to provide an indicator of when prices might be high or low. The Bollinger bands are a price and volatility indicator composed of three lines. A middle line is the average price of the past $m$ days. A top and bottom line add or subtract $k$ standard deviations from the average, also calculated over $m$ days. The price is considered to be high (low) when it is close to or over (below) the top (bottom) band.

$$\text{BB}_{m,t} = \mu_{m,t} \pm k \times \sigma_{m,t}.$$  

Leung & Chong [198] tested the trading rules of buy (sell) above (below) the bands on global indices from 1985 to 2000. Without transaction costs they reported mixed results. Lento, Gradojevic, & Wright [197] tested the rule on a smaller group of global assets and found, after including transaction costs, that they could not provide excess returns above the buy-and-hold strategy.

A very popular class of technical indicators is the “moving average.” These are filters that remove noise from the prices and they are the focus of this thesis. This thesis drills down on to moving averages in Section 1.3.2.2 which divides the filters into exponential smoothers, linear filters and adaptive filters; and Section 1.3.3 which reviews how they are used for trading.

### 1.2.4 Final Remarks

Section 1.1 covered a huge amount of work on the efficient market hypothesis, concluding that price analysis has so far not been shown to be useful. A more specific field tests stock market prices to see if the price changes are dependent or simply change randomly. This is known
as the random walk hypothesis. If it can be shown that the prices do follow a random walk, then any effort spent to forecast prices is frivolous. Here too, the results are mixed. Abraham, Seyyed, & Alsakran [2] tested the Saudi Arabia, Kuwait and Bahrain markets, finding conflicting results. They showed that the Kuwait market was not a random walk, while the other markets did exhibit price change independence. Lo & MacKinlay [207] did considerable work to reject the random walk hypothesis using a variance-ratio test. However, they also stated that there is no other plausible model for the prices, and they did not reject market efficiency. The theory and experiments suggests that nothing has been found that is better than the random walk.

Fama [97] conducted a review of the random walk hypothesis in 1965, concluding that the evidence for it is strong. Fama stated that those who depend on price based methods of investing must show consistent and reliable results. Despite this statement by a well known researcher of this area, studies do not strive to show consistency of their models. For example, Huang, Nakamori & Wang [154] compared the hit ratio of the SVM and ANN against the random walk and two other models. They demonstrated each model to be significantly better than the random walk. However, their analysis covered only one asset with 640 training samples and 36 testing samples, far too few to be considered significant.

I want to suggest that perhaps the conflicting results are due to a lack of a full exploration of the problem and results. For example, why does one specific rule using the RSI indicator work and other settings do not? Is this just random behaviour or is there a reliable behaviour here? When Savin, Weller & Zvingelis [246] tried to trade the head and shoulder pattern, why did it not work, but incorporating the idea into a indexing strategy did work? I argue that trading results are not enough. An explanation as to why and the evidence for it is also needed.

One specific kind of time series analysis is noise reduction. Removing distortions from the prices is often performed before more advanced analysis or as a stand-alone trading system. The following section discusses noise reduction in further detail.
1.3 Noise Reduction

Researchers across many fields have realized that the effectiveness of predictive technologies does depend, partly, on preprocessing the data [162, 201, 230]. Appropriately preprocessing data decreases model training time and improves forecasting accuracy [153]. However, not all researchers in predictive analysis agree that preprocessing ought to be performed. Nelson et al. [225] believed that, due to the ability of neural networks to approximate any function, preprocessing is irrelevant. Care should be taken; arguments such as this rely on proofs that may not be practical in reality. For example, the proof that neural networks can approximate any function is based on the assumption that no constraint is placed on the model size. The size is finite, but it could be very large [74]. This may not be practical because the model may be slow on a computer or over fit the data.

Traditional statistical analysis performs time series preprocessing before tasks such as regression to combat effects of non-normality or heteroscedasticity [251]. Such data transforms include the log ($\log(y)$), square ($y^2$, or power), square-root ($\sqrt{y}$), and the reciprocal ($\frac{1}{y}$) transforms. These methods are used in the financial literature. For example, Andreou, Georgopoulos & Likothanassis [12] used a combination of the power and log transformation as first used by [37] for the purpose of forecasting exchange rates.

A continuing trend for time series data preprocessing is noise reduction. Financial time series are extensively polluted by noise which is considered to be the unpredictable part of the data. This noise causes great difficulty in forecasting and gives security prices an apparent lack of form. Research has shown that noise reduction increases the performance of forecasting, for example as applied to hydrology [174] with an extended Kalman filter or exchange rates and consumer price index [259] with local projective, singular value decomposition.

The following section (1.3.1) discusses measuring noise and noise reduction. Section 1.3.2 then looks at the current noise reduction models. Finally section 1.3.3 elaborates on how the moving averages are used for investing.
1.3.1 Quantifying Noise & Noise Reduction

The term “noise” is subjective [102], with varying meanings in different disciplines. Signal processing, for example, treats noise as the higher frequencies, and removes them with filters. Other areas treat noise as errors in the data. A formal definition of time series noise identifies two types: dynamical [102, 188] and measurement [102, 188, 199] noise. Dynamical noise is within the system while measurement noise is due to less than perfect measurement devices [95]. Neither is restricted to be erroneous, only to have an unknown model.

A system with dynamical noise can be expressed as:

$$x_t = f(x_{t-1}) + w_t,$$  \hspace{1cm} (1.3)

where $f$ is the system, $x_t$ is the system output at time $t$ and $w_t$ is an independent white noise variable.

A system with measurement noise can be modelled as:

$$y_t = x_t + z_t,$$  \hspace{1cm} (1.4)

where $y_t$ is the measurement of the real system output $x_t$ and $z_t$ is an independent white noise variable. $x$ may follow (1.3).

The presence of noise in any time series data limits the ability to extract useful information [187] for any purpose, not just forecasting. Noise reduction’s ultimate goal is to remove the noise component or at least minimise its effect [102]. Formally, the aim is to take the series $y$ as described in equation (1.4) and estimate the series $x$. As this is an estimation process, estimating $x$ may involve removing some of the noise as well as some of the deterministic part of the time series. For example, Whittaker [274] used a least squares method to reduce noise by any amount. Maximising the noise reduction simply fits a polynomial to the data.
1.3.1.1 Measuring Noise

There are a handful of different methods throughout the literature for measuring the effect or level of noise. The most meaningful ones are presented here.

The **mean square error** (MSE) or equally the root MSE (RMSE) is the most basic measure; see (1.1). It measures the error between the real noise free time series \( y \) and the noise reduced estimate \( \hat{y} \). While this utilizes a standard metric, the drawback is that the actual real noise free time series must be known. While this may work well in a controlled setting, this is not possible on real world data. An alternative direction is to use another method to estimate \( y \) but the quality cannot be ensured and the resulting error will depend on the method’s accuracy. Studies employing this method include [131, 140, 167, 204, 227, 228].

The **signal to noise ratio** (SNR) is a statistical measure comparing the variance of the signal to the variance of noise. This ratio is expressed in the decibel scale to combat the wide range of values. The function is defined as:

\[
\text{SNR} = 10 \cdot \log_{10} \left[ \frac{\text{var}(y)}{\text{var}(\hat{y} - y)} \right],
\]

where \( y \) represents the pure noise-free data and \( \hat{y} \) the filtered noisy data. The SNR shares the same problem as the RMSE – the original clean data needs to be known. Studies using this measure include [36, 44, 82, 131, 138, 183, 204, 227, 255].

To use the two discussed methods for stock price analysis is impossible owing to the necessity of knowing the clean data. In recognition of this, several alternative methods have been developed. An example is **autocorrelation analysis** which measures the level of correlation in a series at a certain delay. The rationale is that white noise has no autocorrelation, while an “effective” signal has a much higher result [131, 204]. The non-normalized measure is:

\[
R_{\hat{y}}(t) = \frac{1}{N-t} \sum_{n=1}^{N-t} \hat{y}_n \hat{y}_{n+t}.
\]
Noise reduction is considered to be successful if $R_y(t) - R_y(t) > 0$ for various selections of $t$, and increasing removal of noise as $R_y(t) \to 1$. There is a problem with this method. For proper analysis, multiple values of $t$ should be used. This results in high dimensional data when analysing many models on many data sets. Alternatively, Jayawardena & Gurung [167] measure the correlation between the predicted filtered series and observed series. However, then the best noise reduced series (a result of 1) ought to have as many directional changes as the original series.

Other methods include power spectrum analysis, correlation dimension analysis [28, 167], and the many forms of recursive analysis – see [131, 204] and references therein. All of them suffer from high dimensionality and complexity. The output of the power spectrum analysis, for example, is large enough that most researchers visually inspect the results. This becomes impractical as the number of models, data and adjustable parameters increase. For large scale experiments on real world data, alternative measures of noise reduction need to be developed.

The mathematician E. T. Whittaker presented a noise reduction measure which overcomes these limitations [274]. Whittaker assumed that the noise-free data is smooth and measures this property. Thus, he avoids the messy noise model and the need to examine the original clean data. A time series is smooth if it has a small $d^{th}$ derivative or rate-of-change. The change of $y_t$ is $\nabla y_t = y_t - y_{t-1}$ where $\nabla^2 y = \nabla(\nabla y)$. This is extended to vectors by defining the matrix $D_d$ such that $D_d\mathbf{y} = [\nabla^d y_1, \ldots, \nabla^d y_n]^T$. To reduce this vector to a single figure, Whittaker just squared and summed each element: $||D_d\mathbf{y}||^2$. The drawback is that the measure cannot be compared between datasets that are on different scales.

Because of the need to know the true noise free data, or the large output of a measure, or the inability to compare between models and datasets, I find no measure of noise that can be used to compare the performance of different noise reduction methods on different asset price series.
1.3.2 Models

Previously, the noise reduction literature has divided the existing techniques into three categories: time, frequency, and time-frequency/time-scale domains [285]. A time domain example may be an exponential moving average, the Fourier transform is the most popular frequency method, and time-frequency/time-scale models include wavelet analysis.

I suggest, instead, two categories, independent of time/frequency domain, that focus on the input/output data of the noise-reduction algorithm. These two groups are termed “offline” and “online”. The terms are not original and have been used before.

1.3.2.1 Offline

Another term for offline is graduation. Graduation is an old term from research such as Whittaker’s method of graduation [274]. The common characteristic of this group of models is the assumption that the time series is finished – all data is available for use. This means that every point in the time series is smoothed with information from every other point. Obviously, these methods do not hold well to financial purposes as the price series is not finished but is continually being updated. The rest of this section will discuss details of the following models: singular value decomposition, Whittaker’s, empirical mode decomposition, and wavelet analysis.

**Singular value decomposition** (SVD) decomposes a time series into singular values & vectors and constructs separate signal & noise subspaces. The noise subspace is discarded and the time series is reconstructed [255]. The time series $\mathbf{y}$ of length $N$ is first defined in a trajectory matrix:

$$
\mathbf{Y} = \begin{bmatrix}
    y_1 & y_2 & \cdots & y_M \\
    y_2 & y_3 & \cdots & y_{M+1} \\
    \vdots & \vdots & \ddots & \vdots \\
    y_{N-M} & y_{N-M+1} & \cdots & y_N
\end{bmatrix}
$$
The SVD of this matrix is given as: \( \mathbf{\check{Y}} = \mathbf{USV}^T \) where \( \mathbf{S} \) is a diagonal matrix of the singular values and \( \mathbf{U} \) and \( \mathbf{V} \) are matrices of the left and right singular vectors respectively. The subspace separation can be expressed as:

\[
\mathbf{\check{Y}} = \mathbf{USV}^T = \begin{pmatrix} \mathbf{U}_s & \mathbf{U}_n \end{pmatrix} \begin{bmatrix} \mathbf{S}_s & 0 \\ 0 & \mathbf{S}_n \end{bmatrix} \begin{pmatrix} \mathbf{V}_s^T \\ \mathbf{V}_n^T \end{pmatrix},
\]

where the subscripts \( s \) and \( n \) mean signal and noise respectively.

Projecting \( \mathbf{\check{Y}} \) into the signal and noise subspaces completes the noise reduction:

\[
\mathbf{\check{Y}}_s = \mathbf{U}_s \mathbf{U}_s^T \mathbf{\check{Y}} = \mathbf{\check{Y}} \mathbf{V}_s \mathbf{V}_s^T = \mathbf{U}_s \mathbf{S}_s \mathbf{V}_s^T,
\]

\[
\mathbf{\check{Y}}_n = \mathbf{U}_n \mathbf{U}_n^T \mathbf{\check{Y}} = \mathbf{\check{Y}} \mathbf{V}_n \mathbf{V}_n^T = \mathbf{U}_n \mathbf{S}_n \mathbf{V}_n^T,
\]

\( \mathbf{\check{Y}}_n \) is simply discarded.


**Whittaker’s method of graduation** cleverly combines approximation and his measure of smoothness into a least squares algorithm. Say the noisy data is in vector \( \mathbf{y} \) and the smooth data is represented by \( \mathbf{\hat{y}} \). The first concern is minimizing the sum of squared differences between them, which can be written as \( \| \mathbf{y} - \mathbf{\hat{y}} \|^2 \). The other concern is reducing the noise; the smaller \( \| \mathbf{D}_d \mathbf{\hat{y}} \|^2 \) is the smoother the approximation. The problem is written as [90]:

\[
\mathbf{\hat{y}} = \arg \min_{\mathbf{y}} \{ \| \mathbf{y} - \mathbf{\hat{y}} \|^2 + \lambda \| \mathbf{D}_d \mathbf{\hat{y}} \|^2 \} = (I + \lambda \mathbf{D}_d^T \mathbf{D}_d)^{-1} \mathbf{y},
\]
where $\lambda$ controls the level of smoothing and $d$ is frequently chosen to be 2 or 3. Effectively, this is a kernel algorithm where the kernel matrix is: $(I + \lambda D_d D_d)^{-1}$. For more detail see the discussion in [90], the original article [274], and some application to economics [142].

**Empirical mode decomposition** (EMD) transforms the data much like the Fourier transform, bypassing the typical restrictions such as periodicity and stationarity which are not always observed in financial data [269]. The EMD decomposes a time series into several progressively slower oscillating sinusoid-like series with varying amplitude and frequency. These new series are known as intrinsic mode functions (IMFs). If the time series is represented by the vector $\mathbf{y}$ then it can be described as:

$$\mathbf{y} = \sum_{i=1}^{N} \text{IMF}_i + \mathbf{r}. \quad (1.7)$$

The vector $\mathbf{r}$ is a non-oscillating series considered to be the main trend. The EMD algorithm finds the IMFs and $\mathbf{r}$. The IMFs oscillate in their own time scale and are significantly simpler than $\mathbf{y}$ and are expected to be more easily extrapolated [269]. A noise reduction algorithm would be to reconstruct $\mathbf{y}$ with equation (1.7) by ignoring the first $k$ IMFs, as the lower IMFs are the noisiest.

Tsakalozos, Drakakis & Rickard [269] used the EMD to analyze and forecast exchange rates and stock markets. After consultation concerning state of the art methods, the researchers concluded that their method performed to a high standard. Other EMD research includes Zhang, Lai, & Wang [286] who analysed oil prices and Drakakis [88] who examined stock market trading volume. For more detail see [35, 36, 88, 128, 286].

**Wavelet** methods for noise reduction are considered to be state-of-the-art. There are many very different variations. However, the background and basic method will be discussed here. Wavelet analysis is similar to Fourier analysis. The major differences are that it is performed in the time domain and, rather than using sines and cosines, it makes use of scaled and translated versions of a wavelet function $\psi(x)$ defined as:

$$\psi_{j,k}(x) = \delta \cdot \psi(2^j x - k),$$
where $\delta$ is a constant, $j$ is the scale and $k$ the translation. Similarly to Fourier analysis, a function $f(x)$ can be modeled as:

$$f(x) = \sum_{\forall j} \sum_{\forall k} c_{j,k} \psi_{j,k}(x),$$

where $c_{j,k}$ are wavelet coefficients which are derived from the wavelet transform:

$$c_{j,k} = \int_{-\infty}^{\infty} f(x) \psi_{j,k}(x) \, dx.$$

The Fourier analysis is only able to determine what global frequencies are within a signal, not where they occurred. The wavelet analysis is able to tell both by observing the wavelets and coefficients in the function approximation. Similarly to the EMD, wavelet analysis does not rely on the restriction of periodicity or stationarity. Wavelet analysis essentially splits the data into multiple series where one represents the trend, one represents the noise, and the rest represent the various oscillations in the data.

A basic algorithm of noise reduction by wavelets is discussed in [79]. Noting that the coefficients indicate the contribution of the wavelet at each particular scale and translation, a small coefficient indicates little contribution. Through some thresholding function, small coefficients are set to 0 before reconstructing the time series. Cannas et al. [54] took noisy river flow time series and separated the noise with discrete and continuous wavelet analysis. They reported an increase in performance (40% to 57%) when forecasting the monthly data. For further information and more advanced noise reduction strategies see [86, 87, 151, 204, 238].

Offline models are often used as a preprocessing tool for advanced regressions. For example, Yu, Wang & Lai [283] use EMD to filter and decompose oil prices before forecasting with a neural network. Their model outperforms models without EMD for both RMSE and hit rate. Some models are used for raw technical analysis without explicit forecasting. For example, Harris & Yilmaz [132] use the method of Whittaker [274] (they cite [142] whose method is the same), and report higher returns and Sharpe ratios than online filters. Either way, the offline models are used just like the online models, only with a larger amount of processing.
Offline methods are perhaps not suitable for financial applications for several reasons. The first is that as each new price value is available, the entire smooth curve requires recalculation. In addition, because reducing the noise on any given data point relies on the information in future data points, forecasting becomes even more difficult. Finally, researchers use these noise reduction models to boost the reported performance of their forecasting models by preprocessing the price data with a graduation model. The forecasting algorithm is then applied to the graduated data, effectively producing a model which has future data as input to forecast the exact same future data. Examples of such studies include [217, 269].

1.3.2.2 Online

Online models filter the data as it is being received. They can be written as:

\[ \tilde{y}_t = f(y_{t+l}, y_{t+l-1}, \ldots, y_{t+m}) \]

where \( l \) is the number of future points required to calculate \( \tilde{y}_t \), best known as the lag, and \( m \) sets how much data to use. There are three different names based on \( l \), They are, prediction, filtering, and smoothing. Prediction involves estimating the noise free value using old noisy data, \( l < 0 \). Filter models estimate the noise free signal using all available information present, \( l = 0 \). Smoothing models use some future information to clean the data, \( l > 0 \) [222]. Online models provide more accurate estimates as lag increases. That is, as more and more noise reduction is desired (increasing \( l \)), the output is delayed more and more.

This section will cover some of the models that have the potential to smooth financial data and not just the ones used for this purpose. First, I must provide some clarification. The three types of models prediction, filtering and smoothing have the potential to smooth time series data. They only differ by \( l \). From here on, they shall be referred to as filters while “smooth” will be used to refer to the smooth curve as a result of filtering. Smoothing and filtering will be used interchangeably. Not all the models in this section have been applied to financial smoothing; many of them cannot be readily applied. They are included here because they are still valid filters with valid and useful points.
State space models focus on the goal of exact estimation of the state of the system generating the signal. This problem comes from control theory [173] and is the only filter reviewed that follows the noise model presented in section 1.3.1. State space models assume a system, where the state is hidden, from which noisy measurements are made. The goal is to estimate the correct state via filtering. The system can be generalized [15] to:

\[
x_{t+1} = ax_t + w_t, \\
y_t = bx_t + v_t,
\]

where \(x_t\) is the state at time \(t\) and \(y_t\) is the noisy measure and \(w\) and \(v\) are random variables. These equations can be of either scalars or vectors.

Many of the state space models conduct two levels of filtering. One is a forward pass where each point is filtered by a priori knowledge and the knowledge gained from previous values in the time series. The next level is a backward pass, correcting each value, resulting in a smooth series [56, 116, 185]. If a backward pass is used then it becomes an offline model. However, some state space models only apply a forward pass. A filter of this kind generally follows two stages for each state: predicting the next state, and, once the actual measurement is available, updating this prediction to yield the estimated state [15, 17, 173].

Kalman published the groundbreaking discrete state space solution to the Weiner filter in 1960 [173] — the Kalman filter (KF). Moore [222] and Rubinstein [241] performed experiments with delays in Kalman filters, showing that these suboptimal smoothers were able to achieve good results comparable to the optimal smoother. Kownacki [189] used the Kalman filter for acceleration and angular rate signals. Altunkaynak [10] used a genetic algorithm and the Kalman filter to predict suspended sediment concentration.

The Kalman filter is the exact solution to the state estimation problem given the restrictions of a linear system polluted with Gaussian noise [17]. The extended Kalman filter (EKF) was derived to overcome the linear limitations of the original filter and cater for nonlinear systems [273]. Unfortunately, the EKF does have problems. It has been termed a first order approximation [271], injecting large errors into the
mean and covariance of transformed random variables, possibly leading to poor performance. The unscented transform was applied to the EKF to overcome these problems [170]. The resulting model is the unscented KF (UKF) producing a third order approximation [271].

Despite the success of these Kalman based filters, some systems may have non-Gaussian multi-modal probability distributions. In recent years, particle filtering has been used for state estimation to overcome the Gaussian limitations [17]. The particle filter uses many 'particles' to represent a probability density function (PDF) for the system and, by using Bayesian theory, construct each state's PDF recursively from previous estimates and current measurements [17]. Fatemi [103] used particle filtering to reduce the noise in a quasi-periodic signal by estimating the Gabor coefficients and showed that in some cases it is better than (or at least close to) wavelet graduation methods. In [85], multiple state space noise reduction algorithms of the filter, prediction, and smoothing form are calculated for the particle filter as well as the linear Gaussian case of the Kalman filter.

State space linear (finite impulse response, FIR) filters have been developed in recent years as an alternative to the Kalman filter [182]. FIR filters, notably those developed in [180,182,191], provide advantages over the Kalman filter with an infinite impulse response (IIR) behavior; a priori information concerning the initial state is unnecessary. Due to the FIR structure, the filter is more robust to temporary uncertainties than the Kalman filter [182]. The FIR state space models are also bounded input/bounded output (BIBO) stable [257]. Research to improve these models is still current. Ahn & Kim [5] derived a maximum likelihood state space FIR filter. A method of incorporating knowledge of the initial condition was presented in [181].

State space filters are fantastic models in every way except one. They require knowledge of the underlying system and measurement bias (refer to equation (1.8)). A filter that does not require knowledge of the data generating system may be more useful for asset prices. This leads into exponential smoothing.

