Financial Model Estimation and Portfolio Rebalancing

by

Jiayin Kang

A thesis submitted in partial fulfillment of the requirements for the degree

of

Master of Science

[in

Mathematical Finance]

Department of Mathematical and Statistical Science

University of Alberta

Abstract

In this thesis we organize the contents in three parts. The first part is about portfolio rebalancing with changing benchmarks and the second part is about modeling of fractional Brownian motion in financial market while the last part is the conclusion.

In the first part, we introduce backgrounds in portfolio rebalancing and the rational why rebalancing is beneficial for a multi asset class portfolio. Then we describe four commonly used portfolio rebalancing methods and report other related comparisons. Then we introduce the proposed new portfolio rebalancing method and provide the back-testing results comparing with other methods using market data from June 2000 to July 2014 for a hypothetical multi-client institutional fund.

In the second part we introduce the properties and results of the mixed Brownian and fractional Brownian process with Hurst parameter H: $3/4 < H < 1$. Then we estimated Hurst parameter H for the Equity, Fixed Income, and Forex markets across all the countries to get an overall picture of the financial markets all over the world.

Preface

Research of the thesis in Chapters 1-3 has been performed in collaboration with Dr. Melnikov from University of Alberta, and Mr. Jerry (Bofu) Yang from Alberta Investment Management Corporation and research in Chapters 4-7 has been performed in collaboration with Dr. Melnikov. Experimental data in Chapter 3 was collected by Mr. Jerry (Bofu) Yang. Chapters 1-3 will be submitted for publication. My work includes summarizing results in Chapters 1-2 and 4-5, development of the new rebalancing method in Chapter 3, and empirical analysis in Chapter 3 and 6.

Acknowledgments

First of all, I would like to express my deep gratitude to my supervisor, Dr. Alexander Melnikov for his support and guidance in my research. His helpful suggestions and advice have helped me enormously in writing my thesis. It is always a great pleasure to work under his supervision.

I would also like to thank my co-supervisor, Mr. Jerry (Bofu) Yang, CFA, FRM, for his supports in this research. His industrial experience and knowledge helped my research a lot.

Lastly, I would like to thank all the students and professors who made an impact in my graduate career at the University of Alberta. All the lasting memories will be cherished forever.

Table of Contents

Part I

1. Introduction to Portfolio Rebalancing 1

1.1 Traditional Portfolio Rebalancing 1

1.2 Reason of Rebalancing 2

2. Main Approaching Methods and Comparison 6

2.1 Calendar Method 6

2.2 Traditional Two Bands Method 8

2.3 Dynamic Programming Method 14

2.4 Masters Method 19

2.5 Comparison among Traditional Methods 22

3. New Portfolio Rebalancing Method 26

3.1 New Idea in Portfolio Rebalancing 26

3.2 New Rebalancing Method 29

3.3 Back-Testing 33

Part II

4. Modeling with Fractional Brownian Motion 42

4.1 Tradition fractional Brownian motion 43

4.2 Mixed Brownian and Fractional Brownian Model 44

5. Why We Need H > 3/4? 45

6. Estimate H and Empirical Results 48

6.1 Estimate Hurst Parameter H 48

6.2 Empirical Results for World Equity Market 49

6.3 Empirical Results for World Forex Market 55

6.4 Empirical Results for World Fixed Income Market 59

Part III

7. Conclusion 65

7.1 Conclusion of Part I 65

7.2 Conclusion of Part II 65

Bibliography 67

Appendix 1. Matlab Code and Output for Chapter 3 70

Appendix 2. Matlab Code and Output for Chapter 6 85

List of Tables

1.1.1 Constant mix portfolio rebalancing results figure, taken from [Andre F. Perold and William F. Sharpe (1988)[2]] 3

1.1.2 To compare several basic rebalancing methods with the buy and hold strategy in several measures, taken from [Christopher Donohue and Kenneth Yip (2003)[3]] . 4

2.1.1 Returns by annual rebalancing vs returns by monthly rebalancing figure, taken from [Martin Leibowitz and Anthony Bova (2011)[4]] 7

2.1.2 Advantage vs annual rebalancing by monthly rebalancing and quarterly rebalancing, taken from [Martin Leibowitz and Anthony Bova (2011)[4]] 7

2.2.1 Three tables for the comparison of Tradition Two Bands method with other simple rebalancing methods figure, taken from [Christopher Donohue and Kenneth Yip (2003)[3]] 12

2.3.1 Two tables for comparison of Dynamic Programming method with other simple rebalancing methods figure, taken from [Sun, Fan, Chen, Schouwenaars, and Albota (2004)[6]] 16

2.3.2 Comparison of the improved Dynamic Programming method with old Dynamic Programming method and other simple rebalancing methods figure, taken from [Kritzman, Myrgren, and Page (2009)[7]] 19

2.5.1 List of portfolio rebalancing methods table, taken from [Ian Carmichael (2009)[9]] 22

3.1.1 Asset allocation trades for four investors table, taken from [William F. Sharpe

(2009)[1]] 27

3.3.1 Five tables for the portfolio holding and benchmark information of five clients of the fund 34

3.3.2 Returns for fixed band with fully rebalancing from 1% to 15%, generated by Matlab 36

3.3.3 Returns for fixed band with half rebalancing from 1% to 15%, generated by Matlab 37

3.3.4 Returns for fixed band with marginal rebalancing from 1% to 15%, generated by Matlab 38

3.3.5 Returns for asymmetrical band with fully rebalancing from 1%, 2% to 6%, 7%, generated by Matlab 38

3.3.6 Returns for asymmetrical band with half rebalancing from 1%, 2% to 6%, 7%, generated by Matlab 39

3.3.7 Returns for asymmetrical band with marginal rebalancing from 1%, 2% to 6%, 7%, generated by Matlab 40

3.3.8 Returns for calendar method with monthly to annually rebalancing, generated by Matlab 41

3.3.9 Returns for Masters method, generated by Matlab 41

3.3.10 Returns for our new portfolio rebalancing method, generated by Matlab 41

6.2.1 Countries with estimated H of stock index > 0.75, generated by Matlab 49

6.2.2 Countries with estimated H of stock index close to 0.75, generated by Matlab 49

6.2.3 Estimated H of stock indexes all over the world, generated by Matlab 50

6.2.4 Countries with estimated H of stock index close to 0.5, generated by Matlab 53

6.2.5 Estimated H of stock index in different regions, generated by Matlab 54

6.2.6 Estimated H of stock index for developed region and developing region, generated by Matlab 55

6.3.1 Countries for estimated H of foreign exchange all over the world, generated by Matlab 56

6.3.2 Countries with estimated H of foreign exchange close to 0.5, generated by Matlab 57

6