The following discussions on exponential smoothing, linear filters and adaptive filters handle the details of moving average filters. The term "moving average" is a broad term used to describe asset price filters.
Exponential smoothing (ES) filters owe their origins to tracking models on U.S. navy submarines and were later developed for naval inventory control. The creator, Brown, first published [49] after developing trend and seasonality into the models. Later, the influential article [277] tested work in [145, 146] which independently developed similar models for additive trends and seasonality. There are 15 standard models, each an extension of the original works by Brown [49, 50], Holt [144] and Winters [277]. Gardner, in 2006 [112], compiled an excellent history, review and details of all the ES filters, expanding on a previous survey in 1985 [111]. Exponential smoothing models are equivalent to state-space models [112] even though they do not require knowledge of the underlying system. The first statistical foundation of simple ES was provided by Muth [223] who demonstrated that, given a random walk plus noise, ES gave the optimal estimates of the underlying system.

Each model assumes trend and seasonality and handles the trend in one of five different ways and seasonality in one of three. The trend is considered to be either non-existent (none), additive, damped additive, multiplicative, or damped multiplicative. The seasonal patterns are expected to be either non-existent (none), additive or multiplicative [112, 162, 264].

The most basic exponential smoothing filter assumes no trend nor seasonality and is frequently used as a stock market filter under the name exponential moving average [156, 215]. The price at time $t$ ($p_t$) is filtered to $\tilde{p}_t$ by:

$$\tilde{p}_t = \alpha p_t + (1 - \alpha)\tilde{p}_{t-1},$$

where $\alpha$ is the smoothing factor between 0 and 1.

The models are all recursive models and the specifications do not indicate the starting values. Various methods are used by other researchers such as Kalekar [172] who suggests that the trend should be initialized to the first change in the data ($p_2 - p_1$) or estimated by various averaging methods. A large study [211] set initial values by backcasting, least squares and simple methods such as setting to 0.

Exponential smoothing can also be used for extrapolation. Forecasts provided by these models have been examined and confirmed by many studies such as [61, 245]. Exponential smoothing is commonly used for
sales forecasting, inventory and other business related financial data [78]. For example, Ashuri & Lu [18] applied exponential smoothing to construction costs and Corberán-Vallet, Bermúdez, & Vercher [70] forecast the Industrial Production Index. Recently, ES has been combined with neural networks to produce superior forecasts on the stock market as in [77, 210].

**Linear filters** are an abundant and flexible class. They take the form of:

\[ \tilde{y}_t = \sum_{i=1}^{n} w_i y_{t-i-n+i}, \]  

(1.10)

where \( w = [w_1, \ldots, w_n] \) are the filter weights and \( l \) is the lag. Each model has a different method of deriving the weights. Linear filters are also known as finite impulse response (FIR) filters in the signal processing field.

Shiu [256] wrote that, for more than 150 years, actuaries have been using moving averages to smooth time series. Indeed, the earliest recorded use was by John Finlaison in 1829 for smoothing mortality rates [143]. One early form was developed by Spencer [260, 261, 272] who introduced a 21-term moving average. Shiu [256] defined smooth as having small \( d^{th} \) differences for \( d \in \mathbb{N} \) and derived a matrix derivation for the minimum-\( R_d \) moving weighted average formula with the constraint that the model was exact for polynomials up to a certain degree \( r \). However, in practice, much simpler models are employed.

The first is the simple moving average (SMA) [89]. This filter assigns \( w_1 = w_2 = \cdots = w_n = 1/n \) - literally a moving average. The output of the SMA at time \( t \) is a better estimate for the data at approximately \( t-n/2 \). So, analysts changed the coefficients to provide more emphasis on the most recent data, thus reducing the lag [215]. The weighted moving average (WMA) [89] sets the vector \( w = [1, 2, \ldots, n] \cdot (n(n+1)/2)^{-1} \). The Hull moving average (HMA) [157] is a modification of the WMA which is also designed to have less lag. Given that WMA\((y, n)\) is the WMA of series \( y \) with \( n \) coefficients, the HMA is calculated as:

\[ s = \text{WMA}(y, \text{round}(n/2)), \]
\[ l = \text{WMA}(y, n), \]
\[ \tilde{y} = \text{WMA}(s + (s - 1), \text{round} [\sqrt{n}]). \]

A Gaussian implementation is where the coefficients \( w \) are selected from a Gaussian kernel [130]:

\[ K_\sigma(x) = e^{-\frac{x^2}{2\sigma^2}}, \quad (1.11) \]
\[ \hat{y}_t = \frac{\sum_{i=0}^{N-1} K_\sigma(i) y_{t-i}}{\sum_{i=0}^{n-1} K_\sigma(i)}. \quad (1.12) \]

The use of the Gaussian function is gaining weight in the trader community with the introduction of the Arnaud Legoux Moving Average (ALMA) [196] which uses a given offset \( O \in \mathbb{R} \):

\[ \hat{y}_t = \frac{\sum_{i=0}^{n-1} K_\sigma(i - O) y_{t-i}}{\sum_{i=0}^{n-1} K_\sigma(i - O)}. \]

FIR filters have been developed in other literature for similar purposes. The Savitzky-Golay filter [247], for example, was developed in the chemistry literature. For each interval of size \( n \) a polynomial is fitted and the center value is set as the smoothed value of the center of the interval.

Alistair Gray and Peter Thomson have contributed a huge amount of work to the area of linear filtering. They issued a report in 1997 [125], and a further publication in [126]. They designed linear filters which maximise the amount of smoothing. Unfortunately, these filters have not been developed for asset price smoothing. The filters are based on a general white noise process, while price changes are close to a Gaussian process, suggesting a simpler model. Their model requires four parameters. Also, their model is developed under a polynomial constraint which may restrict better smoothing ability.

Adaptive filters are a class of filters where the model parameters are updated as new data is received. Some of the basic models have been modified in this manner. Ehlers tried various different values for the FIR coefficients [89], attempting to achieve acceptable noise reduction for the delay trade-off. One attempt was to set the weight \( w \) of a price \( p_t \) to \( w = p_t - p_{t-m} \) and then to normalize the coefficients to sum to one in
the FIR window. Another was to calculate the weight for a price at time $t$ as $w = \sum_{i=t-n}^{t} (p_i - p_t)^2$, where $n$ is the FIR window size, and again, the weights are normalized to sum to one. Ehlers [89] changed these FIR filters to have an infinite response. The weight ($w$) of a price ($p_t$) is set to $w = p_t - \hat{p}_{t-1}$, and then the coefficients are normalized to sum to one in the FIR window.

There are a few algorithms for turning exponential smoothing (ES) models into adaptive filters. Kaufman’s adaptive moving average (KAMA) is an adaptive N-N ES model [175], providing a calculation of $\alpha$ at each time step:

$$\alpha_t = (E_t(\alpha_L - \alpha_S) + \alpha_S)^2,$$

$$E_t = \frac{\sum_{i=0}^{N} |y_{t-i} - y_{t-N}|}{\sum_{i=0}^{N} |y_{t-i} - y_{t-i-1}|},$$

where $\alpha_L$ is the largest desired value for $\alpha$, $\alpha_S$ is the smallest, and $N$ is $E$’s window width. The variable index dynamic average (VIDYA) [59] is another adaptive ES model in the trading literature based on volatility:

$$\hat{p}_t = a \frac{s_n}{s_m} p_t + (1 - a \frac{s_n}{s_m})\hat{p}_{t-1},$$

where $s_n$ and $s_m$ are the standard deviations of prices over the last $n$ and $m$ periods respectively.

**Signal processing** is rich with filters. These are designed to remove frequencies from a stationary oscillating signal. The problem when using these models on stock prices is that financial data is non-stationary. However, while they are derived for different purposes, some signal processing filters are linear. For example, the Parks McClellan FIR filter [236] finds the FIR coefficients that provide the closest approximation to a given frequency response. Another notable filter is the Butterworth filter [250] which takes a frequency threshold and builds the optimal filter to remove the frequencies above this threshold. Because they focus on frequencies rather than arbitrary noise reduction, these filters will not receive further consideration.
1.3.2.3 Concluding Remarks

A surprising finding is that, despite the aim for a smooth output, no moving average model in the financial literature has been designed to do this as best as possible. The ALMA filter does have a parameter that will alter the smoothness, however, the filter weights are selected from a Gaussian distribution presumably because the curve appears smoother to the naked eye. While there is no inherent problem with this, I believe a better linear smoother can be built. After all, if we want to discover the relationship between smoothness and returns, we will need to be able to find the maximum amount of smoothing possible.

All online time series noise reducers lag behind the price. That is, there is a delay between when a turning point occurs in the price and when it appears in the smooth output. The major hindrance of moving averages is their lag. Yet, surprisingly, the studies I have reviewed do not measure lag. Lag is a well defined and measured concept in signal processing. In this field, the lag of individual frequencies are the issue and measured with group delay and phase delay. If the filter is symmetric or anti-symmetric, all frequencies are delayed by the same amount, this provides a suitable measure. However, not all moving averages are symmetric or anti-symmetric. In technical analysis, the lag of the entire series is the problem, not separate frequencies. The lag of some filters are known, for example the SMA and WMA, though, methods of calculating lag for any filter do not seem to be available.

The next section will look at how online filters have been used to invest money in financial markets.

1.3.3 Moving Averages as Technical Analysis

This section will look at the most popular trading strategy for moving averages. All studies examined used only this strategy; no others were discussed. The strategy is called the moving average cross-over rule. In other literature it is called the moving average convergence divergence system. In this thesis, we will call this rule the dual filter rule (DFR) because it uses two filters – a distinguishing feature which will become important in later chapters.
The DFR strategy uses the cross-over between two filters as the signal to buy or sell. The filter with the lesser lag is called the short filter, the other, the long filter. Figure 1.2 provides an example of this system. When the short MA moves below the long one, close any open trades and then sell. When the short MA moves above the long, close any open trades and then buy.

![Dual filter rule example](image)

**Figure 1.2:** Dual filter rule example. The simple moving average is used in this example. SMA(10) means a SMA of size 10. When the short (red) filter moves below the long (blue), sell and vice versa.

The literature also examines two modifications. The version just discussed and shown in Figure 1.2 is called the variable moving average trading rules. The “variable” refers to the amount of time in each trade; the time between each buy and sell varies. Alternatively, in the fixed moving average rule, the amount of time spent in a trade is set, most often to 10 days [45, 129, 233]. For example, when a buy signal is generated, the trader buys into the asset and sells again after 10 days, ignoring all buy or sell signals during the trade.

Finally, because the short moving average may bounce above and below the long filter repeatedly, some studies place a threshold around the long band, often of 1%. For example, a buy signal is only generated if the short filter moves above the long filter plus 1%.

This DFR rule with the variable (VMA), fixed (FMA) time and band
modifications are the only ones considered in the literature. William Brock, Josef Lakonishok and Blake LeBaron in 1992 [45] examined the dual moving average strategy with the SMA, testing both VMA and FMA (10 days) with bands of 0% or 1%. They tested the Dow Jones Industrial Average from 1897 to 1986. They concluded that the rule does have predictive power. However, there are a few problems with this study. They did not include transaction costs which may negate all profits. They tested only 5 combinations of moving averages (1-50, 1-150, 5-150, 1-200, and 2-200) which they claim to be popular. They could be popular due to data-mining bias. The study was duplicated in [155] in the United Kingdom markets. The conclusions were the same.

Other researchers include Parisi & Vasquez [233] who conducted similar experiments to Brook, Lakonishok, & LeBaron [45] on 10 years of a Chilean stock market index reporting positive results. They also ignored transaction costs and only examined five filter combinations. Gunasekarage & Power [129] tested the SMA-based VMA and FMA with bands on ten years of four emerging south Asian stock exchanges. With only five filter combinations and no transaction costs, the results were better than a benchmark naive strategy. Ellis & Parbery [94] investigated the KAMA with bands optimising the parameters on a training set and recording results on a separate test set. The Australian All Ordinaries, DJIA and S&P 500 indices were used with a transaction cost of 1%. Unfortunately, optimisation provided no benefits and all excess profits disappeared with transaction costs. Ming-Ming & Siok-Hwa [221] tested the SMA with all three rules on nine daily Asian stock market indices from 1988-2003. Transaction costs were included and they reported excellent results. However, their actual results were mixed and they based their conclusions on the best results, rendering them useless. El-khodary [91] builds a decision support system with the SMA and EMA with the base dual filter rule on 10 securities from the Egyptian Stock Exchange. Again, no transaction costs were included and the best results were used as the conclusion ignoring the mixed results. Cai et al. [53] tried using a neural network to enhance the SMA 5-10 DFR on 10 Emerging equity markets in Latin America and Asia from 1982 - 1995. They reported mixed results comparing the model’s performance against random investing. Mixed or negative results are precisely to be expected if the strategy has no advantages. Hsu et al. [150] also enhanced the
strategy, this time with a particle swarm optimisation algorithm. There were 413 Taiwanese mutual funds covering approximately 10 years. The algorithm was trained on past data and then tested on one year of future prices. The process was repeated to cover 10 years of trading. Even with transaction costs and testing over the 2007-2008 global financial crises, they reported positive results.

The full set of references in this review that include moving averages for trading are: \[30, 45, 53, 91, 94, 129, 132, 150, 155, 221, 233\].

Two things are striking from previous studies on the dual filter rule. Firstly, ignoring complex optimisation algorithms, researchers only test a small subset of moving average combinations. For example, only five are tested in [45]. Bigalow & Elliot [30] even claim that technical analysts consider the 50-200 moving average rules to be the most profitable. Yet, no reason is given. The claimed superiority of this rule may be pure chance. The small search space might be due to the high dimension and computational complexity of testing all combinations of two filters.

Secondly, there are no attempts to explore the effects of smoothing on returns. This is remarkably surprising as the purpose of smoothing is to highlight trends to enhance returns.

1.4 Summary

This chapter has explored a huge amount of research from the theoretical possibility of investing to time series analysis and noise reduction to specific trading strategies.

In section 1.1 we learned that evidence for and against the weak-form market is not only inconsistent, but conflicting. This is best illustrated by the two studies [33] & [168] which conducted the same experiment with opposite conclusions. After learning about time series and technical analysis in section 1.2 I suggested that a full exploration of the results are needed to ensure consistency. That is, to figure out why something is or is not profitable. Later in this thesis, when we look at trading with filters in chapter 4, I will show clear relationships in the data that show exactly what is happening with the moving averages and what they are
Section 1.3 found that linear filters are the most practical for price smoothing. However, with the exception of the ALMA, there are no filters designed to actively control smoothing. The ALMA itself does not appear to be quantitatively built. In chapter 2 I will design a new filter which does quantitatively control smoothing for an asset price following a stochastic process. Chapter 2 will also develop a new measure of time series smoothness as no viable metrics were present in the literature. A means of measuring lag will also be presented.

Section 1.3.3 found that researchers test moving average trading strategies using two filters and only a very small subset of parameters. They claim that their selected parameters are popular. However, I do suspect that the small search space might be due to the high dimension of testing all possible parameters. In chapter 4 when we look at the strategy, I will show you how, for our specific purposes, it has a much smaller number of parameters. We will use this to explore a large area of possible parameters from which we can draw solid conclusions.

Finally, the motivating curiosity of this thesis is: what is the relationship between smoothing and profits? A question not answered by the literature and a question that will be answered in chapter 4.
Chapter 2

Smoothing Stock Prices

*Our experience in the financial arena has taught us to be very humble in applying mathematics to markets, and to be extremely wary of ambitious theories, which are in the end trying to model human behavior. We like simplicity, but we like to remember that it is our models that are simple, not the world.*

– Emanuel Derman & Paul Wilmott (2009) in [84]

Working on a stock portfolio is like working on a construction site. Your job is to build a massive long lasting monument or asset for the project’s beneficiaries. However, chaos rules the work ground. Shouting across the carnivorous pit expands a sea of men wearing the same safety clothes and hard hats. Your brain rattles, bruises and aches against the violent sounds of huge engines. Instead of florescent uniforms, investment has smart suites. Super-computer arrays sitting close to the exchange are the industry’s heavy machinery. While the deafening sounds are mirrored by perplexing noisy stock prices. In this chapter, we are going to develop a filter that reduces the noise in stock prices to give a clearer picture of its behaviour.

For the purpose of this thesis and as discussed in Chapter 1 a filter takes a time series and removes some unwanted component. For financial
prices, we want to remove noisy fluctuations and see only a steady smooth curve. The idea is to clearly see where the price is heading and potentially make a profit.

The literature review discovered that while there were many published studies on the profitability of moving averages, the actual filters are limited. There is little mathematical focus on removing insignificant fluctuations (smoothing) in the financial texts. The models are linear and smoothness is a linear concept. It seems intuitive to include a parameter that controls smoothness. This chapter will design filters that are capable of such control.

While there appears to be many non-linear models for filtering, indeed the exponential moving average (EMA) [215] is very popular among traders, the majority of filters used in practice are linear filters. Examples are, the Simple Moving Average (SMA) [29], Weighted Moving Average (WMA), Hull Moving Average (HMA) [157], and the Gaussian distribution based filter the Arnaud Legoux Moving Average (ALMA) [196]. As the literature shows no substantial evidence that non-linear models perform better in financial time series analysis, this thesis will be limited to the derivation of a linear filter.

Using the filters developed in this chapter, we can control the amount of smoothing with the option of maximising the smoothness. The family of filters is split into three types: (1) a filter which maximises the smoothness, termed the Linear Gaussian Smoother (LGS); (2) a generalisation of the LGS where the user can control the amount of smoothing, called the Symmetric Linear Gaussian Smoother (SLGS); (3) a filter identical to the SLGS except that it is no longer symmetric and is weighted towards the latest prices: Asymmetric Linear Gaussian Smoother (ALGS).

As we progress through this chapter I will contribute:

1. A measure of time series smoothness called the Smoothness Index (SI) (Section 2.3.1.1).

2. The expected SI of any linear filter given some basic assumptions about asset prices (Section 2.3.1.2).

3. The expected lag of any linear filter given the same basic assumptions (Section 2.3.2.1).
4. The expected error (fit) between any linear filter and an input time series that follows some basic assumptions (Section 2.3.4.3).

5. A filter which maximises smoothness after learning the patterns in any input time series (Section 2.4.1).

6. The expected filter that maximises smoothness given some basic assumptions about the input time series (Section 2.4.2).

7. Experiments and their results using the contributed measures of smoothness, lag and fit to compare different moving averages and the smoother just developed (Section 2.6).

First, we will discuss in more detail the purpose and my goals of filtering in Section 2.1. Then, I will define the basic assumptions that we will later rely on and present the real world time series that we will use for later experiments in Section 2.2. We will cover the metrics in Section 2.3, the new filter in Section 2.4, the three filter types in Section 2.5 and the experiments in Section 2.6.

## 2.1 Filter Purpose & Goal

This section does not aim to provide a rigorous definition of what I hope to achieve. Rather, here I will discuss the overall focus. Generally speaking, what do we want from a filter and what does this mean mathematically?

Here is the situation, asset prices fluctuate wildly yet there still appears to be trends of prolonged up and down movement. We would like to take advantage of these trends, but first we need to be able to identify them. Our choice of method is to filter the prices removing as much variability as possible while retaining the trends. We can say, then, that removing noisy fluctuations is the sole purpose of any filter we may be interested in.

Notice in Figure 2.1 the result of applying different filters. Each one, to varying degrees, appears more “smoother” than the original prices. In addition, offline time series noise reduction models such as those based on wavelets [204], EMD [35], SVD [138] and least squares [274] all produce a
“smooth” curve. This is an idea that we can use. “Smooth” has been well defined since at least the early 1900s with the work of Whittaker [274]. A smooth time series has small derivatives. Thus, mathematically, the goal of a filter is to output a time series with derivatives smaller than the input series.

Note that I do not mean that “smooth” is an infinitely differentiable function nor a function with bounded derivatives. Neither have I defined how much smoother the filtered prices ought to be; only that they should be smoother. This gives us room later on to change the level of smoothing and see how it affects profits. Section 2.3.1 defines this more rigorously.

Demanding more smoothing does come at a cost of lag. Lag is the time delay between the prices and the filter’s best estimate. For example, if the filter output at time $t$ is the best estimate for time $t - l$, then $l$ is the lag. A filter of less smoothing but the same amount of lag ought to be calculable. After all, one can simply lower the filtering, and manually shift the output to increase the lag back to the original level. We can then logically say that lag places an upper limit on the amount of smoothing but not a lower limit. A filter with a parameter for smoothness that is independent of lag, except for the imposed upper limit, would be the closest we can get to separating smoothing and lag with a linear filter. Ultimately, as we will later be looking at the relationship between smoothing and profits, we want to be able to achieve the greatest amount of smoothing possible for any given lag.

![Figure 2.1: Example of smooth filters on the EUR/USD currency pair.](image-url)
2.2 Datasets & Data Model

Historical prices from fourteen different financial instruments were used throughout this thesis. There are six currency pairs, five stocks, two indices and gold. A list of the assets, their interval, length, start and end dates are in Table 2.1. A larger amount of securities could have been used. For example, Hsu et al. [150] used over 400 assets. However, the results to come are consistent across each time series. Also, many studies only use a small number of assets. For example, [45, 63] both use only one index and [193] only uses two currencies.

The most common and basic model for security prices follows a geometric Brownian motion [276]. For our purposes, we can write the return simply as a Gaussian random variable:

\[
\frac{p_t - p_{t-1}}{p_{t-1}} = \phi_t, \quad \phi \sim \mathcal{N}(\mu^*, \sigma^2),
\]

\[
p_t = p_{t-1}(1 + \phi_t).
\]

People make this model easier to work with by using Ito’s lemma to derive [160]:

\[
y_t = \log(p_t),
\]

\[
y_t = y_{t-1} + \epsilon_t, \quad \epsilon \sim \mathcal{N}\left(\mu^* - \frac{\sigma^2}{2}, \sigma^2\right) = \mathcal{N}(\mu, \sigma^2),
\]

that is, today’s logged price is yesterday’s logged price plus a Gaussian random variable with mean \(\mu = \mu^* - \frac{\sigma^2}{2}\) and variance \(\sigma^2\). While it is known that returns do not follow a Gaussian distribution [34], it is very close and works reasonably well as will be seen.

I make one small change to this model by requiring that:

\[
\mu \approx 0,
\]

or a weaker version:

\[
\mu^2 < \sigma^2.
\]

these two assumptions are made to ease some derivations later on. Clearly,
the model in (2.1) and the conditions (2.2) & (2.3) are very unreasonable if we expect to trade and turn a profit. They also do not conform to the discussion in section 2.1 as they do not reflect changing trends. This model is merely a very high level abstraction of reality which we will use to simplify filters, their derivation and properties. We will rely on a trick of smoothing to ensure that trends are highlighted.

To test equations (2.1) & (2.2) we can perform two statistical tests. Equations (2.1) & (2.2) can be checked simultaneously with a one sample t-test that the data comes from a normal distribution with mean zero and unknown variance. Equation (2.1) can be rigorously checked with the Anderson-Darling goodness-of-fit test [11] that the distribution is normal. This test places more weight on the tails of the distribution.

The p-values are displayed in Table 2.2 for each of the time series. The p-values for the t-test are all above 0.01 giving statistical support that the distribution is normal with a mean of zero. However, the p-values for the Anderson-Darling test are below 0.001 demonstrating that despite the assumption of normality, it is indeed incorrect.
<table>
<thead>
<tr>
<th>Name</th>
<th>Code</th>
<th>Interval</th>
<th>Length</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUR/USD</td>
<td>EURUSD</td>
<td>Daily</td>
<td>2000</td>
<td>7.94 Years</td>
<td>15-Oct-2003 00:00</td>
</tr>
<tr>
<td>Google</td>
<td>GOOG</td>
<td>Daily</td>
<td>1720</td>
<td>6.83 Years</td>
<td>25-Oct-2004 00:00</td>
</tr>
<tr>
<td>Dow Jones Industrial Average</td>
<td>INDU</td>
<td>Daily</td>
<td>2000</td>
<td>7.94 Years</td>
<td>21-Jul-2003 00:00</td>
</tr>
<tr>
<td>NASDAQ</td>
<td>NASDAQ</td>
<td>Daily</td>
<td>2000</td>
<td>7.94 Years</td>
<td>10-Apr-2003 00:00</td>
</tr>
<tr>
<td>XAU/USD</td>
<td>XAUUSD</td>
<td>Daily</td>
<td>2000</td>
<td>7.94 Years</td>
<td>22-Oct-2003 00:00</td>
</tr>
<tr>
<td>AUD/JPY</td>
<td>AUDJPY</td>
<td>Hourly</td>
<td>2000</td>
<td>83.33 Days</td>
<td>18-May-2011 07:00</td>
</tr>
<tr>
<td>EUR/CHF</td>
<td>EUR/HF</td>
<td>Hourly</td>
<td>2000</td>
<td>83.33 Days</td>
<td>18-Feb-2011 07:00</td>
</tr>
<tr>
<td>USD/CHF</td>
<td>USDCHF</td>
<td>10 Minute</td>
<td>2000</td>
<td>13.89 Days</td>
<td>26-May-2011 05:50</td>
</tr>
<tr>
<td>USD/JPY</td>
<td>USDJPY</td>
<td>10 Minute</td>
<td>2000</td>
<td>13.89 Days</td>
<td>26-May-2011 05:50</td>
</tr>
<tr>
<td>Westpac Banking Corp.</td>
<td>WBC</td>
<td>Daily</td>
<td>2105</td>
<td>8.35 Years</td>
<td>01-Jan-2005 11:00</td>
</tr>
<tr>
<td>AUD/USD</td>
<td>AUDUSD</td>
<td>Daily</td>
<td>2107</td>
<td>8.36 Years</td>
<td>01-Jan-2005 11:00</td>
</tr>
<tr>
<td>BHP Billiton Limited</td>
<td>BHP</td>
<td>Daily</td>
<td>2107</td>
<td>8.36 Years</td>
<td>01-Jan-2005 11:00</td>
</tr>
<tr>
<td>Exxon Mobil Corporation</td>
<td>XOM</td>
<td>Daily</td>
<td>2107</td>
<td>8.36 Years</td>
<td>01-Jan-2005 11:00</td>
</tr>
<tr>
<td>Walt Disney</td>
<td>DIS</td>
<td>Daily</td>
<td>2186</td>
<td>8.67 Years</td>
<td>31-Dec-2004 11:00</td>
</tr>
</tbody>
</table>

Table 2.1: All the datasets used in this thesis.
## Chapter 2. Smoothing Stock Prices

### Table 2.2: Testing properties of each time series. The table shows the p-values for two tests on each dataset. T is the one sample t-test that the distribution is normal with a mean of zero and unknown variance. A-D is the Anderson Darling test that the distribution is normal.

<table>
<thead>
<tr>
<th>Data</th>
<th>T</th>
<th>A-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUR/USD</td>
<td>0.468</td>
<td>p &lt; 0.001</td>
</tr>
<tr>
<td>Google</td>
<td>0.457</td>
<td>p &lt; 0.001</td>
</tr>
<tr>
<td>Dow Jones Industrial Average</td>
<td>0.587</td>
<td>p &lt; 0.001</td>
</tr>
<tr>
<td>NASDAQ</td>
<td>0.273</td>
<td>p &lt; 0.001</td>
</tr>
<tr>
<td>XAU/USD</td>
<td>0.013</td>
<td>p &lt; 0.001</td>
</tr>
<tr>
<td>AUD/JPY</td>
<td>0.479</td>
<td>p &lt; 0.001</td>
</tr>
<tr>
<td>EUR/CHF</td>
<td>0.316</td>
<td>p &lt; 0.001</td>
</tr>
<tr>
<td>USD/CHF</td>
<td>0.276</td>
<td>p &lt; 0.001</td>
</tr>
<tr>
<td>USD/JPY</td>
<td>0.494</td>
<td>p &lt; 0.001</td>
</tr>
<tr>
<td>Westpac Banking Corp.</td>
<td>0.656</td>
<td>p &lt; 0.001</td>
</tr>
<tr>
<td>AUD/USD</td>
<td>0.478</td>
<td>p &lt; 0.001</td>
</tr>
<tr>
<td>BHP Billiton Limited</td>
<td>0.360</td>
<td>p &lt; 0.001</td>
</tr>
<tr>
<td>Exxon Mobil Corporation</td>
<td>0.437</td>
<td>p &lt; 0.001</td>
</tr>
<tr>
<td>Walt Disney</td>
<td>0.268</td>
<td>p &lt; 0.001</td>
</tr>
</tbody>
</table>
2.3 Metrics

The two properties of a filter we are most interested in is the smoothness and lag. Though, we also need the distance between the filter’s output and the original prices to be as small as possible. This is called fitness. After all, a filter that does not resemble the prices is of little use.

This section discusses the details of these three properties, derives a metric to measure them and the expected value of these metrics given the data follows (2.1).

2.3.1 Smoothness

The literature review discovered that the most widely used method for measuring the level of noise reduction is the signal to noise ratio (SNR) in (1.5). The SNR assumes that the original noise-free signal is known, and assumes that the filtered series has no lag.

Despite the SNR being unfit for financial data, there are two important considerations to be made. First of all, it acts as a measure of predictability of the filtered dataset. When dealing with time series analysis, it is sometimes assumed that the clean time series is completely deterministic. Given this assumption, the point by point comparison between the filtered data and the clean data in (1.5) measures predictability.

Second, measuring the variance of this error series is an easy way of quantifying the level of raw noise in the data. It effectively measures the fluctuations. Considering that the clean signal is noise free, the SNR compares the raw noise level to the clean signal to reach a final measure of the level of noise in the series. By calculating the variance of both the clean data and the error of the filtered data, it can be determined how much noise has been removed.

The important thing to remember with the SNR is that it is a single value which describes the level of noise on a scale independent to the time series in question.
2.3.1.1 Measuring Smoothness

In section 2.1 I talked about how the filter output needs to be smoother than the input prices. By smoother, I mean has smaller derivatives. The measure for smoothness is borrowed from Whittaker [274] who poses the metric as:

$$||D_{N,d}y||^2,$$  \hfill (2.4)

where $y$ is a time series vector of size $N$ and $D_d$ is the differencing matrix such that $D_{N,d}y$ is a time series of the $d^{th}$ difference of $y$. The size of the vector, $N$, is often dropped. The matrix $D_{N,d}$ has dimension $(N-d) \times N$. Examples of $D_d$ include:

$$D_{5,1} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix},$$  \hfill (2.5)

$$D_{5,2} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}. \hfill (2.6)$$

The measure for smoothness builds upon $||D_d y||^2$ with several considerations. If $d = 1$, then this measures the error when using the previous value of $y$ as a forecaster. Similarly, $d = 2$ is using the previous 1st derivative to forecast. Clearly, the smoothness is a measure of forecasting error when using yesterday’s $(d-1)^{th}$ derivative to forecast today’s value. Notice that the 1st derivative may be smooth in which case $d = 2$ would result in a smaller error making forecasting 1-step ahead with $d = 2$ superior to $d = 1$. Thus, the smoothness of an arbitrary time series $y$ is the minimum value of (2.4) with respect to $d$. The function $S(y)_d$ is the smoothness of $y$ at $d$:

$$S_d(y) = \frac{1}{N-d}||D_d y||^2,$$

where $N$ is the length of $y$. The function $S(y)$ is the smoothness of $y$:

$$S(y) = \min_{d \in \mathbb{N}} \{S_d(y)\}.$$
The Smoothness Ratio (SR) is the smoothness of the filtered data $\tilde{y}$ normalized by the smoothness of the noisy data $y$:

$$SR(y, \tilde{y}) = 1 - \frac{S(\tilde{y})}{S(y)},$$

which can be interpreted as the percentage of noise filtered from the original series $y$ to produce the smooth curve $\tilde{y}$. Just like the SNR, this measure quantifies relative level of noise. But unlike the SNR, the SR does not assume that the clean signal is known, and does not make assumptions about the lag.

The smoothness ratio is also similar to the SNR in that it utilizes the variance of the data:

$$SR(y, \tilde{y}) = 1 - \frac{\text{var}(D_{d_1}\tilde{y})}{\text{var}(D_{d_2}y)}.$$

Because differencing removes the mean, the assumption that $\mu \approx 0$ must be held if either $d_1 = 1$ or $d_2 = 1$. At $d = 1$ the mean is still present in the data and $\|D_d y\|^2$ does not remove the mean as $\text{var}(D_d y)$ does.

One further step is taken. When comparing different models by plotting lag on the x-axis and smoothness on the y-axis, there is very little space between the curves making them hard to distinguish. Fortunately, the curves roughly follow a hyperbolic tangent path allowing the inverse hyperbolic tangent function to be used to separate the curves. i.e.:

$$\tanh^{-1}(x) = \frac{1}{2} \ln \left( \frac{1 + x}{1 - x} \right).$$

The final measure of smoothness is called the **Smoothness Index (SI)** and is defined as a function of a time series $y$ and its filtered version $\tilde{y}$:

$$SI(y, \tilde{y}) = \tanh^{-1}(SR(y, \tilde{y}))$$

$$= \frac{1}{2} \left[ \ln(1 + SR(y, \tilde{y})) - \ln(1 - SR(y, \tilde{y})) \right]$$

$$= \frac{1}{2} \left[ \ln(2 - \frac{S(\tilde{y})}{S(y)}) - \ln\left(\frac{S(\tilde{y})}{S(y)}\right) \right].$$

If the series $\tilde{y}$ is a polynomial, then at some derivative $d$, $S_d(\tilde{y}) = 0$. 
If we assume that $S(\tilde{y}) > 0$, then we can simplify to:

$$SI(y, \tilde{y}) = \frac{1}{2} \ln \left( \frac{2 - S(\tilde{y})/S(y)}{S(\tilde{y})/S(y)} \right) = \frac{1}{2} \ln \left( \frac{2}{S(\tilde{y})/S(y) - 1} \right) = \frac{1}{2} \ln \left( \frac{2S(y)}{S(\tilde{y})} - 1 \right).$$

This equation is highlighted in (2.7).

**Important Equation(s) 1 (Smoothness Index)**

We can measure the smoothness of a time series $y$ that has been filtered into $\tilde{y}$ with the Smoothness Index (SI):

$$SI(y, \tilde{y}) = \frac{1}{2} \ln \left( \frac{2S(y)}{S(\tilde{y})} - 1 \right). \quad (2.7)$$

Figure 2.2 demonstrates this measure before and after $\tanh^{-1}$ is applied. Notice that the curves are close together and indistinguishable from each other. However, the data is now separated and clearly distinguishable.

![Figure 2.2: Smoothness ratio vs smoothness index. An example of the SR before and after $\tanh^{-1}$ is used to separate the curves. The curves here are real examples of filters.](image)

### 2.3.1.2 Expected Smoothness

As stated, equation (2.7) is a model independent data-driven metric. That is, the actual filter does not matter. Only a price series and the filter's output are needed to calculate the SI. The following theorem,
lemma and corollaries demonstrate a model-driven data independent approach, otherwise known as parametric. This is achieved by deriving the expected value of the SI which does not require knowledge of the dataset. The advantage of doing this is to avoid processing large amounts of data. A parametric method provides a fast and accurate way of measuring the SI of a filter.

To derive this parametric method the filter is assumed to be linear, the time series is assumed to follow equation (2.1) and we are operating on the logged time series. Thus, the changes in the input series are Gaussian distributed.

First, the differencing vector \( (d_d) \) must be properly defined as the windowed vector that is moved through the matrix \( D_d \). Following from the examples in (2.5) & (2.6):

\[
d_1 = [-1, 1,]
\]
\[
d_2 = [1, -2, 1].
\]

The elements of vector \( d_d \) can be mathematically described as:

\[
d_{d,i}^* = {d \choose i} (-1)^{d-i},
\]

\[
\sum_{i=0}^{d} d_{d,i}^* = 0,
\]

(2.8)

where \( d_{d,i}^* \) is the \textsuperscript{i}th element of \( d_d \), \( i \) iterates from 0 to \( d \), and \( \binom{n}{i} \) is a binomial coefficient.

The next lemma, theorem and corollaries derive the expected smoothness of logged asset prices.

**Lemma 1.** The expected smoothness of a time series \( y_t = y_{t-1} + \epsilon_t \) where \( \epsilon \sim \mathcal{N}(\mu, \sigma^2) \) at \( d \) is equal to:

\[
E[S_d(y)] = \begin{cases} 
\mu^2 + \sigma^2 & d = 1 \\
||d_{d-1}||^2 \sigma^2 & d > 1
\end{cases},
\]

(2.9)

and is undefined for \( d < 1 \).
Proof. The smoothness is measured by:

\[ E[S_d(y)] = E[\left|\frac{D_d y}{N - d}\right|^2] = E[(\nabla^d y_t)^2], \]

where \( N - d \) is the number of rows of \( D_d y \). In the case of \( d = 1 \),
\[ E[(\nabla^1 y_t)^2] = E[\epsilon_t^2] = \mu^2 + \sigma^2. \]

When \( d > 1 \) something different happens. We can write:

\[ E[(\nabla^d y_t)^2] = E[(d^T y_{d+1,t})^2] = E[(d^T_{d-1} \epsilon_{d,t})^2], \]

where \( y_{d+1,t} \) is a vector of the last \( d + 1 \) values of \( y \) ending with \( y_t \) and \( \epsilon_{d,t} \) is a vector of the last \( d \) values of \( \epsilon \) ending with \( \epsilon_t \). Moving forward:

\[ E[(d^T_{d-1} \epsilon_{d,t})^2] = E[d^T_{d-1} \epsilon_{d,t} \epsilon^T_{d,t} d_{d-1}] = d^T_{d-1} E[\epsilon_{d,t} \epsilon^T_{d,t}] d_{d-1} = d^T_{d-1} [\sigma^2 I + \mu^2 J] d_{d-1} = \sigma^2 d^T_{d-1} d_{d-1} + \mu^2 d^T_{d-1} J d_{d-1}, \]

remembering (2.8), \( J d_{d-1} \) is a vector of zeros. Thus, this reduces to:

\[ \sigma^2 d^T_{d-1} d_{d-1} = ||d_{d-1}||^2 \sigma^2. \]

\[ \square \]

**Corollary 1.** The expected smoothness of a time series \( y_t = y_{t-1} + \epsilon_t \) where \( \epsilon \sim N(\mu, \sigma^2) \) and \( \mu^2 < \sigma^2 \) is equal to: \( E[S(y)] = \mu^2 + \sigma^2 \).

Proof. We have:

\[ E[S(y)] = \min_{d \in \mathbb{N}} \{ E[S_d(y)] \} \]

Lemma 1 gives us \( E[S_d(y)] \). Observe again equation (2.9). For all \( d > 1 \) it is easy to see that \( ||d_{d_1-1}||^2 \sigma^2 < ||d_{d_2-1}||^2 \sigma^2 \) where \( d_1 < d_2 \). As \( \mu^2 < \sigma^2 \), in the case where \( d_1 = 1 \) & \( d_2 = 2 \) it stands that \( \mu^2 + \sigma^2 < 2 \sigma^2 \). \[ \square \]

Now, the expected smoothness of the filtered time series needs to be determined. First we need to know a new matrix. The matrix \( \overrightarrow{Y} \) is called
the trajectory matrix of the time series $y$. It is defined as:

$$\mathbf{Y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \\ y_2 & \cdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ y_{m-n+1} & \cdots & \cdots & y_m \end{bmatrix},$$

which is an $(m - n + 1) \times n$ matrix where $m$ is the length of $y$.

**Theorem 1.** The expected smoothness of a linear filter’s ($w$) output on a time series $y_t = y_{t-1} + \epsilon_t$ where $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$ and $\mu \approx 0$ at $d$ is equal to:

$$E\left[ S_d\left( \mathbf{Y}w \right) \right] = \sigma^2 ||D_{d-1}^T w||^2.$$

**Proof.** The smoothness is measured by:

$$E\left[ S_d\left( \mathbf{Y}w \right) \right] = E\left[ \frac{||D_d \mathbf{Y}w||^2}{N - d} \right] = E[(D_d \mathbf{Y}w)_i^2],$$

where $N$ is the number of rows of $\mathbf{Y}$. This can be written as:

$$E[(D_d \mathbf{Y}w)_i^2] = E\left[ (\epsilon^T D_{d-1}^T w)^2 \right],$$

where $\epsilon = [\ldots, \epsilon_{t-1}, \epsilon_t]$:

$$E\left[ (\epsilon^T D_{d-1}^T w)^2 \right] = E[w^T D_{d-1} \epsilon \epsilon^T D_{d-1}^T w] = w^T D_{d-1} (\sigma^2 I + \mu^2 J) D_{d-1}^T w = \sigma^2 w^T D_{d-1} D_{d-1}^T w$$

Since $\mu \approx 0$

$$= \sigma^2 ||D_{d-1}^T w||^2.$$

Finally, the expected smoothness index (SI) can be derived in the same way:

**Corollary 2.** The expected smoothness index of a filter $w$ on a time series $y_t = y_{t-1} + \epsilon_t$ where $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$ and $\mu \approx 0$ is equal to:

$$E[SI(y, \mathbf{Y}w)] = \frac{1}{2} \ln(2(\min_{d \in \mathbb{N}} \{ ||D_{d-1}^T w||^2 \})^{-1} - 1).$$
Proof.

\[ E[SI(y, \overrightarrow{Y}w)] = \frac{1}{2} \ln \left( 2 \frac{E[S(y)]}{E[S(\overrightarrow{Y}w)]} - 1 \right) \]
\[ = \frac{1}{2} \ln \left( 2 \frac{E[S(y)]}{\min_{d \in N} \{ E[S_d(\overrightarrow{Y}w)] \}} - 1 \right). \]

By Corollary 1 and Theorem 1 we have:

\[ E[SI(y, \overrightarrow{Y}w)] = \frac{1}{2} \ln \left( 2 \frac{\sigma^2}{\min_{d \in N} \{ \sigma^2 \|D_{d-1}^T w\|^2 \}} - 1 \right) \]
\[ = \frac{1}{2} \ln \left( 2 \frac{2}{\min_{d \in N} \{ \|D_{d-1}^T w\|^2 \}} - 1 \right). \]

The expected smoothness index is highlighted in (2.10).

**Important Equation(s)** 2 (Expected Smoothness Index)

We can measure the expected Smoothness Index (SI) of a time series \( y \) that follows the model (2.1), and has been filtered with a linear filter \( w \) with:

\[ E[SI(y, \overrightarrow{Y}w)] = \frac{1}{2} \ln \left( \frac{2}{\min_{d \in N} \{ \|D_{d-1}^T w\|^2 \}} - 1 \right). \]  \hspace{1cm} (2.10)

This will commonly be denoted as \( E[SI(w)] \) or as simply \( E[SI] \) if the filter is obvious.

Even though we have just walked through the derivation of (2.10) we can do a simple experiment to quickly see that it accurately models (2.7). If we generate random linear filters and random data series that conform to (2.1) we should see a strong correlation between the SI and \( E[SI] \). There is no need to statistically test the relationship as we already know that (2.10) is correct. I simple want to see that is it accurate.

The experiment proceeds as follows. (a) Generate a random vector of a random length between 2 & 50 from a standard normal distribution, take the cumulative sum, shift the result so that the minimum value is equal to zero and then normalise the vector to sum to one. This
is our filter weights. (b) Generate a random price series of size 2000 with changes drawn from a standard normal distribution. (c) Calculate and save the SI and $E[SI]$. Do (a) to (c) 1,000 times. Also, for each of the logged asset time series generate and apply a random filter 100 times saving the SI and $E[SI]$. The results are shown in Fig. 2.3. The experiment shows a strong correlation between the SI (2.7) and $E[SI]$ (2.10) on both the randomly generated data and the real world asset prices.

\[ E[SI] = SI \]

\[ E[SI] = SI \]

\[ E[SI] = SI \]

Figure 2.3: Correlation between the SI and its expected value. Filters were randomly generated and applied to both random time series and the real world time series in Section 2.2.

2.3.2 Lag

In a linear filter, lag can be very easily controlled by building a symmetric linear filter. For example, $l$ is the lag of the filter and the size ($n$) of the filter is $n = 2l + 1$. Unfortunately, the filters that investors use and some of the filters developed in this work do not conform to this notion of symmetry. An example is the weighted moving average (WMA). Thus, this section develops a data-driven method of calculating the lag as well as the expected lag.

Because we are not dealing with stationary signals nor data which we could consider to be a combination of sinusoids, I have chosen not to use the measures of group delay and phase delay from the signal processing literature. Instead, I will use a plain definition built on the least squared error.

Lag is defined here as the time offset between a time series and its most accurate estimate. If the estimate of time $t$ is actually the best
estimate of time \( t - l \), then \( l \) is the lag. So, \( \tilde{y}_t \) is the best estimate of \( y_{t-l} \) if, on average:

\[
(\tilde{y}_t - y_{t-l})^2 < (\tilde{y}_t - y_t)^2
\]

Then, to determine the lag of \( \tilde{y} \) we simply need to find the lag that minimises the mean squared error (MSE):

\[
\mathcal{L}(\tilde{y}) = \arg \min_{l=0,1,\ldots} \left\{ \frac{1}{T-l} \sum_{t=l+1}^{T} (\tilde{y}_t - y_{t-l})^2 \right\}
\]

The measure for lag is highlighted in (2.11).

**Important Equation(s) 3 (Lag)**

We can measure the lag of a filtered time series \( \tilde{y} \) to its underlying series \( y \) with:

\[
\mathcal{L}(y, \tilde{y}) = \arg \min_{l=0,1,\ldots} \left\{ \frac{1}{T-l} \sum_{t=l+1}^{T} (\tilde{y}_t - y_{t-l})^2 \right\}
\]

(2.11)

### 2.3.2.1 Expected Lag

Lag has been defined as minimizing the MSE. Using the data model (2.1), the lag of a filter \( w \) can be estimated without analysing the data. The following will assume that \( w \) sums to 1 as it has been in the literature so far.

Measuring the expected error at \( l \) lag gives:

\[
E \left[ \frac{1}{N} ||y_t - \bar{Y}w||^2 \right] = E[||y_{t-l} - \tilde{y}_t w||^2].
\]

(2.12)

It is possible to write:

\[
\tilde{y}_t = y_{t-l} \mathbf{1}^T + [\ldots, -\epsilon_{t-1} - \epsilon_{t-2}, -\epsilon_{t-1}, 0, \epsilon_{t+1}, \epsilon_{t+1} + \epsilon_{t+2}, \ldots].
\]
Then (2.12) becomes:

$$E[||y_{t-l} - \bar{y}_t w||^2] = E[||y_{t-l} - (y_{t-l}^T + \epsilon^T H_l)w||^2],$$

where

$$\epsilon = [\cdots, \epsilon_{t-1}, \epsilon_t]^T$$

and

$$H_l = \begin{bmatrix}
\ddots & \vdots & \vdots & \vdots & \vdots \\
\vdots & -1 & 0 & 0 & 0 \\
\vdots & -1 & -1 & 0 & 0 \\
\vdots & 0 & 0 & 0 & 0 \\
\vdots & 0 & 0 & 0 & 1 \\
\vdots & 0 & 0 & 0 & 0 \\
\vdots & \ddots & \vdots & \vdots & \vdots 
\end{bmatrix}$$

where the zero column is the lth column. Then, since $1^T w = 1$:

$$E[||y_{t-l} - \bar{y}_t w||^2] = E[||\epsilon^T H_l w||^2]$$

$$= E[w^T H_l^T \epsilon \epsilon^T H_l w]$$

$$= w^T H_l^T (\sigma^2 I + \mu^2 J) H_l w.$$  

Since $\mu \approx 0$, then:

$$E[||y_{t-l} - \bar{y}_t w||^2] = \sigma^2 ||H_l w||^2. \quad (2.13)$$

Finally, the lag of the filter $w$ is the lag found by minimising (2.13):

$$E[\mathcal{L}_w] = \arg\min_{l=0 \ldots n} \{ \sigma^2 ||H_l w||^2 \} = \arg\min_{l=0 \ldots n} \{ ||H_l w||^2 \} \quad (2.14)$$

The expected lag is highlighted in (2.15).

Despite the fact that (2.15) is correct, we can repeat the experiment in Figure 2.3 and visually see that it is correct. If we measure the lag (2.11) and expected lag (2.15) instead of the (expected) smoothness index we ought to see a strong correlation between the two. Figure 2.4 shows the results of the experiment and we can see a strong correlation.
Important Equation(s) 4 (Expected Lag)

We can measure the expected Lag of a time series $y$ that follows the model (2.1), and has been filtered with a linear filter $w$ with:

$$E[\mathcal{L}(y, \hat{y}_w)] = E[\mathcal{L}_w] = \arg\min_{l=0,...,n} \{ ||H_w||^2 \}$$  \hspace{1cm} (2.15)

Figure 2.4: Correlation between lag and its expected value. Filters were randomly generated and applied to both random time series and the real world time series in Section 2.2.

2.3.3 Smoothness vs Lag

The final aim of any filter is to maximise smoothing while minimizing lag. On the surface, this problem looks like a multi-objective optimization problem. But in fact, focusing on maximising the smoothness will achieve both goals. For example, suppose that filter A has a SI of 0.9 at lag 10 and 0.7 at lag 5. Filter B has a SI of 0.9 at lag 5. Filter B has effectively reduced the lag at SI 0.9 from 10 to 5. Thus, any optimization performed need only maximise the smoothing at each lag.

Figure 2.5 shows the relationship between lag and smoothness for three basic filters. The expected SI is the highest $E[\text{SI}]$ for each lag and filter. As we expected, there is an increase of smoothness as lag increases.
2.3.4 Closeness of Fit

It seems obvious to say that the output of a filter ought to be close to the original prices. Many models in the academic literature do indeed include a parameter to control the fitness. However, the most common filters (SMA, WMA, HMA), and one of the filters soon to be developed, do not incorporate a definition of fit. Thus, this section will discuss two definitions to ensure or control fitness, one without a parameter and one with a parameter.

2.3.4.1 Boundary Condition

The first method to ensure fitness, is to ensure the filter conforms to a boundary condition. Quite simply, the filter output should not be higher than the maximum value in the window and also should not be lower than the lowest value:

**Definition 1.** Take a filter \( w \) and apply it to a time series \( y \) in the fashion of \( \hat{y} = \hat{Y}w \). Then \( \hat{y} \) is said to be “within the boundary” of \( y \) if, for each row of \( \hat{Y} (\hat{y}_t) \), \( \hat{y}_t \) is \( \leq \max \{ \hat{y}_t \} \) and \( \geq \min \{ \hat{y}_t \} \).

This boundary condition can be met by ensuring that

\[
\min \{w\} \geq 0, \quad (2.16)
\]
\[
\sum w = 1. \quad (2.17)
\]

Note that \( \sum w = 1 \) preserves a constant signal.

To prove that, if (2.16) & (2.17) are true, the boundary condition is also true, we first ensure that \( w \) sums to 1 by rewriting it as: \( w/1^T w \).
CHAPTER 2. SMOOTHING STOCK PRICES

Then, we can write:

\[
\begin{align*}
\vec{y}_t^T w &= \tilde{y}_t \\
\vec{y}_t^T \vec{1} w &= \tilde{y}_t \vec{1}^T w \\
\vec{y}_t^T w - \tilde{y}_t \vec{1}^T w &= 0 \\
(\vec{y}_t^T - \tilde{y}_t \vec{1}^T) w &= 0
\end{align*}
\]

So long as all \( w \) are greater than or equal to zero, one of two situations will arise:

1. Only one element in \( w \) will be greater than zero. In this case, \( \tilde{y}_t \) will equal the corresponding price, and definition 1 holds.

2. More than one element in \( w \) will be greater than zero. In this case, the elements in \( (\vec{y}_t^T - \tilde{y}_t \vec{1}^T) \) corresponding to the non-zero \( w \)s cannot be all positive or all negative. Since we know that all prices (that is elements in \( \vec{y}_t \)) will be positive, \( \tilde{y}_t \) must be less than at least one price and greater than at least one price. Thus, definition 1 holds.

2.3.4.2 Mean Squared Error

The second method of controlling fitness is to simply include an error term in the equation: \( ||y - \hat{Y} w||^2 \).

2.3.4.3 Measuring Closeness of Fit

Measuring the closeness of fit is the same regardless of which method to control the fit is being used. Simply measure the Mean Squared Error (MSE) between the filter output and the data at the respective lag. The means to calculate the error and the expected error were developed in the discussion on lag in section 2.3.2. The expected error is a simple modification of equation (2.14):

\[
\min_{l=0...n} \{ \sigma^2 ||H_l w||^2 \}.
\]

Filters can be easily compared independently of scale and data by
simply dropping the $\sigma^2$ term, so the following will be used from now on:

$$\text{Fit}(y, \tilde{y}) = \min_{l=0...n} \left\{ \frac{||y_{t-l} - \tilde{y}||^2}{\sigma^2 N} \right\}$$

$$E[\text{Fit}(y, \tilde{Y} w)] = \min_{l=0...n} \left\{ \|H_l w\|^2 \right\}$$

called the expected fit. These measures are highlighted in (2.18) and (2.19) respectfully.

**Important Equation(s) 5 (Measuring (Expected) Fit)**

We can measure the Fit of a time series $\tilde{y}$ to the underlying time series $y$ with:

$$\text{Fit}(y, \tilde{y}) = \min_{l=0...n} \left\{ \frac{||y_{t-l} - \tilde{y}||^2}{\sigma^2 N} \right\} \quad (2.18)$$

We can measure the expected Fit of a time series $y$ that follows the data model (2.1), and has been filtered with a linear filter $w$ with:

$$E[\text{Fit}(w)] = E[\text{Fit}(y, \tilde{Y} w)] = \min_{l=0...n} \left\{ \|H_l w\|^2 \right\} \quad (2.19)$$

Even though we know that (2.19) is correct, as with the smoothness index (SI) and lag, we can run a quick experiment to visually see that it is indeed correct.

The experiment goes as follows. Generate 14 time series with changes drawn from a standard normal distribution. For each of the random series: (a) generate a random vector of a random length between 2 & 50 from a standard normal distribution, take the cumulative sum, shift the result so that the minimum value is equal to zero and then normalise the vector to sum to one. This is our filter weights. (b) Measure the fit and expected fit. Do this 100 times. For each of the 14 logged asset prices, do the same thing. The results are plotted in figure 2.6.

This experiment shows some interesting, but normal, behaviour. Look at Figure 2.6 at the randomly generated data. Each dot is a completely different random filter, and each colour is a time series. We can see that even though the time series forms straight lines (high correlation) there
is some heteroscedasticity depending on the underlying time series and the fit. We can see more pronounced variability in the real world data. Still, the correlation is high which is all we need.

![Randomly Generated Data](image1)

![Real Data](image2)

**Figure 2.6:** Correlation of fit and its expected value. Each dot represents a random linear filter and each colour is a time series.

As expected, at least on the simple filters, the error increases as both lag and smoothness increase. See Figure 2.7. The expected fit was calculated for each set of filter parameters that maximised the $E[SI]$ for each $E[Lag]$.

![Error vs Lag](image3)

![Error vs Smoothness](image4)

**Figure 2.7:** Filter expected fit vs expected lag and smoothness index. The expected fit was calculated for each filter that maximised the $E[SI]$.

## 2.4 Filter Derivation

The last section (2.3) developed the concepts and measurements of lag, smoothness and fitness. Now, a new family of filters are developed in this section. The first step is to derive a general data-drive filter. This filter uses the time series to find the weight vector $w$ that maximises
smoothness. Then, using the data model (2.1), the expected filter is derived. That is, a filter is derived where the time series is not needed to calculate the weight vector.

2.4.1 Data Driven Derivation

Here, the minimization problem is clear: find the weights that produce the smoothest curve subject to any constraints and a measure of fit. For the moment, the boundary condition for closeness of fit will be ignored and focus will be placed on the MSE method of controlling the fit.

There is only one constraint placed upon the linear filters. The filter weights in the simple models sum to one. An explanation was only found in [126] which noted that this constraint ensured that the filter preserved a constant signal.

Finding the filter weights then becomes a minimisation problem:

\[
\mathbf{w} = \arg \min_{\mathbf{w}} \left\{ ||\mathbf{D}_d \mathbf{Y} \mathbf{w}||^2 + \lambda ||\mathbf{y} - \mathbf{Y} \mathbf{w}||^2 \right\}
\]

s.t. \( \mathbf{1}_n^T \mathbf{w} = 1 \),

where \( \mathbf{D}_d \) is the differencing matrix (see page 52), \( \mathbf{w} \) is a vector of the filter weights, \( \mathbf{Y} \) is an \( N \times n \) matrix of the time series such that \( \mathbf{Y} \mathbf{w} \) is the filtered time series, \( \mathbf{y} \) is the middle column of \( \mathbf{Y} \), and \( \mathbf{1}_n \) is a vector of 1s the same length as \( \mathbf{w} \) (\( n \)). The size \( n \) relates to lag by: \( 2l + 1 \), where \( l \) is the lag.

Here, I rewrite the minimisation problem. By rescaling \( \lambda \) we can more easily control the trade-off between smoothness and fitness. Set the range of \( \lambda \) to \( [0 - 1] \), then the minimisation problem can be written as:

\[
\mathbf{w} = \arg \min_{\mathbf{w}} \left\{ \lambda ||\mathbf{D}_d \mathbf{Y} \mathbf{w}||^2 + (1 - \lambda) ||\mathbf{y} - \mathbf{Y} \mathbf{w}||^2 \right\}
\]

s.t. \( \mathbf{1}_n^T \mathbf{w} = 1 \) \hspace{1cm} (2.20)

Thus, when \( \lambda = 1 \) only the smoothing term is having an effect and when \( \lambda = 0 \) only the error term is having an effect.

Using Lagrange multipliers to solve the minimisation problem with
respect to \( w \):

\[
\delta \left[ \lambda \|D_d \tilde{y} w\|^2 + (1 - \lambda) \|y - \tilde{y} w\|^2 \right] = \gamma[\delta(1_n^T w)]
\]

\[
2\lambda(D_d \tilde{y})^T D_d \tilde{y} w - 2(1 - \lambda) \tilde{y}^T (y - \tilde{y} w) = \gamma 1_n
\]

We can drop the 2s here as \( \gamma \) “absorbs” them \((\gamma = \gamma/2)\):

\[
\gamma 1_n = \lambda(D_d \tilde{y})^T D_d \tilde{y} w - (1 - \lambda) \tilde{y}^T y + (1 - \lambda) \tilde{y}^T \tilde{y} w.
\]

Then expand:

\[
\gamma 1_n = \lambda(D_d \tilde{y})^T D_d \tilde{y} w - (1 - \lambda) \tilde{y}^T y + (1 - \lambda) \tilde{y}^T \tilde{y} w.
\]

Solve for \( w \):

\[
\begin{bmatrix}
\lambda(D_d \tilde{y})^T D_d \tilde{y} + (1 - \lambda) \tilde{y}^T \tilde{y}
\end{bmatrix} w = (1 - \lambda) \tilde{y}^T y + \gamma 1_n
\]

\[
F = \lambda(D_d \tilde{y})^T D_d \tilde{y} + (1 - \lambda) \tilde{y}^T \tilde{y}
\]

\[
w = (1 - \lambda) F^{-1} \tilde{y}^T y + \gamma F^{-1} 1_n
\]

Now to solve for \( \gamma \) which controls the constraint \( 1_n^T w = 1 \). Putting (2.21) into (2.20):

\[
1_n^T [(1 - \lambda) F^{-1} \tilde{y}^T y + \gamma F^{-1} 1_n] = 1
\]

\[
(1 - \lambda) 1_n^T F^{-1} \tilde{y}^T y + \gamma 1_n^T F^{-1} 1_n = 1
\]

\[
\gamma 1_n^T F^{-1} 1_n = 1 - (1 - \lambda) 1_n^T F^{-1} \tilde{y}^T y
\]

\[
\gamma = \frac{1 - (1 - \lambda) 1_n^T F^{-1} \tilde{y}^T y}{1_n^T F^{-1} 1_n}
\]

Putting this back into (2.21) we have the final data-driven filter:

\[
w = \gamma F^{-1} 1_n + (1 - \lambda) F^{-1} \tilde{y}^T y
\]

\[
w = \frac{1 - (1 - \lambda) 1_n^T F^{-1} \tilde{y}^T y}{1_n^T F^{-1} 1_n} F^{-1} 1_n + (1 - \lambda) F^{-1} \tilde{y}^T y
\]

This filter is highlighted in (2.24).
Important Equation(s) 6 (Data Driven Filter)

The filter weights $w$ that smooth a time series $y$ are:

$$
w = \frac{1 - (1 - \lambda)1^T_n F^{-1} \hat{Y}^T y}{1^T_n F^{-1} 1_n} F^{-1} 1_n + (1 - \lambda)F^{-1} \hat{Y}^T y, \quad (2.24)$$

$$F = \lambda (D_d \hat{Y})^T D_d \hat{Y} + (1 - \lambda)\hat{Y}^T \hat{Y},$$

where $n$ is the filter size and is calculated by $2l + 1$ where $l$ is the lag, $d$ is the level of differencing and $\lambda$ controls smoothing. When $\lambda = 1$, the filter focuses on smoothing and forgets fitness; when $\lambda = 0$ the filter focuses on fitness and forgets smoothing.

2.4.2 Expected Filter Derivation

The minimisation problem defined in the last section is:

$$w = \arg \min_w \left\{ \lambda \|D_d \hat{Y} w\|^2 + (1 - \lambda)\|y - \hat{Y} w\|^2 \right\}$$

$$1^T_n w = 1$$

Taking the expected value simply gives us:

$$E[w] = \arg \min_{E[w]} \left\{ \lambda E\left[\|D_d \hat{Y} E[w]\|^2\right] + (1 - \lambda)E\left[\|y - \hat{Y} E[w]\|^2\right]\right\}$$

$$1^T_n E[w] = 1 \quad \text{(2.25)}$$

Solving for $E[w]$ in the same way as solving for $w$ in the last section gives us:

$$E[w] = \frac{1 - (1 - \lambda)1^T_n E[F]^{-1} E[\hat{Y}^T y]}{1^T_n E[F]^{-1} 1_n} E[F]^{-1} 1_n$$

$$+ (1 - \lambda)E[F]^{-1} E[\hat{Y}^T y]$$

$$E[F] = \lambda E[(D_d \hat{Y})^T D_d \hat{Y}] + (1 - \lambda)E[\hat{Y}^T \hat{Y}]$$

The answer then depends on the expected value of two matrices: the covariance matrix $(D_d \hat{Y})^T D_d \hat{Y}$ and Gramian matrix $\hat{Y}^T \hat{Y}$. These will
be derived next then the two unwanted parameters, $\sigma$ and $N$ (the time series length) will be removed.

**Expected Value of the Covariance Matrix**

This section finds the expected covariance matrix: $(D_d \vec{Y})^T D_d \vec{Y}$ given the data model (2.1). To illustrate, the derivation to $d = 1$ & $d = 2$ will be provided.

First note that:

$$E[\epsilon^2] = \mu^2 + \sigma^2$$

$$E[\epsilon_i \epsilon_j] = \mu^2 \quad (i \neq j)$$

Now let’s examine when $d = 1$ starting with the matrix $D_1 \vec{Y}$:

$$D_1 \vec{Y} = \begin{bmatrix}
\epsilon_2 & \epsilon_3 & \ldots & \epsilon_{n+1} \\
\epsilon_3 & \epsilon_4 & \ldots & \epsilon_{n+2} \\
\vdots & \vdots & \ddots & \vdots \\
\epsilon_{N-n+1} & \epsilon_{N-n+2} & \ldots & \epsilon_N
\end{bmatrix}$$

Then $E[(D_1 \vec{Y})^T D_1 \vec{Y}]$ has $(N-n)E[\epsilon^2]$ along the diagonal and $(N-n)E[\epsilon_i \epsilon_j]$ everywhere else. This reduces to:

$$E[(D_1 \vec{Y})^T D_1 \vec{Y}] = (N-n)(\sigma^2 I + \mu^2 J)$$

Now to examine $d = 2$:

$$D_2 \vec{Y} = \begin{bmatrix}
\epsilon_3 - \epsilon_2 & \epsilon_4 - \epsilon_3 & \ldots \\
\epsilon_4 - \epsilon_3 & \epsilon_5 - \epsilon_4 & \ldots \\
\vdots & \vdots & \ddots
\end{bmatrix}$$

Then $E[(D_2 \vec{Y})^T D_2 \vec{Y}]$ has $(N-n-1)E[(\epsilon_{x_1} - \epsilon_{x_2})^2]$ along the diagonal, $(N-n-1)E[(\epsilon_{x_1} - \epsilon_{x_2})(\epsilon_{x_3} - \epsilon_{x_1})]$ on the off diagonal and
\((N - n - 1)E[(\epsilon_{x_1} - \epsilon_{x_2})(\epsilon_{x_3} - \phi_{x_4})]\) everywhere else. This reduces to:

\[
E[(D_2 \vec{Y})^T D_2 \vec{Y}] = (N - n - 1)\sigma^2 \begin{bmatrix}
2 & -1 & 0 & \ldots & 0 \\
-1 & 2 & -1 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & -1 & 2
\end{bmatrix}
\]

Define \(M_2\) such that:

\[
E[(D_2 \vec{Y})^T D_2 \vec{Y}] = (N - n - 1)\sigma^2 M_2
\]

When \(d = 1\) \(M_1 = I\):

\[
E[(D_1 \vec{Y})^T D_1 \vec{Y}] = (N - n)(\sigma^2 M_1 + \mu^2 J)
\]

The derivation of \(M_d\) follows a similar pattern for all other \(d\). Thus:

\[
E[(D_d \vec{Y})^T D_d \vec{Y}] = \begin{cases} (N - n)(\sigma^2 M_1 + \mu^2 J) & : d = 1 \\ (N - n - d + 1)\sigma^2 M_d & : d > 1 \end{cases}
\]

Under the assumption that \(\mu \approx 0\) it follows that:

\[
E[(D_d \vec{Y})^T D_d \vec{Y}] = (N - n - d + 1)\sigma^2 M_d \quad (2.27)
\]

The matrix \(M_d\) can be more efficiently computed by observing that the elements actually follow a binomial pattern:

\[
M_{d,i,j} = \binom{2d - 2}{i - j + d - 1} (-1)^{i-j} \quad (2.28)
\]

where \(\binom{z}{k}\) is a binomial coefficient and \(\binom{z}{k} = 0\) for \(k < 0\) and \(k > z\). It can also be written as: \(M_{n,d} = D_{n+d-1,d-1} D_{n+d-1,d-1}^T\).

Expected Value of the Gramian Matrix

This section examines the structure of \(E[\vec{Y}^T \vec{Y}]\). Note that \(E[\vec{Y}^T Y]\) is simply a column of \(E[\vec{Y}^T \vec{Y}]\).
As the data model (2.1) is recursive, we could alternatively write:

\[ y_t = s + \sum_{i=1}^{t} \epsilon_i, \]

where \( s \) is some unknown start value. Then \( \vec{Y} \) is a \( k \times n \) matrix and takes on the following form:

\[
\vec{Y} = s\mathbf{1}_{N-n+1}^T + \mathbf{A}
\]

\[
\mathbf{A} = \begin{bmatrix}
\epsilon_1 & \epsilon_1 + \epsilon_2 & \ldots & \sum_{i=1}^{n} \epsilon_i \\
\epsilon_1 + \epsilon_2 & \ldots & \ldots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\sum_{i=1}^{N-n+1} \epsilon_i & \ldots & \ldots & \sum_{i=1}^{N} \epsilon_i
\end{bmatrix}
\]

\( k = N - n + 1, \)

and \( y \) can be written as:

\[
y = s\mathbf{1}_{N-n+1} + b \\
b = \text{middle column of } \mathbf{A}.
\]

The variable \( s \) can actually be dropped here by rewriting the fitness objective function in the minimisation problem:

\[
||y - \vec{Y}w||^2 = ||s\mathbf{1}_{N-n+1} + b - (s\mathbf{1}_{N-n+1}\mathbf{1}_n^T + \mathbf{A})w||^2 \\
= ||s\mathbf{1}_{N-n+1} + b - s\mathbf{1}_{N-n+1}\mathbf{1}_n^Tw - Aw||^2
\]

We know that \( \mathbf{1}_n^Tw = 1 \) so:

\[
||y - \vec{Y}w||^2 = ||b - Aw||^2
\]

Thus, from now on, \( \vec{Y} \) and \( y \) will be modelled as if \( s = 0 \):

\[
\vec{Y} = \begin{bmatrix}
\epsilon_1 & \epsilon_1 + \epsilon_2 & \ldots & \sum_{i=1}^{n} \epsilon_i \\
\epsilon_1 + \epsilon_2 & \ldots & \ldots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\sum_{i=1}^{N-n+1} \epsilon_i & \ldots & \ldots & \sum_{i=1}^{N} \epsilon_i
\end{bmatrix}
\]

\( b = \text{middle column of } \vec{Y}. \)
To look at each element of $E[\hat{Y}^T \hat{Y}]$ first denote $r$ as a column of $\hat{Y}$ (row of $\hat{Y}^T$), the index of the row of $E[\hat{Y}^T \hat{Y}]$ as $r$ and the column as $c$. Start the indexation at 0. The matrix is symmetric and it simplifies matters for now to say that $r \leq c$. Each element of $E[\hat{Y}^T \hat{Y}]$ can be written as $E[r_r^T r_c]$: \[ E[r_r^T r_c] = E[\sum_{i=1}^{k}(\sum_{j=1}^{i+r}(\sum_{j=1}^{i+c} \epsilon_j)(\sum_{j=1}^{i+c} \epsilon_j))] = \sum_{i=1}^{k} E[(\sum_{j=1}^{i+r} \epsilon_j)(\sum_{j=1}^{i+c} \epsilon_j)] \quad (2.29) \]

The expected value of the product is:
\[ E[(\sum_{j=1}^{i+r} \epsilon_j)(\sum_{j=1}^{i+c} \epsilon_j)] = (i + r)(\mu^2 + \sigma^2) + (i + r)(i + c - 1)\mu^2 \]
\[ = (i + r)\sigma^2 + (i + r)(i + c)\mu^2 \]

Substituting this into (2.29) we see:
\[ E[r_r^T r_c] = \sum_{i=1}^{k} [(i + r)\sigma^2 + (i + r)(i + c)\mu^2] \]
\[ = \sigma^2 \left( \sum_{i=1}^{k} i + kr \right) + \mu^2 \left( \sum_{i=1}^{k} i^2 + ic + ri + rc \right) \]
\[ = \sigma^2 \left( \sum_{i=1}^{k} i + kr \right) + \mu^2 \left( \sum_{i=1}^{k} i^2 + (c + r) \sum_{i=1}^{k} i + krc \right) \]

The sum of successive numbers from 1 to $k$ and the sum of successive squares can be written as:
\[ T_k = \sum_{i=1}^{k} i = \frac{k(k + 1)}{2}, \quad S_k = \sum_{i=1}^{k} i^2 = \frac{k(k + 1)(2k + 1)}{6} \]

Then:
\[ E[r_r^T r_c] = \sigma^2 \left( \sum_{i=1}^{k} i + kr \right) + \mu^2 \left( \sum_{i=1}^{k} i^2 + (c + r) \sum_{i=1}^{k} i + krc \right) \]
\[ E[r_r^T r_c] = \sigma^2 (T_k + kr) + \mu^2 (S_k + (c + r)T_k + krc) = E[\hat{Y}^T \hat{Y}]_{r,c} \]

With the assumption that $\mu \approx 0$:
\[ E[\hat{Y}^T \hat{Y}]_{r,c} = E[r_r^T r_c] = \sigma^2 (T_k + kr) \]
Because this matrix is symmetric we can drop the $r \leq c$ assumed above and simply put:

$$E[\overrightarrow{Y}^T \overrightarrow{Y}]_{r,c} = \sigma^2(k + k \cdot \min\{r, c\})$$

$$= \sigma^2 k \left(\frac{k+1}{2} + \min\{r, c\}\right)$$

We can drop the specification of the rows and columns by noting that:

$$[U^T U]_{r,c} = \min\{r, c\}$$

where $U$ is an upper triangle matrix of 1s. Setting $u_l$ as the $l^{th}$ column of $U$ we can write the expected value of the Gramian and its $l^{th}$ column as:

$$E[\overrightarrow{Y}^T \overrightarrow{Y}] = \sigma^2 k \left(\frac{k+1}{2} J + U^T U\right)$$

(2.30)

$$E[\overrightarrow{Y}^T y_l] = \sigma^2 k \left(\frac{k+1}{2} J + U^T u_l\right)$$

(2.31)

**Final Filter Model**

Now that we know the expected values of the Gramian & Covariance matrices we can find the expected value of $w$. But first, we are going to use a little trick to remove the $k$ variable that is present in (2.30) from the expected value of $w$. Notice in (2.25) the error term is:

$$E \left[||y - \overrightarrow{Y} E[w]||^2\right]$$

Minimising this is the solution to $E[\overrightarrow{Y}^T \overrightarrow{Y}] E[w] = E[\overrightarrow{Y}^T y]$ which is also the solution to the minimisation problem: $||E[\overrightarrow{Y}^T y] - E[\overrightarrow{Y}^T \overrightarrow{Y}] E[w]||^2$. Thus the two minimisations are equivalent and we can use the second one instead.

Using (2.30) & (2.31) We can write:

$$\mathbf{Y} = ||E[\overrightarrow{Y}^T y] - E[\overrightarrow{Y}^T \overrightarrow{Y}] E[w]||^2 = ||\sigma^2 k \left(\frac{k+1}{2} + U^T u_{l+1}\right)$$

$$- \sigma^2 k \left(\frac{k+1}{2} + U^T U\right) E[w]||^2$$
Expanding:
\[ \Upsilon = \| \sigma^2 k \frac{k+1}{2} 1_n + \sigma^2 k U^T u_{t+1} - \sigma^2 k \frac{k+1}{2} 1_n 1_n^T E[w] - \sigma^2 k U^T U E[w] \| ^2 \]

We know that \( 1_n^T E[w] = 1 \) which removes the sum of consecutive numbers:
\[ \Upsilon = \| \sigma^2 k U^T u_{t+1} - \sigma^2 k U^T U E[w] \| ^2 \]

This is the minimisation problem to:
\[ \sigma^2 k U^T U E[w] = \sigma^2 k U^T u_{t+1} \]

The term \( \sigma^2 k \) does not affect the outcome. We could drop the entire term, but the \( \sigma^2 \) will be useful later on. So we will only drop the \( k \) parameter:
\[ \sigma^2 U^T u = \sigma^2 U^T u_{t+1} \]

The solution of which is the minimization of \( \| \sigma u_{t+1} - \sigma U E[w] \| ^2 \). Thus:
\[ \arg \min_{E[w]} \| E[\tilde{Y}^T y] - E[\tilde{Y}^T \tilde{Y}] E[w] \| ^2 = \arg \min_{E[w]} \| \sigma u_{t+1} - \sigma U E[w] \| ^2 \]

Putting this into (2.25) gives us:
\[ E[w] = \arg \min_{E[w]} \left\{ E \left[ \| D_d \tilde{Y}^T E[w] \| ^2 \right] + \lambda \| \sigma u_{t+1} - \sigma U E[w] \| ^2 \right\} \]
\[ 1_n^T E[w] = 1 \]

Solving for \( E[w] \) gives us:
\[ E[w] = \frac{1 - (1 - \lambda) 1_n^T E[F]^{-1} \sigma^2 U^T u_{t+1}}{1_n^T E[F]^{-1} 1_n} E[F]^{-1} 1_n \]
\[ + (1 - \lambda) E[F]^{-1} \sigma^2 U^T u_{t+1} \]

\[ E[F] = \lambda E[(D_d \tilde{Y})^T D_d \tilde{Y}] + (1 - \lambda) \sigma^2 U^T U \]

Inserting (2.27):
\[ E[w] = \frac{1 - (1 - \lambda) 1_n^T E[F]^{-1} \sigma^2 U^T u_{t+1}}{1_n^T E[F]^{-1} 1_n} E[F]^{-1} 1_n \]
+ (1 − λ)E[F]^{-1}σ^2U^Tu_{l+1}

E[F] = λ(N − n − d + 1)σ^2M_d + (1 − λ)σ^2U^TU

The variance can be quickly removed from the filter by observing that \( σ^2 \) is cancelled out due to division:

\[
E[w] = \frac{1 - (1 - λ)1_n^TE[F]^{-1}U^Tu_{l+1}E[F]^{-1}1_n}{1_n^TE[F]^{-1}1_n} + (1 - λ)E[F]^{-1}U^Tu_{l+1}
\]

\[
E[F] = λ(N − n − d + 1)M_d + (1 − λ)U^TU.
\]

Finally, the \((N − n + 1 − d)\) term beside the \(M_d\) matrix does nothing more than rescale the \(λ\) term. Thus, this can also be dropped. We have arrived at the final filter which is highlighted in (2.32).

**Important Equation(s) 7 (Expected Filter)**

The expected filter of (2.24) given the data model (2.1) is:

\[
E[w] = \frac{1 - (1 - λ)1_n^TE[F]^{-1}U^Tu_{l+1}E[F]^{-1}1_n}{1_n^TE[F]^{-1}1_n} + (1 - λ)E[F]^{-1}U^Tu_{l+1}, \tag{2.32}
\]

\[
E[F] = λM_d + (1 - λ)U^TU.
\]

The filter’s parameters are:

- \(d\) Controls the level of differencing that is to be minimised.
- \(l\) Specifies the lag.
- \(λ\) A value between 0 and 1 that controls the amount of smoothing. Zero means no smoothing while one means maximum smoothing.

The terms in the equations are:

- \(n\) The filter size is \(n = 2l + 1\).
- \(M_d\) Is an \(n \times n\) matrix described in (2.27).
- \(U\) An \(n \times n\) upper triangle matrix of ones.
- \(u_{l+1}\) The \((l + 1)\)th column of \(U\).
- \(1_n\) A \(n \times 1\) vector of 1s.

Even though we know that (2.32) is correct because of the rigorous
CHAPTER 2. SMOOTHING STOCK PRICES

derivation, we can run a quick experiment to visually see that it is indeed accurate. In this experiment I iterated lag across 10 rounded evenly spaced values between and including 1 and 50. I iterated $d$ from 1 to 5 and $\lambda$ at 5 evenly spaced intervals between and including 0.1 and 1. Letting $\lambda$ equal zero has been omitted as the filter weights are just zero with one in the centre which may skew results. For each combination of the parameters I:

1. Calculated the filter from a random series of length 2,000 with changes drawn from a standard normal distribution.
2. Calculated the filter from a randomly chosen logged asset price series.
3. Calculated the expected filter.

If (2.32) is correct then the expected filter weights ought to be highly correlated with the actual filter weights. Figure 2.8 shows what this data looks like when dumped to a graph. Each dot is a single filter weight and they are highly correlated.

Figure 2.8: Correlation of the filter weights and their expected value. Each dot is a filter’s weight. The real world data are the time series in Section 2.2.

Figure 2.9 shows two specific filters. Here, I calculated the filters with lag = 20, $d = 5$, and $\lambda = [0.5, 1]$ on 50 random time series with changes drawn from a standard normal distribution. They are graphed in blue with the expected filter plotted in red in the foreground. Again, the fit is accurate.
2.5 Filter Family

The filter equations derived above can be used for three different filters based on parameters. These filters are named as Linear Gaussian Smoother (LGS), Symmetric Linear Gaussian Smoother (SLGS), and Asymmetric Linear Gaussian Smoother (ALGS).

2.5.1 Linear Gaussian Smoother (LGS)

This is a symmetric filter that maximises smoothing. It is a special case of (2.32) where $\lambda = 1$.

$$E[w] = \frac{M_d^{-1}1_n}{1_n^T M_d^{-1}1_n}$$

(2.33)

Trench’s algorithm [268] is used to invert $M_d$.

The two parameters of this filter are:

- $d$ Controls the level of differencing that is to be minimised.
- $l$ Specifies the lag; the filter size is $n = 2l + 1$.

It is interesting to note that this filter does not require that $\mu \approx 0$. This restriction was used to simplify equation (2.27) for when $d = 1$. Removing the restriction and putting it into equation (2.33) (remember...
\( \mathbf{M}_1 = \mathbf{I} \) gives:

\[
E[\mathbf{w}] = \frac{((N-n)(\sigma^2 \mathbf{I} + \mu^2 \mathbf{J}))^{-1} \mathbf{1}_n}{\mathbf{1}_n^T \frac{((N-n)(\sigma^2 \mathbf{I} + \mu^2 \mathbf{J}))^{-1} \mathbf{1}_n}{\mathbf{1}_n^T (\mathbf{M}_1^{-1})^{-1} \mathbf{1}_n} = \frac{(\sigma^2 \mathbf{I} + \mu^2 \mathbf{J})^{-1} \mathbf{1}_n}{\mathbf{1}_n^T (\mathbf{M}_1^{-1})^{-1} \mathbf{1}_n}
\]

It can easily be seen that a matrix of the form \((x \mathbf{I} + y \mathbf{J})^{-1}\) can be expressed as \(a \mathbf{I} + b \mathbf{J}\) for some \(a\) and \(b\), thus:

\[
E[\mathbf{w}] = \frac{(a \mathbf{I} + b \mathbf{J}) \mathbf{1}_n}{\mathbf{1}_n^T (a \mathbf{I} + b \mathbf{J}) \mathbf{1}_n} = \frac{a \mathbf{1}_n + bn \mathbf{1}_n}{\mathbf{1}_n^T (a \mathbf{1}_n + bn \mathbf{1}_n)} = \frac{(a + bn) \mathbf{1}_n}{\mathbf{1}_n^T (a + bn) \mathbf{1}_n} = \frac{a \mathbf{1}_n + bn \mathbf{1}_n}{\mathbf{1}_n^T \mathbf{M}_1^{-1} \mathbf{1}_n}
\]

While the user of the filter can specify the use of \(d\), this can be handled by an algorithm if the goal is to reach maximal smoothing. The algorithm simply tries successive values of \(d\) until one of the following conditions are met:

1. The new filter \((w)\) is not as smooth as the last tested filter \((w)\).
2. The filter \((w)\) is not symmetric.
3. There are values less than zero in the filter \((w)\).

These conditions exist because, as \(d\) increases, the matrix \(\mathbf{M}_d\) becomes more and more unstable and causes ill effects when inverted. Figure 2.10 shows the \(d\) parameter that maximises the expected smoothness index \((E[\text{SI}])\) for each lag.

**Figure 2.10:** Differencing parameter. A graph of the parameter \(d\) that maximises the expected smoothness index for each lag.
Figure 2.11

Figure 2.11a shows the effects that \(d\) and \(l\) (lag) have on the SI. The parameter \(d\) extends no further than the value that maximizes the SI as shown in Figure 2.10. The surface in Figure 2.11a clearly shows the increased smoothing ability when lag is increased and \(d\) is increased.

The expected Fit of the LGS is presented in Figure 2.11b. Remembering that calculation of \(E[\text{Fit}]\) adjusts for lag, notice that as lag increases so does the error. This is due to the weighted averaging effect. As the lag increases so does the filter size. More and more data is used which removes more fluctuations. This increases the error while improving the smoothing. However, note that \(d\) also increases smoothing yet decreases the error. Recall from Figure 2.12 that \(d\) is tightening the filter weights around the centre. Thus, the central value, the location of the lag, receives more attention from the filter. The output has more fluctuations yet the fluctuations are smooth.

To see the effect of \(d\) on the filter weights, consult Figure 2.12 which shows the filter with lag = 50 and the corresponding weights for \(d = [1 - 6]\). As \(d\) increase the weights resemble more and more a bell shaped curve (similar to the Gaussian curve) that tightens around the centre.

Even though Figure 2.12 suggests that the filter is symmetric, the following corollary states this formally:

**Corollary 3.** The filter \(w\) given in (2.33) is a symmetric filter.

**Proof.** The scalar division in (2.33) can be ignored; it is sufficient to show that \(M_d^{-1}\mathbf{1}_n\) is symmetric.

We know that \(M_d\) is a symmetric Toeplitz matrix. It is well known
that a symmetric Toeplitz matrix is persymmetric and that the inverse of a persymmetric matrix is also persymmetric. That is:

\[
\mathbf{ETE} = \mathbf{T}
\]
\[
\mathbf{ET}^{-1}\mathbf{E} = \mathbf{T}^{-1},
\]

where \( \mathbf{T} \) is an arbitrary symmetric Toeplitz matrix and \( \mathbf{E} \) is a special permutation matrix called an exchange matrix, \( \mathbf{E} \) is all zeros except for 1 across the counterdiagonal. The exchange matrix reverses the rows of a matrix or vector multiplied on its right. Given this:

\[
\mathbf{EM}^{-1}\mathbf{E} = \mathbf{M}^{-1}
\]
\[
\mathbf{EM}^{-1}\mathbf{E}\mathbf{1}_n = \mathbf{M}^{-1}\mathbf{1}_n
\]
Because $E1_n = 1_n$, we have:

$$EM^{-1}1_n = M^{-1}1_n.$$ 

Thus, the LGS filter is symmetric.

Unfortunately, the LGS filter does not have a built in mechanism to ensure the output is close to the input. This is because $\lambda = 1$ which completely ignores the error term from (2.32). Recall the boundary condition defined on page 63. A filter $w$ must hold two constraints to be considered “close” to the input time series: $1_n^TW = 1$ and $w \geq 0$. The first constraint is already ensured, as it is built into the filter, but there is no proof for the second constraint.

Calculating $w$ for all lags from zero to 50, and for all $d$ from 1 to the $d$ which maximises the SI, then checking the minimum value of $w$ shows that it is above or equal to zero. Thus, for all relevant values of $w$, the second condition holds.

### 2.5.2 Symmetric Linear Gaussian Smoother (SLGS)

The Symmetric Linear Gaussian Smoother (SLGS) is exactly the same as the base filter equation (2.32).

Because the relationship between all the parameters and the SI is not linear, choosing a value for $\lambda$ can be difficult as the SI slowly changes. To see more appropriate changes in the SI the $\lambda$ term needs to be changed according to some non-linear law. Here, the replacement parameter $\gamma$ is introduced and its effect is:

$$\lambda = \frac{1}{1 + e^{-\gamma + 10}}$$

(2.34)

The value for $\gamma$ is taken to be on the range $[0 - 40]$. This simply allows us to vary $\gamma$ linearly, while $\lambda$ is varied non-linearly.

The effect of lag and $\gamma$ on the smoothness are shown in Figure 2.13a. The $d$ parameter was chosen by maximising the SI when $\lambda = 1$ as depicted in Figure 2.10. This surface is not entirely smooth as $\gamma/\lambda$ increases; there are short peaks. Still, the graph clearly shows SI increasing with $\gamma$ and
CHAPTER 2. SMOOTHING STOCK PRICES

Figure 2.13

(a) The Expected SI Surface of the SLGS. Points were removed where the $\log_{10}$ condition number of $F$ was above 14.

(b) The Expected SI Surface for the SLGS with Static Lag. The lag is set to 50.

Figure 2.14

(a) The Expected Fit Surface of the SLGS. Points were removed where the $\log_{10}$ condition number of $F$ was above 14.

(b) The Expected Fit Surface for the SLGS with Static Lag. The lag is set to 50.

Figure 2.13b depicts the expected SI against $d$ when lag = 50. The behaviour is similar to the LGS; as $d$ increases, so does the SI, albeit with a few fluctuations. As can be seen in the figure, $\gamma$ reaches a point where any further increase results in a negligible change in the SI. The tiny changes appear to be negative, that is, the SI is very slowly decreasing.

The error behaves as expected. Figure 2.14a shows the error increasing as both $\gamma$ and lag increase. Figure 2.14b displays the increase in fitness as $d$ grows larger.

As with the LGS, the assurance that the lag is correct comes from the property of filter symmetry. The following theorem demonstrates this for the SLGS.

**Theorem 2.** The filter $w$ given in equation (2.32) is a symmetric filter.
Proof. The minimization function is made up of two objectives:
\[
\mathbf{w} = \arg\min_{\mathbf{w}} \left\{ \lambda \| \mathbf{D}_d \mathbf{w} \|^2 + (1 - \lambda) \| \mathbf{u}_{l+1} - \mathbf{U} \mathbf{w} \|^2 \right\}, \quad \text{s.t. } \mathbf{1}_n^T \mathbf{w} = 1
\]
The first objective is to minimise smoothness while the second is to minimise error. If both of these objectives are independently symmetric then \( \mathbf{w} \) will also be symmetric. Theorem 3 (on page 80) gives the symmetry of the smoothness objective. Thus it remains to be shown that the error objective is symmetric.

The solution to \( \arg\min_{\mathbf{w}} \| \mathbf{u}_{l+1} - \mathbf{U} \mathbf{w} \|^2 \) s.t. \( \mathbf{1}_n^T \mathbf{w} = 1 \) is:
\[
\mathbf{w} = \frac{1 - \mathbf{1}_n^T \mathbf{U}^{-1} \mathbf{u}_{l+1}}{\mathbf{1}_n^T \mathbf{U}^{-1} \mathbf{U} \mathbf{1}_n} \mathbf{U}^{-1} \mathbf{u}_{l+1} + \mathbf{U}^{-1} \mathbf{u}_{l+1}
\]
The inverse of \( \mathbf{U}^T \mathbf{U} \) is:
\[
(\mathbf{U}^T \mathbf{U})^{-1} = \begin{bmatrix}
2 & -1 & 0 & 0 & \ldots \\
-1 & 2 & -1 & 0 & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \ldots & -1 & 2 & -1 \\
0 & \ldots & 0 & -1 & 1
\end{bmatrix}
\]
Then write:
\[
\mathbf{a} = (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{1}_n = [1, 0, \ldots, 0]^T
\]
Then the solution becomes:
\[
\mathbf{w} = \mathbf{a} - \mathbf{a}^T \mathbf{u}_{l+1} + \mathbf{U}^{-1} \mathbf{u}_{l+1} = \mathbf{U}^{-1} \mathbf{u}_{l+1} = [0, 0, \ldots, 1, \ldots, 0]
\]
since \( \mathbf{u}_{l+1} \) is the middle column of \( \mathbf{U} \). Recall that the filter size is \( n = 2l + 1 \) and the one in the equation above is in the centre position. Thus the solution to the error objective is symmetric.

Since both the error and smoothness objectives are symmetric, \( \mathbf{w} \) is also symmetric.

Figure 2.15 depicts the impulse response generated by the SLGS for lag = 20 and selected values of \( \lambda \) and \( d \). The plots demonstrate the convergence of \( \lambda \) at either extremes. The use of the error term now
CHAPTER 2. SMOOTHING STOCK PRICES

Figure 2.15: Examples of the SLGS weights. The lag is set to 20 at selected values of $\lambda$ and $d$.

produces an oscillating impulse response depending on the selection of $d$. 


2.5.3 Asymmetric Linear Gaussian Smoother (ALGS)

The ALGS is a slight modification of the SLGS into an asymmetric filter. Recall that the error term in equation (2.32) minimizes the error between the filter output $\overrightarrow{Y}w$ and the middle column ($\overrightarrow{y}$) of the input $\overrightarrow{Y}$. The modification changes this vector $\overrightarrow{y}$ to be the last column of $\overrightarrow{Y}$. This is simply done by changing the column $u_{l+1}$ (which is the centre column of $U$) to the last column $u_n$. This change results in the asymmetry and also in a change of meaning for the parameters $\lambda$ and the filter size which is no longer tied to lag:

- $\lambda$ Controls the error, smoothness, and lag. Can be set by $\gamma$ from (2.34).
- $d$ Controls the level of differencing that is to be minimised.
- $n$ Filter size, places an upper limit on smoothness and lag.

The filter equations are now:

$$E[w] = \frac{1 - (1 - \lambda)1_n^TE[F]^{-1}U^Tu_n}{1_n^TE[F]^{-1}1_n}E[F]^{-1}1_n + (1 - \lambda)E[F]^{-1}U^Tu_n$$

$$E[F] = \lambda M_d + (1 - \lambda)U^TU.$$

The SI (Figure 2.16) behaves similarly to the SLGS; increases with $\gamma$ and $d$.

![Figure 2.16: SI Surface of the ALGS. The filter size ($n$) is set to 201. While a square mesh of filters for parameters $d$ and $\gamma$ was calculated, values were removed where the log condition number of $F$, $\log_{10}(\kappa[F])$, was above 13.](image)

Lag is not a parameter in the ALGS, so we must rely on an estimate of the lag from $E[w]$ (2.15). The lag surface is pictured in Figure 2.17a against the $d$ and $\gamma$ parameters. Both lag (Figure 2.17a) and fit (Figure 2.17b) increase with the increase of $\gamma$ and with the decrease of $d$.

The effects of $\lambda$ and $d$ on the filter weights can be seen in Figure 2.18.
Figure 2.17: The lag and fitness surfaces of the ALGS. The filter size \((n)\) is set to 201. While a square mesh of filters for parameters \(d\) and \(\gamma\) was calculated, values were removed where the log condition number of \(F\), \(\log_{10}(\kappa(F))\), was above 13.

Figure 2.18: Examples of the ALGS weights.

2.6 Filter Comparison

While the ultimate purpose of filters for trading financial instruments is to produce as much profit as possible, the filters are seen as a price smoother first and a profit machine second. This section will compare the smoothing performance of the newly developed filters against the filters found in the literature.

The exact same parameters for each filter will be used across each of the datasets. The task of determining this set of parameters is discussed
in section 2.6.1. Then, by looking at the expected values for lag, smooth-
ness and error as well as the respective measurements on real data sets, 
the following questions will be addressed:

- Which filter has the most amount of smoothing for any given lag?
- Which filter has the least amount of lag for any given SI?
- Does the new family of filters sacrifice fitness?

The optimisation strategies for each of the filters now follow.

2.6.1 Filter Parameter Optimization

Before it is possible to determine the competitive standing of each filter, 
the parameters need to be assigned values. Because the main concern is 
to improve the SI per lag, this will be the basis for selection. A set of 
parameters for each of the filters is obtained by maximising the expected 
SI for each lag on the range \([1 - 50]\). This section details the optimisation 
for each of the filters.

Simple Moving Average

The Simple Moving Average (SMA) [215] appears to be the most popular filter among investors and 
researchers alike. This filter has only a single input, the filter size. As this filter is symmetric, 
the size \((n)\) is set to \(n = 2l + 1\). No optimisation is needed.

Weighted Moving Average

The Weighted Moving Average (WMA) [215] is an asymmetric filter with the filter size as the 
only parameter. To optimize, the filter size \(n\) was iterated over the range \([2 - 1000]\) and the 
smoothest size for lags \([1 - 50]\) was recorded.
CHAPTER 2. SMOOTHING STOCK PRICES

Hull Moving Average

The Hull Moving Average (HMA) [156] is an asymmetric filter with the filter size as the only parameter. To optimize, the filter size $n$ was iterated over the range $[2 - 1000]$ and the smoothest size for lags $[1 - 50]$ was recorded.

Arnaud Legoux Moving Average

The Arnaud Legoux Moving Average (ALMA) [196] is a more difficult filter to optimise, with three parameters; the filter size $n$, the offset $o$, and the bandwidth $\sigma$.

We can optimise this filter by iterating $n$ through the integers from 2 to 201; $o$ is iterated through 10 evenly spaced values between 0 and half the filter size. At each combination of $n$ and $o$, $\sigma$ was optimised over the range of 0 to 100. The optimisation algorithm finds the minimum value of the objective function on the specified range. It is built on the golden section search and parabolic interpolation described in [43, 108]. This algorithm is implemented as a standard algorithm in Matlab as fminbnd. The objective function is $-E[SI(w)]$.

Once $\sigma$ has been optimised for all combinations of $n$ and $o$, the smoothest parameters for each lag from 1 to 50 are kept.

Savitzy-Golay Filter

The Savitzi-Golay (SG) filter [247] has been included to provide a comparison against filters that are not used in the financial world but do indeed take a time series and smooth it. The filter is a popular filter originally developed in the field of chemistry. The filter is nothing more than a local polynomial model. A polynomial is fitted to a window of data and the center value is taken to be the smoothed value of that window.
Optimising this filter is very simple. It turns out that the polynomial order that produces the smoothest filter is order one. However, this then reduces the filter to the SMA filter. Thus, the order is set to two. Also, since the filter is symmetric, the filter size need only be $2 \times \text{lag} + 1$.

**Linear Gaussian Smoother**

This filter also has only two parameters to optimise, the filter size $n$ and the level of differencing $d$. Since the filter is symmetric, the filter size is set to $2l+1$. The $d$ value is optimised with the algorithm mentioned in section 2.5.1 (on page 78).

**Symmetric Linear Gaussian Smoother**

Despite the obvious situation where $\lambda$ controls smoothing and the mathematical equation dictates that $\lambda = 1$ ought to maximise smoothing, we know that this is not the case by looking at Figure 2.13a. This result is attributed to poor condition of the matrix used to compute the filter, namely the $F$ matrix. Observe the peaks that occur on the surface and notice that they extend higher (higher SI) than when $\gamma$ is at the maximum.

To optimise this filter for each lag, all combinations of the parameters $\gamma$ on the integer range [0-40] and $d$ on the range [1-20] will be tested. Filters whose $F$ matrix has a $\log_{10}$ condition number above 13 are discarded and not included in the optimisation. The parameters with the highest expected SI are selected as the filter parameters corresponding to that lag.

**Asymmetric Linear Gaussian Smoother**

This filter is very different from the SLGS. The $\lambda$ ($\gamma$) parameter now controls both lag and smoothing. The filter size simply enforces an upper limit
CHAPTER 2. SMOOTHING STOCK PRICES

on the lag and smoothing. So, from the pool of all combinations of 
\( n = [2 - 102], \gamma = [0 - 40], \) and \( d = [1 - 20], \) the smoothest parameters 
for each lag will be selected. Again, filters with a \( F \) matrix condition 
number whose \( \log_{10} \) value is above 13 are discarded.

2.6.2 Comparison Methodology & Results

This section discusses the method of comparing the filters and the results. 
The price series are the same as were used in section 2.2.

2.6.2.1 Comparing the Smoothness Index

The first question to be answered is: which filter has the most amount 
of smoothing for any given lag?

The filter optimisations in section 2.6.1 determine the smoothest filter parameters for each lag. A quick plot of the data serves an immediate idea as to the answer. Figure 2.19a shows the expected SI vs the expected lag for all the filters up to lag 50. Figure 2.19b shows the average SI across each price series for each lag. The only filter that provides any competition for the new set of filters (LGS, SLGS, ALGS) is the ALMA, which appears to be the smoothest filter for higher lags. The ill-conditioning of \( F \) and \( M \) is causing the spiking on the (S/A)LGS filters.

(a) Expected SI vs expected lag. The expected SI for each of the filters against the expected lag.

(b) Data-driven SI vs lag. The average SI for each of the filters across the asset series.

Figure 2.19: Smoothness Index vs Lag. The spikes on the (A/S)LGS filters are due to the ill-conditioned \( F \) and \( M \) matrices in the filter equation.
Figure 2.20 shows the percentage of lags on the range 1-50 that each filter has the highest expected SI. Together, the ALGS and LGS dominate by covering 86% of the range while the ALMA is the best filter for the top 14%.

Figure 2.20: Expected lags with the highest expected SI. Each square marks which filter has the greatest \( E[SI] \) for that particular lag. The values on the right are the percentage of lags that each filter is the smoothest for.

Figure 2.21 shows the percentage of lags on the range 1-50 for which each filter has the highest SI. Every filter was applied to every price series for every lag. The SI and lag were recorded. Every bar in Figure 2.21 is the percentage of lags that a filter is the smoothest for on a particular dataset. The figure indicates the filters but not the price series. The “Average” value is the average percentage value of each filter. The “Average” value is the average percentage value of each filter. Just as with the expected SI, the ALGS and LGS dominate by covering, on
average, 86.86% of the lags. Again, the only other filter that could be a competitor is the ALMA reaching, on average, 13.43% of the lags.

It is then clear that the new family of filters has the highest level of expected SI and measured SI overall for the lags 1-50. But this superiority is lost on higher lags due to the poor condition of the F and M matrices.

2.6.2.2 Comparing Lag

The second question to be answered is: which filter has the least amount of lag for any given SI?

In the previous section (2.6.2.1), it was simple to compare the SIs for each lag as it is possible to obtain the SI (measured or expected) at every lag. This is not true in reverse. That is, one cannot obtain a measure of lag for any SIs. SI is a function of lag, but lag is not a function of the SI. The results from the last section can be used after some preprocessing.

The preprocessing is simple. Using the same results from the previous section, the lag for each filter is linearly interpolated onto the SI. The SI range starts at the floor of the smallest recorded value and progresses at integer increments to the ceiling of the largest measured value. The results are shown in Figure 2.22a. The same interpolation process was performed on the average measured SI for each of the filters across the datasets, and is displayed in Figure 2.22b.

![Figure 2.22: Lag vs the Smoothness Index. The lag was interpolated onto the SI to provide a measure at each integer increment of the SI. The interpolation hides the spikes in figure 2.19.](image-url)
CHAPTER 2. SMOOTHING STOCK PRICES

Figure 2.23: Expected SIs with the lowest expected lag. Each square marks which filter has the lowest expected lag for that particular SI. The values on the right are the percentage of SIs where each filter has the least amount of lag.

Figure 2.24: Data-driven SIs with lowest lag. The graph shows the percentage of SIs for which each of the filters has the lowest lag. Each bar represents one dataset and the “Average” value is the average percentage value of each filter.

Figure 2.23 shows the percentage of expected SIs where each filter has the least amount of lag. Together, the ALGS and LGS dominate by covering 80% of the range, while the ALMA is the best filter for the top 10% and the SG filter the best for the bottom 10%. The data-driven perspective is shown in Figure 2.24. Each bar represents a filter (labelled) on a dataset while the “Average” value is the average percentage of each individual filter across the asset time series. As before, the lag was interpolated onto the SIs for each dataset. There was no averaging in this experiment. Again, it is clearly seen the ALGS and LGS filters dominating at a sum total of approximately 86.1% with the ALMA as the closest alternative at 8.59%.
Overall, the new family of filters has the lowest amount of lag for a given SI. More specifically, the new filters are generally the best over the SI range of 3 to 18 as shown by figure 2.23.

### 2.6.2.3 Comparing Fitness

Here, we discover whether or not the new family of filters presented in this thesis sacrifice fitness for their smoothing ability.

Figure 2.25a displays the expected error against the expected lag. It appears that there is a sacrifice in fitness for the new family of filters. The line for the ALMA is below any of the (A/S)LGS filters for most of the lags and the SG filter is almost on par with the ALGS.

![Expected fit vs expected lag.](image1)

Figure 2.25b shows the average fit between the filter output and the asset prices against the lag. Each line is the average fit of a filter across all the time series. The graph is extremely similar to the expected output in Figure 2.25a. Again, it appears that there is a sacrifice in fitness for the new family of filters. The line for the ALMA is below any of the (A/S)LGS filters for most of the lags.

Figure 2.26 shows that the ALMA has the least amount of expected error for 76% of the lags while the SLGS only covers 10%. Figure 2.27 shows the percentage of lags for which each filter has the least error for each of the data sets. Just as with the expected error, the ALMA produces the best fit for 76% of the lags; the ALGS and SLGS cover 3.57% and 7.86% respectfully.
Figure 2.26: Expected lags with the lowest expected fit. Each square marks which filter has the lowest expected fit for that particular lag. The values on the right are the percentage of lags where each filter has smallest fit.

Figure 2.27: Data-driven lags with lowest fitness. The graph shows the percentage of lags that each of the filters has the lowest fit. Each bar represents one dataset and the "Average" value is the average percentage value of each filter.

Some increase in error is to be expected with the increase in smoothing ability. In comparison to the remaining filters, the improvement of smoothing does not decrease the fitness substantially.

2.7 Conclusions

This chapter has refined the concept of smoothness, developed a measure for lag and presented a new family of filters. Whittaker [274] used the concept of smoothness to graduate time series. The measure of smoothness was taken and extensively built upon to allow for different levels of
smoothing and designed to have a scale independent of the data. The method for calculating the expected smoothness index given a linear filter was also shown. Measuring the lag of a filter was discussed. A new method of calculating the lag of a filtered asset price series and the expected lag of a filter was presented and demonstrated to be accurate.

Combining the new measures for smoothness and lag with a measure of fitness, three new moving averages, LGS, SLGS, and ALGS were fashioned. The filters are specially designed to optimally smooth a time series whose changes are Gaussian. The filters were compared against standard alternatives and found to be superior in two ways. The (S/A)LGS filters were, overall, the smoothest and had the least amount of lag. For this extra ability, a small amount of fitness must be sacrificed.

As promised in the introduction, I have made the following contributions in this chapter:

1. A measure of time series smoothness called the Smoothness Index (SI) (Section 2.3.1.1).
2. The expected SI of any linear filter given some basic assumptions about asset prices (Section 2.3.1.2).
3. The expected lag of any linear filter given the same basic assumptions (Section 2.3.2.1).
4. The expected error (fit) between any linear filter and an input time series that follows some basic assumptions (Section 2.3.4.3).
5. A filter which maximises smoothness after learning the patterns in any input time series (Section 2.4.1).
6. The expected filter that maximises smoothness given some basic assumptions about the input time series (Section 2.4.2).
7. Experiments and their results using the contributed measures of smoothness, lag and fit to compare different moving averages and the smoother just developed (Section 2.6).

The linear filters are minimising the amount of lag that must be accepted. It is conceivable that the gap between the end of the filtered curve and the current moment may be filled in by estimation. This is the subject of the next chapter.
Chapter 3

Forecasting the Smoothed Prices

In the 20th century, the United States endured two world wars and other traumatic and expensive military conflicts; the Depression; a dozen or so recessions and financial panics; oil shocks; a flu epidemic; and the resignation of a disgraced president. Yet the Dow rose from 66 to 11,497.


Market participants appear to hold very different strategies. On one side, there are the investors who spend their days sifting through mountains of financial statements, assessing the health of companies and their intrinsic value – bargain hunters. On the other hand, there are the technical analysts who stand by their mathematical models of asset behaviour. They spend their time inventing new forecasting systems based on price calculations to drive their decision making – automated trading. In 2009 73% of trading in the United States was conducted by computer algorithms [139]. Both of these cultures successfully exist along side each other. The thought of coexistence is astounding when considering the rapid rise in power of mathematical approaches and their relative infancy.

This chapter is certainly about forecasting. Understanding what that exactly means is sensible before further discussion. Predicting asset
prices to an accuracy where an individual can place 100% confidence is not possible. Some researchers do assert to come extremely close; for example, [13, 262]. Yet, they go somewhat un-referenced and ultimately unverified for their extraordinary claims.

Using the data model (2.1) discussed in chapter 2 we really cannot forecast stock prices. The best we can do is say that \( \log(y_{t+1}) = \log(y_t) \). However, we can forecast and make inferences on the future filtered asset price. The extrapolation is an estimate of the average or most probable future.

Demonstrations are made as to which filter is the best for forecasting. However, this is by no means the focus. The intention is to discuss the ability to extrapolate any linear filter. Currently, the smoothest filters are the family of filters developed in chapter 2. Extrapolating is merely a means to extend the filter to the present moment.

This chapter will demonstrate a logical forecasting algorithm that could not be out-performed by the more advanced techniques of recursive least squares, kernel recursive least squares and singular spectrum analysis. A quick discussion of experiments testing advanced methods will show this to be the case. In addition, the expected forecast error of a filter will also be shown.

### 3.1 Model

The model is so simple it hardly warrants its own section: project the logged price forward in a horizontal line then filter it. Since the distribution of changes has an assumed mean of zero (see Section 2.2), the average future values are likely to be the same as the most recent value. Figure 3.1 demonstrates this visually.

Mathematically, if we know everything up to time \( t \) we can calculate the filter up to time \( t \) giving us \( \tilde{y}_t \):

\[
\tilde{y}_t = \sum_{i=1}^{n} w_{n-i+1} y_{t-i+1}.
\]

Now, all we know about \( t + 1 \) is that \( E[y_{t+1}] = y_t \) and we can use this to
calculate an estimate of $\tilde{y}_{t+1}$:

$$\tilde{y}_{t+1} = \sum_{i=1}^{n-1} w_{n-i}y_{t-i+1} + w_n y_t.$$ 

Adding in a horizon term, we can simply write:

$$\tilde{y}_{t+h} = \sum_{i=1}^{n-h} w_{n-i}y_{t-i+1} + y_t \sum_{i=1}^{h} w_{n-i+1}. \quad (3.1)$$ 

The reason we forecast the filtered data is if $\tilde{y}_t$ tells us the best decision to make at time $t - l$ then maybe we could use $\tilde{y}_{t+l}^*$ to infer the most likely best decision for today. We are not trying to get a forecast of
We are trying to get a forecast of $\tilde{y}_{t+h}$ to infer what action should be taken today.

### 3.2 Measuring the Model Error

Calculating the error simply uses the MSE:

$$\frac{1}{N}||\tilde{y}^* - \tilde{y}||^2,$$

where $\tilde{y}^*$ is the forecast of the filtered prices $\tilde{y}$.

It is easy to calculate the expected error assuming that the underlying time series follows our standard model (2.1) and the forecasting model is (3.1).

There are two things an analyst may be inclined to forecast. The first is forecasting the filter as has been discussed so far. Yet, when analysts/traders convert the filter output into a buy/sell signal the rate of change is the focus. This will become apparent in chapter 4. The only thing to note now is that the expected error will not be the same when considering a differenced curve.

#### 3.2.1 Expected Error when Forecasting the Filter

The mean squared error equation is:

$$Q = \frac{1}{N}||\tilde{Y} F_{0,n,h} w - \tilde{Y} w||^2,$$

where $F_{0,n,h}$ is an $n \times n$ matrix that turns the filter ($w$) into its $h$-step forecaster with right padded zeros. Recall the model in (2.1) where the changes are random with a mean of zero. Thus, the expected logged price tomorrow is today’s logged price. So, when forecasting one step ahead, instead of multiplying the last weight $w_n$ by tomorrow’s value, multiply by today’s value. Then, the filter weight for today is $w_{n-1} + w_n$. For
example, if the size of \( w \) is 5 and the horizon is 2:

\[ F_{0,5,2} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}, \]

\[ F_{0,5,2}w = [w_1, w_2, w_3 + w_4 + w_5, 0, 0]^T. \]

Taking the expected value:

\[ E[Q] = E[(\hat{y}_t(F_0 - I)w)^2]. \quad (3.2) \]

A row of \( \hat{Y} \) looks like

\[ \hat{y}_t = s + [\epsilon_1, \epsilon_1 + \epsilon_2, \epsilon_1 + \epsilon_2 + \epsilon_3, \ldots] = s1_n^T + \epsilon_t^T U, \]

where \( U \) is an upper triangular matrix of ones. Putting this back into (3.2) and expanding:

\[ E[Q] = E[(s1_n^T + \epsilon_t^T U)(F_0 - I)w)^2] = E[(s1_n^T F_0 w - s1_n^T w + \epsilon_t^T UF_0 w - \epsilon_t^T Uw)^2]. \]

Since \( 1_n^T F_0 = 1_n^T \) and of course \( 1_n^T w = 1 \):

\[ E[Q] = E[(s - s + \epsilon_t^T UF_0 w - \epsilon_t^T Uw)^2] = E[(\epsilon_t^T U[F_0 - I]w)^2]. \]

Expand the power:

\[ E[Q] = E[w^T (F_0 - I)^T U^T \epsilon_t \epsilon_t^T U (F_0 - I)w] = w^T (F_0 - I)^T U^T E[\epsilon_t \epsilon_t^T] U (F_0 - I)w. \]

Assuming \( \mu \approx 0 \) then:

\[ E[\epsilon_t \epsilon_t^T] = \sigma^2 I + \mu^2 J = \sigma^2 I, \]

\[ E[Q] = \sigma^2 w^T (F_0 - I)^T U^T U (F_0 - I)w = \sigma^2 ||U(F_0 - I)w||^2. \]
We can avoid the matrices by computing:

$$E[Q] = E[MSE_f(w, h, d = 0)] = \sigma^2 \sum_{i=1}^{h} \left( \sum_{j=1}^{i} w_{n-j+1} \right)^2.$$  

For a scale free metric, simply drop $\sigma^2$, I will do this from now on.

### Important Equation(s) 8 (Forecasting Error)

We can measure the normalised mean squared error of forecasting a linearly filtered time series where the underlying series follows the model in (2.1) with:

$$NMSE_f(y, w, h, d = 0) = \frac{1}{\sigma^2_N} ||\tilde{Y}F_{0,n,h}w - \tilde{Y}w||^2. \quad (3.3)$$

Normalised simply means that the measure has been adjusted for variance, i.e. multiplied by $1/\sigma^2$.

We can measure the expected normalised mean squared error of forecasting a linearly filtered time series that follows the data model (2.1) with:

$$E[NMSE_f(y, w, h, d = 0)] = \sum_{i=1}^{h} \left( \sum_{j=1}^{i} w_{n-j+1} \right)^2. \quad (3.4)$$

This is also the expected forecasting variance and can be used to construct confidence intervals for the forecast values. For example, if the variance is left in the equation, then the standard deviation of errors is $\sqrt{E[MSE_f(y, w, h, d = 0)]}$.

Even though we know that (3.4) is correct due to the derivation we can run a quick experiment to visually see that it is accurate. (a) Generate a random vector of a random length between 2 & 50 from a standard normal distribution, take the cumulative sum, shift the result so that the minimum value is equal to zero and then normalise the vector to sum to one. This is our filter weights. (b) Generate a random price series of size 2000 with changes drawn from a standard normal distribution. (c) Randomly pick one of the asset prices and log it. (d) Randomly choose a horizon between 1 and half the length of the filter. (e) Calculate and
save the \( E[\text{MSE}_f(w, h, d = 0)] \) and \( \text{MSE}_f(w, h, d = 0) \) on the random and real data. Do (a) to (e) 1,000 times. We ought to see a strong correlation between the error and its expected value. The results are shown in Figure 3.2. Both the random data and the real world data display some heteroscedasticity and they are both highly correlated which is all we need.

![Figure 3.2: Correlation of the forecast error and its expected value. The real data is the time series presented in Section 2.2.](image)

3.2.2 Expected Error when Forecasting Change

Deriving the expected error when forecasting the rate of change is almost the same as forecasting the original curve. We difference the logged price data first, then apply the filter. This is the same as applying the filter and then differencing.

The mean squared error equation is:

\[
Q = \frac{1}{N} || \mathbf{D}_1 \mathbf{Y} \mathbf{F}_{1,n,h} \mathbf{w} - \mathbf{D}_1 \mathbf{Y} \mathbf{w} ||^2 = \frac{1}{N} || \mathbf{D}_1 \mathbf{Y} (\mathbf{F}_{1,n,h} - \mathbf{I}) \mathbf{w} ||^2,
\]

where \( \mathbf{F}_{1,n,h} \) is an \( n \times n \) matrix that turns the filter (\( \mathbf{w} \)) into its \( h \)-step first difference forecaster with right padded zeros. Essentially, we are filtering and forecasting the series of changes \( \mathbf{D}_1 \mathbf{Y} \). Recall the model in (2.1) where the changes are random with a mean of zero. Thus, the expected change is 0. So, when forecasting one step ahead, instead of multiplying the last weight \( w_n \) by tomorrow’s change \( y_{t+1} - y_t \), multiply by 0. For example, if the size of \( \mathbf{w} \) is 5 and the horizon is 2:

\[
\mathbf{F}_{1,5,2} = \text{diag}([1, 1, 1, 0, 0]),
\]


\[ \mathbf{F}_{1,5,2} \mathbf{w} = [w_1, w_2, w_3, 0, 0]^T, \]

where \( \text{diag}(\mathbf{a}) \) means a square matrix with \( \mathbf{a} \) down the diagonal and zeros everywhere else.

It is easy to see that a row of \( \mathbf{D}_1 \hat{\mathbf{Y}} \) looks like:

\[ [\epsilon_1, \epsilon_2, \epsilon_3, \ldots ] = \mathbf{e}_i^T. \]

Taking the expected value:

\[ E[Q] = E[(\mathbf{e}_i^T(\mathbf{F}_{1,n,h} - \mathbf{I})\mathbf{w})^2] = \mathbf{w}^T(\mathbf{F}_{1,n,h} - \mathbf{I})^TE[\mathbf{e}_i\mathbf{e}_i^T](\mathbf{F}_{1,n,h} - \mathbf{I})\mathbf{w}. \]

Assuming \( \mu \approx 0 \) then:

\[ E[\mathbf{e}_i\mathbf{e}_i^T] = \sigma^2\mathbf{I} + \mu^2\mathbf{J} = \sigma^2\mathbf{I}, \]

\[ E[Q] = \sigma^2|| (\mathbf{F}_{1,n,h} - \mathbf{I})\mathbf{w} ||^2. \]

For a scale free metric, simply drop \( \sigma^2 \). This can be made more efficient by dropping the matrices and computing:

\[ E[Q] = E[\text{MSE}(\mathbf{w}, h, d = 1)] = \sigma^2 \sum_{i=1}^{h} w_{n-i+1}^2 \]

Measuring the change forecasting error is highlighted in (3.5) and (3.6).

The equation (3.6) is also the expected variance of forecasting error which can then be used to build confidence intervals.

We can verify that 3.6 is accurate by repeating the experiment from figure 3.2. This time, however, we are using \( E[\text{MSE}_f(\mathbf{w}, h, d = 1)] \) and \( \text{MSE}_f(\mathbf{w}, h, d = 1) \). The results are shown in Fig. 3.3. The accuracy for both the random data and real data is satisfactory.
Important Equation(s) 9 (Change Forecasting Error)

We can measure the normalised mean squared error of forecasting a differenced linearly filtered time series where the underlying series follows the model in (2.1) with:

\[
\text{NMSE}_f(y, w, h, d = 1) = \frac{1}{\sigma^2 N} \| D_1 \hat{Y}(F_{1,n,h} - I)w \|^2. \quad (3.5)
\]

Normalised simply means that the measure has been adjusted for variance, i.e. multiplied by \(1/\sigma^2\).

We can measure the expected normalised mean squared error of forecasting a differenced linearly filtered time series that follows the data model (2.1) with:

\[
\mathbb{E}[\text{NMSE}_f(y, w, h, d = 1)] = \sum_{i=1}^{h} w_{n-i+1}^2. \quad (3.6)
\]

Figure 3.3: Accuracy of the forecast error of the rate of change and its expected value. The graphs show the accuracy of \(\mathbb{E}[\text{MSE}_f(y, w, h, d = 1)]\) on randomly generated data and real world data.
3.3 Filter Comparison

The forecasting ability of the linear filters will not be compared on real datasets. The aim is to compare the differences between filters. Thus, only the expected error is needed.

Looking at a filter’s weights gives a good idea of its predictive ability. Observe in (3.4) & (3.6) that the error will be small if the last $h$ (horizon) number of weights are small. Obviously, the error increases as the horizon increases but it increases at a power law of the $h$ last weights. Therefore, expecting that the (S/A)LGS family of filters and the ALMA will have a very low error for small horizons is reasonable. These filters can also be expected to have a much larger error for longer horizons close to their lag, possibly larger than the simpler filters (SMA, WMA, HMA).

Figure 3.4 shows the expected forecasting errors for lags 10, 20 and 50. As indicated in the current discussion, the expected forecasting error increases according to a power law as the horizon increases. Determining anything else from these two figures is difficult as the difference between some of the horizons is indistinguishable. Turn to figure 3.5 to see details on the best filter for each horizon. For both $d = 0$ and 1 the new family of filters, predominantly ALGS, dominate the expected forecasting performance for lag 10. However, this superior performance is quickly handed to the ALMA filter as the lag is increased. The ALGS is the best for 100% of the $d = 0$ horizons and 90% for $d = 1$ at lag 10. The ALMA almost completely takes over at lag 50 with 82% of the $d = 0$ horizons and 88% for $d = 1$.

Now that we have seen the forecasting performance for different horizons, consider the performance for extrapolating the filter all the way to the current time; that is, where the horizon equals lag. I used the smoothest parameters for each lag found in section 2.6.1 and calculated the expected error when forecasting as far as the expected lag. The results for predicting the curve are displayed in Figure 3.6. The new family of filters together have the lowest $E[MSE]$ for 98% of the lags from 1–50. However, Figure 3.7 shows that, when forecasting the rate of change, the new family of filters are over-come by the basic moving averages (WMA, SMA, HMA) which are at 90% in contrast to the ALGS at 4%.
Figure 3.4: Expected errors for each filter and horizon. The results for forecasting both the original curve ($d = 0$), the first difference ($d = 1$), and lags 10, 20, and 50 are included.
Figure 3.5: Best filter for each horizon. Each sub-graph shows which filters have the lowest expected error for a given horizon. Both $d = 0, 1$ are shown for lags 10, 20 and 50. Each square indicates that filter has the least expected error for that horizon.
Figure 3.6: Expected errors when extrapolating out the filter lag. The error when forecasting up to the current time period. The top shows the error for each lag and the bottom shows the best filter for each lag. Each square shows that the corresponding filter is the best for the corresponding lag.
Figure 3.7: Expected errors when extrapolating out the rate of change vs lag. The error when forecasting the rate of change up to the current time period. The top shows the error for each lag and the bottom shows the best filter for each lag. Each square shows that the corresponding filter is the best for the corresponding lag.
3.4 Alternative Predictive Algorithms

The beauty of the forecasting model demonstrated in the previous section is its simplicity. Throughout the course of this research attempts were made to develop more accurate models. Despite the well developed theory currently available in the forecasting literature, not a single proposed model was able to come close to rivalling the simple model on real datasets.

Algorithms that were explored built upon recursive least squares [203], kernel recursive least squares [203], and singular spectrum analysis [135]. They were combined to forecast the filter output, forecast the rate of change, and also analyse higher derivatives and then combine forecasts in an ensemble method.

Some of the best forecasting models exploit some clear and simple feature about the time series in question. For example, an additive trend and seasonal exponential smoothing model is used to exploit the clear direction of the data and the obvious repeated oscillations. In the current situation, we don’t know anything remotely concrete about the trend nor about any cyclical behaviours. Only one characteristic is guaranteed; the filter output is smooth. In this thesis, a smooth curve is as small as linearly possible at some differencing level $d$: $\min\{||D_d\tilde{yw}||^2\}$. A by-product of this is that all differencing levels $i < d$ are larger: $||D_i\tilde{yw}||^2 > ||D_d\tilde{yw}||^2$ and also smooth.

A useful property of differencing is that it removes any possibility of a trend. Each differencing level oscillates around zero. The data does not continue moving indefinitely in any direction; it must return to zero. There are clear oscillations.

The best 1-step ahead forecast of a smooth curve may be made with some $d^{th}$ derivative. Unfortunately, for longer horizons there is a problem. Because reversing the differencing involves summing (integrating) the series, errors will propagate forward in significant multiples. For example, forecasting 2 steps ahead at $d = 8$ will result in two error values at this level; $e_1$ is the error for the first step while $e_2$ is the error for the second step. After reversing the transformation, the 1st step error remains at $e_1$
while the 2nd step error is now $8e_1 + e_2$.

If individual forecasts were made at each differencing level up to $d$, it seems feasible to think that an algorithm to then find the forecast that is closest to all forecasts would yield a better result.

Further discussion and detailed results are omitted from this thesis. Needless to say, the results were unsatisfactory. The models based on recursive least squares, kernel least squares, singular spectrum analysis, and the proposed method for merging forecasts at successive derivatives were unable to achieve a forecast error (MSE) below twice the error of the simple method used in this chapter.

\section*{3.5 Conclusions}

The majority of this chapter contributes to a simple forecasting model. This model, after tremendous effort to beat, is the best model. The new family of filters, (A/S)LGS, were shown to have the least amount of expected error for the smaller lags when comparing horizons. The ALMA has superior performance over the higher lags. When examining only the case when the horizon equals the expected lag, the new family of filters provide the best model to forecast the smooth filter output. However, when forecasting the rate of change, the simple filters are the best. Finally, the efforts spent to find a forecasting algorithm to improve upon the benchmark were briefly discussed.

It now remains to be seen how much of an effect smoothing has on profitability.
Chapter 4

Effects of Smoothing on Profit

This planet has - or rather had - a problem, which was this: most of the people living on it were unhappy for pretty much of the time. Many solutions were suggested for this problem, but most of these were largely concerned with the movement of small green pieces of paper, which was odd because on the whole it wasn’t the small green pieces of paper that were unhappy.


Every single study that I examined in the literature review, using filters to milk money from markets, did not ask why do smoothers work or not work? Trading with moving averages has not been demonstrated – without a shadow of doubt – to be successful. Surely, with trillions of dollars at risk, understanding why you are moving large quantities of imaginary green goods is imperative.

Writings published outside of academic circles discuss countless investment strategies. Author and fund manager Ernest Chan [57] and analyst David Aronson [16] warn that trading suffers from data-mining bias. This is the tendency to find excellent results in a huge pool of potential methods; some methods will randomly be better than others. Data-mining bias offers an explanation for the enormous quantity of trad-
ing systems. One major trading strategy is prevalent among academic circles; the dual moving average system. This chapter will only be concerned with this plan because it is the most common price smoothing tactic.

In this chapter, we will see that the popular dual moving average trading plan, as its name suggests, uses two filters. However, the stratagem is identical to investing with only one filter. One smoother is easy to analyse. On the other hand, systems with two filters have more parameters than can be comfortably explored. We know that applying these filters to purge fluctuations from prices seems like a practical idea. The majority of the chapter will explore how smoothing affects profits and if it gives a financial “edge.”

I define an edge as a process or information that can be consistently used to improve investing performance. That is, to reduce losses and/or increase profits.

The exploration begins by breaking down the dual filter rule discussed in section 1.3.3.

4.1 Trading System

The trading strategy used here is what I call the single filter rule (SFR). In this tactic, the prices are filtered and we simply buy when the curve is going up, and sell when it is going down. Figure 4.1 shows an example of this system. Lag has been removed so that the curve lines up with the prices perfectly and you can see the rationale behind this strategy. The change in the curve is graphed below the prices. When the curve is going up, the change is positive; when the curve is going down, the change is negative. The buy and sell actions take place as the change in the curve crosses the 0 line.

The specific trading strategy in this thesis is:

1. Filter the logged prices as per the model in (2.1).

2. When the filter turns from down to up, close any selling positions and open a buy trade. This counts as two individual transactions
and incurs the cost of two transactions.

3. When the filter turns from up to down, close any buying positions and open a sell trade. This counts as two individual transactions and incurs the cost of two transactions.

4. Each transaction costs 0.1%.

A transaction cost of 0.1% has been used in previous studies such as [224].

During the literature review, I did not find any studies of moving averages that trade with a single filter. They all use two filters, the dual filter rule (DFR). The next section shows that the DFR is exactly the same as the SFR with the exception of the filter weights.

The studies that I examined in the literature review make use of the dual filter rule (DFR). They all analyse a small subset of parameters of the filters that vary their noise reduction capabilities. Some of the parameter settings completely turn off the smoothing on one of the filters. This results in a trading system with only one filter. However, the context is still on two filters. This poses a problem for this thesis because the
smoothness index measures the smoothing of only one filter.

The next section shows that the DFR is exactly the same as the single filter rule (SFR) with the exception of the filter weights. This result allows us to confidently use only one filter knowing that any conclusions can be compared to the research on the DFR.

4.1.1 Similarity Between the SFR and the DFR

The dual filter rule (DFR) uses the difference between two filters as a signal to buy or sell. The filter with the least lag is called the short filter, the other, the long filter. Figure 4.2 provides an example of this system. When the short MA moves below the long one, sell. When the short MA moves above the long, buy. The lower part of the figure shows the difference between the two filter outputs. As with a single filter, the decision takes place when the two filters cross. This is at zero in the lower part of the figure. The rest of this section shows how this is the same as the single filter rule (SFR).

Figure 4.2: Dual filter rule example. When the short (red) filter moves below the long (green), sell and vice versa. The oscillating signal around zero is the difference between the two filters. The term LGS(10, 11) means the linear Gaussian filter with lag = 10 and \(d = 11\).

The first step is to combine the two filters. The difference between
the two filter outputs can be written as

\[ \vec{Y}Pw_s - \vec{Y}w_l, \]

where \( P \) pads the top of \( w_s \) (the short filter) with zeros so that it is the same length as \( w_l \) (the long filter). The matrix \( \vec{Y} \) is the time series trajectory matrix such that \( \vec{Y}w_l \) is the filtered time series. This equation can be rearranged into a single filter:

\[ \vec{Y}Pw_s - \vec{Y}w_l = \vec{Y}(Pw_s - w_l) \]

\[ \text{DMA}_1(w_s, w_l) = Pw_s - w_l, \]

where DMA stands for dual moving average and \( \text{DMA}_1 \) indicates the first step; this is illustrated in Figure 4.3.

\[ \text{LGS}(10, 11) \quad - \quad \text{LGS}(20, 6) = \text{DMA}_1 \]

\[ \begin{align*}
1 & \quad 21 & \quad 41 \\
\text{LGS}(10, 11) & \quad - & \quad \text{LGS}(20, 6) & \quad = & \quad \text{DMA}_1 \\
1 & \quad 41 \\
\end{align*} \]

**Figure 4.3:** Step 1 converting two filters into one. Continuing with the example in figure 4.2; the filter weights that will produce the oscillating decision signal equals the short filter weights, LGS(10, 11), minus the long filter weights, LGS(20, 6). The short filter weights are padded on the left with zeros to match the length of the long filter.

The single moving average system makes the buy and sell decisions using the change in the filter’s output. The change is calculated as \( \tilde{y}_t - \tilde{y}_{t-1} \) which is also a linear filter. Two linear filters can be combined into one equivalent linear filter. This is accomplished through convolution. Given some arbitrary filter \( u \), the weights that filter and then calculate the change can be written as:

\[ d_1 * u, \]

where \( d_1 = [-1, 1]^T \) and \( * \) is the convolution operator. Since this problem deals with a vector (discrete time), the above equation can be rewritten with linear algebra:

\[ d_1 * u = D_{n+1,1}^T u = \text{weights for filtering and differencing}, \quad (4.1) \]
where \( n \) here means the length of \( u \) and:

\[
D_{n+1,1}^{T} = \begin{bmatrix}
-1 & 0 & 0 & \ldots & 0 & 0 \\
1 & -1 & 0 & \ldots & 0 & 0 \\
0 & 1 & -1 & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & 1 & -1 \\
0 & 0 & 0 & \ldots & 0 & 1 \\
\end{bmatrix}.
\]

Now, the weights for the oscillating signal are produced by \( \text{DMA}_1(w_s, w_l) \). They can be treated as the change of a filter. To find the filter whose change is given by \( \text{DMA}_1 \) simply substitute into (4.1):

\[
D_{n+1,1}^{T} u = \text{DMA}_1(w_s, w_l) \\
\quad u = (D_{n+1,1}^{T})^+_l \text{DMA}_1(w_s, w_l) \\
\quad = \text{DMA}_2(w_s, w_l),
\]

where \( A_l^+ \) is the left pseudo-inverse of \( A \). Following the structure of \( D_{n+1,1}^{T} \) the pseudo-inverse is:

\[
(D_{n+1,1}^{T})^+_l = [-L_n, 0_n],
\]

where \( L_n \) is an \( n \times n \) lower triangular matrix of ones and \( 0_n \) is a zero column vector of length \( n \). The matrix \( (D_{n+1,1}^{T})^+_l \) produces a negative cumulative sum that skips the last value.

The filter \( \text{DMA}_2 \) now gives the dual moving average signal to buy if the output is going up and vice versa for the sell decision. The example is continued in Figure 4.4. The \( \text{DMA}_2 \) filter output looks like the original prices, but it is not close to the input, it is on a different scale. Ensuring
that the weights to sum to one rectifies the problem:

$$\text{DMA}(w_s, w_l) = \frac{\text{DMA}_2(w_s, w_l)}{1_n + 1 \text{DMA}_2(w_s, w_l)}$$

Of course, this will change the filter. However, the following equality holds thereby ensuring that the trading signals to buy or sell are exactly the same:

$$\text{sign}(D_t \mathbf{\bar{Y}} \text{DMA}(w_s, w_l)) = \text{sign}(D_t \mathbf{\bar{Y}} \text{DMA}_2(w_s, w_l))$$

Figure 4.5 is an update of Figure 4.2 that converts the two LGS filters into one DMA filter.

Merging two filters into one carries a series of benefits. Calculating a single SI is not possible on two smoothers, but simple for one. Thus, measuring the effect of smoothing on profit is easy if only using one filter. Also, except for the difference in filter weights, the two systems are identical. Thus, DFR can be ignored and the single filter scheme can be used. This brings the added benefit of reduced complexity. With two filters there is twice the amount of parameters to iterate through. Using only one filter is much faster.
4.2 Effect of Smoothing

In this section we finally explore the question that has driven this thesis so far; how does smoothing affects profits? In doing so, we will answer four more questions:

1. Is smoothing more profitable than not smoothing?
2. Is more smoothing more profitable?
3. What is smoothing actually doing?
4. Does forecasting the filter offer any improvement?

The meaning of “profitable” is relaxed for all market conditions. Suffering some financial losses is unavoidable. In these cases, investors who lose the least wealth are the winners. This work considers a reported loss profitable if the alternative is a larger loss. The goal, then, is to maximise the Sharpe ratio (see section 1.2.1.2), no matter if it is still below zero.

4.2.1 Methodology & Results

Because the experiments are quick and simple, they will be discussed together. First, the data generation process is covered. Then, we will explore the data to see the relationship between the smoothness index (SI) and performance. Then, the relationship is quantified and compared between datasets and whether or not there is a fee for transactions. Finally, the reason for the observed behaviour is discussed.

Data Calculation

Because all factors ought to be controlled as independently as possible, the symmetric linear Gaussian smoothier (SLGS) is ideal. It controls lag and smoothness independently of one another – aside from lag providing an upper limit on the SI.

The SLGS is applied to each price series with varying sets of parameters. The lag is iterated across the range $1 - 50$, $\gamma$ across $0 - 40$ at 20
evenly spaced points and \( d \) from \( 1 - 13 \). The maximum value of 40 was chosen for \( \gamma \) because the SI improves slowly at higher values, see Figure 2.13a. The parameter \( \gamma \) was spaced evenly at 20 points and \( d \) capped at 13 simply to limit processing time. In all, there are 13,000 different sets of parameters. The single moving average trading system was tested for each set of parameters for each data set and the following properties were recorded:

1. Expected lag.
2. Expected SI.
3. Number of transactions.
4. Sharpe ratio with fees.
5. Sharpe ratio without fees.

Properties 4 to 6 were also recorded for the corresponding forecast filter with a horizon set to equal the lag.

The one-way transaction cost is 0.1%. The term “one-way” indicates that the cost is charged with every independent transaction. For example, 0.1% is charged when purchasing shares and again when selling the shares. So, if shares are bought and the day rises by 0.3% then the profit for the day is recorded as \( 0.2\% = 0.3\% - 0.1\% \). If the following day declines by 0.1% and the shares are sold, the return for the day is \(-0.2\%\).

The literature examining investment and trading strategies has a consistent and natural focus on the returns. On first glance, this is a very important measure and it appears to be an appropriate objective to maximise. However, the financial world is a complex and very flexible arena. The most notable example is of borrowing. In the current environment of internet investing and trading, borrowing funds is only a click away. On some platforms, it is built right in to every trade. This research will not be using the returns as the objective and reporting measure. Picture a system that is expected to return 25% of your investment. Would you accept that if there was a good chance of losing 100%? Experiencing only one bad year will take you out of the market. This risk is magnified under borrowing. The Sharpe Ratio easily overcomes this problem by expressing return as units of risk. Section 1.2.1.2 covers the details of
CHAPTER 4. EFFECTS OF SMOOTHING ON PROFIT

the measure pointing out that maximising the Sharpe Ratio maximises the likelihood of outperforming the risk free rate [214].

Because the risk free rate changes with time and investor preference, we set the rate when calculating the Sharpe ratio to 0. This is an ordinary practice as seen in [104, 193].

For each of the datasets, the results are compiled as a surface. The three axes are $E[\text{lag}]$, $E[\text{SI}]$ and the Sharpe Ratio. The SIs are rounded to 10 decimal places, i.e. round(Sharpe Ratio $\cdot 10^{10}$) $\cdot 10^{-10}$ - this allows significant speed up of specialised sorting algorithms which require integers. The Sharpe Ratio is calculated with the mean and standard deviation of the sample returns which have the transaction costs subtracted at the time they are incurred. Any Sharpe ratios at identical points of lag and SI are averaged.

An “overall” surface is also built using all the results from all the datasets. The Sharpe ratios are scaled between 0 and 1 for each of the data sets and then averaged across all datasets for each $E[\text{lag}]$ & $E[\text{SI}]$ point.

There are separate surfaces for the Sharpe ratio with and without transaction costs. The two types of surfaces tell us what happens in real life and the market ‘edge’ the strategy provides respectively.

Data Exploration & Results

The overall surface including costs is presented in Figure 4.6. Three things can be quickly observed. First, smoothing certainly provides a financial benefit to a trading scheme. The Sharpe ratio is lowest when the SI is 0. Second, the Sharpe ratio increases with the SI at a decreasing rate. Lastly, the surface is undoubtedly chaotic with many sharp spikes. This last point provides a sufficient case to dismiss any sophisticated optimisation algorithms such as particle swarm optimisation in [150]. Algorithms will merely converge onto a spike. In such a situation, there is absolutely no guarantee that the Sharpe ratio will also sit on an upward spike on future data.

If trading were to be considered free of charge, the Sharpe Ratio becomes an excellent measure of how much information smoothness con-
CHAPTER 4. EFFECTS OF SMOOTHING ON PROFIT

Figure 4.6: 3D Plot of Sharpe Ratio with Fees. This is the overall surface of the Sharpe ratio against expected lag and SI. Each price series has the Sharpe ratio scaled between 0 and 1 before the mean of each point is calculated.

tributes. It can be interpreted as the amount of gross income per unit of risk. If the performance is poor, smoothness does not contribute useful information. If the performance is much better, smoothing provides useful information. With the proliferation of moving average methods in the literature, one might expect experiments to show that smoothness does contribute profitable information.

Examine the difference in Sharpe Ratios between Figures 4.6 and 4.7 (a 2D comparison is in figure 4.8). Both graphs are exactly the same except Figure 4.7 does not include investment costs; the investment process is treated as free-of-charge. When transaction expenses are included, there is a beautiful rise of the Sharpe ratio with the SI. However, when fees are ignored, there is no trend or any other indications that smoothness is beneficial.

Clearly, because of the difference between Figures 4.6 and 4.7 the performance of smoothing is being affected by transaction costs. If the trading signal is not smooth, then there is frequent buying and selling. This increases the amount of money spent on transactions. If the trading signal is smooth, then the rapid changes in market direction disappear along with the transaction costs. Despite the widespread use of filters, smoothing appears to offer no actual “edge” to trading other than the
CHAPTER 4. EFFECTS OF SMOOTHING ON PROFIT

Figure 4.7: 3D Plot of Sharpe Ratio without Fees. This is the overall surface of the Sharpe ratio against expected lag and SI. Transaction costs have been excluded. Each price series has the Sharpe ratio scaled between 0 and 1 before the mean of each point is calculated.

reduction of trading costs. Figure 4.9 shows the decrease in transactions as the SI increases.

The influence of investment expenses can be measured by calculating how many points on the surface are higher than when $E[SI] = 0$ and lag is the same as that point, and expressing it as a percentage of the total number of points. On the overall surfaces, smoothing is better 100% of the time when paying fees, while it is better 41% of the time when ignoring costs. Figure 4.10 shows these values for all the data sets. We can expect the value to fluctuate around 50% if smoothing is useless. Looking at the chart, this is the case for a free market – smoothing offers no advantage whatsoever. This answers the first question: is smoothing better than not smoothing? No, smoothing offers no financial “edge”.

This also answers the third question: what is smoothing actually doing? Smoothing does nothing more than minimize your trading toll.

The second question now must be addressed: does more smoothing give more profit? Unfortunately, the relationship between the Sharpe ratio and SI is not linear (see Figure 4.6). A non-linear approach must be used to gauge if the Sharpe Ratio is increasing or decreasing. Gaussian kernel regression is used to smooth the surfaces with a band-
CHAPTER 4. EFFECTS OF SMOOTHING ON PROFIT

Figure 4.8: 2D Plot of Sharpe Ratios With and Without Fees. These 2D versions of Figures 4.6 and 4.7 were created by removing the lag dimension. Each dot is a point on the surfaces. Each dot is coloured by lag. Blue means lag = 1 while red means a large lag of 50.

Figure 4.9: Percentage of Transactions. The number of transactions as a percentage of the number of price quotes. There are 364,000 points of which a random sub-sample of 10,000 points is displayed.

Figure 4.10: Percent Smoothing is Better. The percentage of the Sharpe surfaces that is higher than not smoothing.
width of 1. Then the change is calculated at every point on the surface in the SI direction. All the positive changes are summed ($\sum p$) and are expressed as a percentage of the sum of all absolute changes ($\sum |c|$): 

$$\frac{\sum p}{\sum |c|} \times 100.$$

I call this measure the Percent Positive Slope (PPS). The magnitude of the slope is used instead of the sign to differentiate between rapid and slow changes.

If the Sharpe Ratio does increase with the SI, the PPS ought to be in the vicinity of 100%. Otherwise, if the ratio randomly bounces around with no real direction, the kernel regression also should bounce up and down with a PPS around 50%. Figure 4.11 shows this statistic for all datasets including the overall surface. On the whole, when transaction costs are included, it seems to be a better idea to have the greatest amount of smoothing. When transaction costs are excluded, there is no benefit to smoothing.

![Figure 4.11: Percent of Positive Sharpe Ratio Slope. The percentage of Gaussian kernel regressed surfaces that have a positive SI slope. The metric is calculated with the magnitude of the slope rather than the sign.](image)

The results in Figure 4.11 are consistent except for one asset. Google (GOOG) has the lowest PPS, 70%, when fees are considered. The reason for this is quite elegant and satisfying.

The amount paid for each transaction does have an effect on different assets or time periods. The returns are measured in percent values, the same as the transaction cost. Yet, the volatility of the returns varies between datasets depending on the actual market for that asset or the time scale of the data set. For example, Google (GOOG) is more volatile
than Westpac (WBC) and hourly data is less volatile than daily data. Despite this variance in volatility between assets and time scales, the transaction cost remains the same. Rationally, the higher the return volatility, the less impact trading expenses ought to have.

![Figure 4.12: PPS vs Price Change Volatility. The symbol $\sigma^2$ is the variance of price changes.](image)

Figure 4.12 shows the relationship between the slope of the SI and the variance of the asset price returns. When transaction cost are accounted for; PPS is high when volatility is low and vice versa. The asset with the highest volatility and lowest PPS is Google (GOOG). The magnitude of price fluctuations is enough to absorb most of the trading fees. However, the 70% PPS achieved by GOOG is still greater than the highest cost-free PPS of 60%.

The theory behind the trading system and smoothing is sound; buy when the price is racing up, sell when plummeting down. The reason is lag. A quick experiment validates this claim; see Figure 4.13. Using the LGS at lag 20 and testing the strategy offsetting at every lag from 0 to 20 shows the change in results. When there is no lag, the annualized return is around 60% while at the correct lag of 20, it is close to 0%.

Figure 4.13 is actually cheating. Of course greater returns are seen when offsetting the lag because we are now trading using future information. The analysis so far has not included extrapolating the smooth output to compensate for the lag. This is the topic of the next section.
CHAPTER 4. EFFECTS OF SMOOTHING ON PROFIT

Figure 4.13: Effect of Lag on Profits. The AUD/USD was smoothed with the LGS at lag 20. The output was offset at every lag measuring the annualized Sharpe Ratio and return.

4.2.2 What about Forecasting?

An algorithm for forecasting was discussed in Chapter 3. The algorithm produces nothing more than a linear filter whose output is the expected future value of another filter. Thus, the previous section also measures the performance of forecasting if the expected lag and SI are calculated on the forecasting filter. However, this does not answer the relevant question: given a filter of a specific lag and SI, does forecasting offer any improvement?

As before, a visual overall examination will be presented before detailed results of each data set.

Figure 4.14 shows the overall Sharpe ratios for the original and forecasting filters. No scaling has been used. Each moving average was predicted ahead by its lag amount. The forecasting results are very much below the original for both paid and free scenarios.

Quantifying the difference between before and after forecasting is only a matter of calculating the percentage of the forecasting surface that is above the original surface. The two surfaces were first interpolated so that the points match up. Figure 4.15 shows the percentage values for each of the datasets (and overall) for both before and after costs. Some of the datasets show a random fluctuation around the 50% level. However, for most, forecasting is considerably worse. Many of the datasets, when transaction costs are included, report total poor performance (0%) when forecasting. In comparison to previous experiments, the results are highly inconsistent. Only one conclusion can be drawn; forecasting is not useful.
CHAPTER 4. EFFECTS OF SMOOTHING ON PROFIT

Figure 4.14: Overall Forecasting Performance. The blue clouds are the Sharpe ratios for the ordinary filters. The red clouds are the Sharpe ratios for the forecasting filters.

Figure 4.15: Percentage of Forecast Surface Above Original Surface. Surfaces were first interpolated to have identical $E[\text{Lag}]$ and $E[\text{SI}]$ points.

4.3 Other Filters

The previous section conclusively showed smoothing provides no real investing edge. The symmetric linear Gaussian smoother (SLGS) made this analysis easy. However, analysts and traders use filters that are asymmetric (i.e. ALMA) and often have a single parameter (i.e. WMA, HMA & SMA). These single parameter filters do not separate lag and smoothness. This section looks at these two filter types, beginning with the asymmetric filters.
4.3.1 Asymmetric Filters

The only filter that is not asymmetric in the literature is the simple moving average. The problem is that the others do not separate smoothing from lag. Thus, this section will only examine the Asymmetric Linear Gaussian Smoother (ALGS), developed in Chapter 2, and the Arnaud Legoux Moving Average (ALMA). The asymmetric filters that are single parameter will be dealt with accordingly.

The same experiments as in Section 4.2 were conducted on these two filters. The ALGS was iterated through sizes 2-200, $d$ from 1-13 and $\gamma$ from 0-40 at 20 equally spaced points. Also, because the ALGS filter does not have a setting for zero smoothing, the Shape ratios for 0 smoothing were added – they are only a lagging of price. The ALMA was iterated through sizes 2-200, and both $\sigma$ and offset ($o$) through 16 equally spaced values from 0-50 inclusive.

Figures 4.16 & 4.17 shows the ALGS surface plots and summaries. Figures 4.18 & 4.19 shows the ALMA results.

![Figure 4.16: ALGS Filter Performance. Each price series has the Sharpe ratio scaled between 0 and 1 before the mean of each point is calculated. The left graph shows the Sharpe Ratios with cost while the right ignores fees.](image)
CHAPTER 4. EFFECTS OF SMOOTHING ON PROFIT

Figure 4.17: ALGS Percent Smoothing is Better. The graph shows the percentage of the Sharpe Ratio surfaces that is higher than not smoothing.

Figure 4.18: ALMA Filter Performance. Each price series has the Sharpe ratio scaled between 0 and 1 before the mean of each point is calculated. This figure is the same as figure 4.16 only now with the ALMA.

Figure 4.19: ALMA Percent Smoothing is Better. The graph shows the percentage of the Sharpe Ratio surfaces that is higher than not smoothing. This figure is the same as figure 4.17 only now with the ALMA.
CHAPTER 4. EFFECTS OF SMOOTHING ON PROFIT

4.3.2 Single Parameter Filters

So far, it is conclusive that smoothing poses no real advantage when using flexible filters. However, traders use moving averages with a single input parameter that controls both lag and smoothing. The three most common linear moving averages are the simple, weighted and Hull.

In these experiments, the LGS filter was iterated over lags 1-50 and $d$ over $1 - 13$. Recall that $d$ controls the type of smoothing. A different $d$ can be considered a different filter. The SMA was iterated over lags 1-50, the WMA and HMA were iterated over all sizes starting at one and stopping when the lag exceeded 50.

![Figure 4.20: Basic Filter Results. The percentage of the overall surface that is higher when SI = 0 for both with and without trading costs.](image)

The percentages of the overall surfaces that are higher when SI = 0 are in Figure 4.20. When costs are included, 100% of the time smoothing is better than not smoothing and when costs are excluded only around 46% of the time; just randomly bouncing around.

I cannot offer an explanation as to why these filters are constantly considered profitable. I speculate that some people may be enticed by the idea that more smoothing helps to earn greater profit. The simple filters do actually show this. See Figure 4.21 which shows that the greater the SI, the greater an edge a trader has. Remember, however, this is only if you must smooth; it is not better than not smoothing. The LGS, which optimally controls smoothing does not exhibit any trend. There is another unknown factor at play. I suggest that since the smoothness of these filters is tied to lag, that lag may be having an effect.
Figure 4.21: Basic Filter Fallacy. The Sharpe ratio is scaled and averaged across each of the asset prices. In all but the LGS, the Sharpe ratio increases with the SI.
4.4 Conclusions

This chapter started by proving the dual moving average system is over-complicated and unnecessary for examining smoothness. A single moving average strategy is identical in all points that matter. Using one filter means it is easy to calculate the lag and SI associated with a strategy, but more importantly, the measure of return.

Four questions were asked and four simple answers have been found.

Is smoothing more profitable than not smoothing? No. Smoothing offers no financial “edge.”

Is more smoothing more profitable? No. There is no gain in gross profits as the SI increases.

What is smoothing actually doing? Smoothing is decreasing the number of transactions. Thus, when accounting for trading fees, smoothing does offer some advantage. More-over, more smoothing means less transactions and thereby less money spent on trading.

Does forecasting the filter offer any improvement? No. Performance decreases when extrapolating the filtered logged prices forward.

Not only are these answers clear, they are consistent across all datasets.
Chapter 5

Summary & Conclusions

The scientist does not study nature because it is useful to do so. He studies it because he takes pleasure in it, and he takes pleasure in it because it is beautiful. If nature were not beautiful it would not be worth knowing, and life would not be worth living. I am not speaking, of course, of the beauty which strikes the senses, of the beauty of qualities and appearances. I am far from despising this, but it has nothing to do with science. What I mean is that more intimate beauty which comes from the harmonious order of its parts, and which a pure intelligence can grasp.

– Henri Poincaré Science et Méthode (1908) in [235]

Any claim to possess a money-making strategy on financial markets refutes the efficient market hypothesis (EMH). The EMH states that a method does not exist to generate excess returns from financial markets [97,99]. Individuals can profit from investing, but they cannot earn more than the risk free rate plus compensation for any risk they take [26,169,178]. DeBondt and Thaler [33] compared portfolios made up of only losing companies against those of winning companies. They rejected the EMH after finding the losing portfolios were able to profit, while the winning portfolios could not. However, Jegadeesh & Titman [168] copied their experiment and also rejected the EMH on the
grounds that buying winning portfolios produces excess returns. Both
draw the same conclusion on contradicting evidence. Thus, I suggest
that methods of accepting or rejecting the validity of technical analysis
are not conclusive and are subject to reasonable contradiction.

One method of trading involves filtering the asset prices to remove
fluctuations. These filters are commonly known as moving averages
[29, 47, 55, 156, 158, 215]. The moving average usually used is the sim-
ple moving average [89]. Surprisingly, there is no measure of noise that
can be used to compare any moving average and any time series. The
signal to noise ratio [36] and mean squared error [227] require the original
clean data which is not possible to obtain for stock prices. Autocorrela-
tion analysis [204], power spectrum analysis, correlation dimension anal-
ysis [28, 167], and recursive analysis [131] output the results in a high di-
ension causing computational complexity issues on large experiments.
Whittaker’s [274] method of measuring smoothness is almost suitable.
Unfortunately, different filters and datasets sit on different scales with
this measure.

Because there is no method to measure the smoothness of an online
filter on asset prices, the effects of smoothing on profits are not known.

In chapter 2, I began by solving the measuring problem. I developed
two new metrics, the smoothness index (SI) and a measure of lag. I
derived their expected value given a linear filter and assumed the input
time series changes are Gaussian. The expected values do not require
knowledge of the Gaussian distribution, only that it is Gaussian; the
mean and variance are factored out. I also refined the error measure
between a filter’s output and the original time series by determining the
expected value of the mean squared error.

Using the expected SI and lag to understand the problem, I derived
a new family of three filters. They optimally control smoothing, do not
constrain to a polynomial model and do not require knowledge of the
underlying Gaussian distribution of price changes. The most basic form,
the linear Gaussian smoother (LGS), takes one parameter from the user
and an algorithm adjusts a second parameter to maximise the SI. The
symmetric linear Gaussian smoother (SLGS) adds another parameter so
that the user can control the amount of smoothing. Finally, the asym-
metric linear Gaussian smoother (ALGS) outputs an asymmetric filter
that increases the SI on the lower lags. However, lag is not so easily set.

The (S/A)LGS filters are significantly smoother than other linear competitors. However, this is not true for all lags. I conducted experiments comparing the SI for a range of filters for each lag from 1 to 50. The (S/A)LGS filters had the highest expected SI for 86 percent of the lags. The ALMA was the only competitive rival being the smoothest for lags 44 to 50. The (S/A)LGS filters fail here because of matrix ill-conditioning when calculating their weights. The new family of filters has the lowest lag for the SI range 3 to 18. This superior performance was achieved by sacrificing a small amount of fitness to the asset prices. However, the difference is small.

In chapter 3, we explored extrapolating a smoother’s output to compensate for lag. The forecasting model assumes that the average logged price change is zero. The (S/A)LGS filters have the least amount of error when forecasting short lags over all horizons, while the ALMA filter is the best when forecasting long lags. When extrapolating away all lag, the (S/A)LGS filters are the best for lags 2 to 50, failing only at lag 1. However, when forecasting the rate of change, the most accurate filters were the weighted and simple moving average. I investigated using more advanced models for prediction, however, none has an error below 200% of the basic algorithm on real world datasets.

In chapter 4, we explored using moving averages for trading. Specifically, we investigate the effect smoothing has on profits. The aim of investing is to maximise the Sharpe Ratio. This measure expresses return as units of risk. A larger Sharpe ratio means more profit for the risk. In this thesis, I have relaxed the definition of “profitable.” If a loss is made, but, the alternative is a larger loss, this strategy is profitable.

When including transaction costs, smoothing is profitable. In addition, as smoothing increases, so does investment performance, but this is solely due to fees. When trading expenses are ignored performance ought to increase. After all, less money is being spent for the same return. However, this is not the case. There is no difference between smoothing and non-smoothing when ignoring costs. Filtering asset prices simply reduces the number of transactions, which reduces the money spent on investing. There are no other advantages. Extrapolating the smooth curve decreases the performance further.
These results are consistent across the (A/S)LGS, SMA, WMA, HMA, and ALMA filters as well as across every asset in this study.

The ultimate conclusion of my thesis is that smoothing asset prices does not offer any financial advantages; it only reduces transaction costs.

5.1 Conclusions & Contributions

The following is a list of the contributions I have made in this work.

In this thesis, I:

1. Developed a new measure of smoothness termed the Smoothness Index (SI) (Section 2.3.1.1).

2. Derived the expected value of the SI assuming an underlying stochastic process with changes drawn from a Gaussian distribution with mean zero (Section 2.3.1.2).

3. Derived the expected value of lag assuming the same model for the underlying data and minimising the Mean Squared Error (MSE) between the filter input and output to find the lag (Section 2.3.2.1).

4. Found the expected error between a linear filter and the input series given the same underlying model and a lag value (Section 2.3.4.3).

5. Developed a filter which maximises smoothness given any input time series (Section 2.4.1).

6. Derived the expected filter given the same underlying model (Section 2.4.2).

7. Used the measures of smoothness, lag and fit to compare the different moving averages used for asset prices that I found in the literature (Section 2.6).

8. Showed that the newly developed filter is the smoothest for lags 1 - 43 and has the lowest amount of lag on the SI range of 3 - 18 (Section 2.6).

9. Showed that when forecasting a filter’s output given the same underlying model, the newly developed filter is the best when fore-
casting the filtered time series, but the SMA and WMA are the best filters to forecast when predicting the rate of change (Chapter 3).

10. Showed that smoothing offers no edge when excluding expenses (Section 4.2).

11. Showed that smoothing merely reduces transaction costs (Section 4.2).

12. Showed that increasing smoothness decreases transaction costs by reducing the frequency of trading. (Section 4.2).

13. Showed that linearly extrapolating a filter does not increase performance (Section 4.2.2).

14. Showed that while smoothing offers no advantage, there might be another factor that does. This factor is possibly lag (Section 4.3.2).

\section*{5.2 Future Research}

The work in this thesis has raised some fascinating questions. These questions make up the recommended avenues for future research and are set apart in three areas.

The \textbf{M-Matrix}, in (2.28) on page 71, is a good example of the ingenuity of mathematics; simple yet elusive. There are three things to be explored with this matrix. (1) When the $d$ parameter is low, the matrix is easily invertible. However, as $d$ gets bigger, the matrix becomes unstable. There may be a way to take advantage of the pattern in the Toeplitz structure of the matrix to calculate a more accurate inverse. (2) If you take a stable M-Matrix, invert it and sum together the columns or rows a smooth series emerges. Interestingly, the series resembles a Gaussian curve. I expect a function exists that can generate this curve given the parameters to the M-Matrix; avoiding the matrix and inverse altogether. (3) The Linear Gaussian Smoother requires all elements in the inverted M-Matrix to be positive. Such a matrix is called an inverse positive matrix. A simple experiment was used in this work to show that
all the different parameters actually used did in fact produce an inverse positive matrix. A purely mathematical proof ought to be pursued.

Filter lag may play an important role in smoothing profitability. Chapter 4 only examined the effect that smoothness has on investment performance, however, Figure 4.21 shows some strange behaviour. None of the filters in this figure separate lag and smoothness. When one decreases or increases, so does the other. The most efficient smoothing filter in Figure 4.21 is the LGS which shows no benefit to smoothing or lag. However, the less efficient filters, for example the SMA, do show a trend when smoothing. The experiments in Chapter 4 showed that smoothness does not have an effect. Thus, another factor must be at play here. Since lag is the next most important factor, I suggest that it should be more thoroughly investigated.

Complex trading strategies using smoothing were not tested in this thesis. Chapter 4 looked at the base filter strategy. Other people modify this system with thresholding [129], neural networks [210], support vector machines [282], or parameter optimisation algorithms such as particle swarm optimisation [150]. Adding a linear predictive layer can be ruled out as the filtering is itself linear. Using parameter optimisation can also be ruled out as the Sharpe surface in Figures 4.6 & 4.7 are “spiky.” Such optimisation will only return a spike. Future research ought to examine how smoothness affects the popular thresholding rule and non-linear predictive systems.
Bibliography


[8] ALEXANDER, S. S. Price movements in speculative markets: Trends or random walks, number 2. Industrial Management Re-


[63] Chen, A.-S., Leung, M. T., and Daouk, H. Application of neural networks to an emerging financial market: forecasting and


[205] Livermore, J. L. How to trade in stocks. Duel, Sloan & Pearce, New York, USA, 1940.


Medical and Biological Engineering and Computing 37, 1 (1999), 99–103.


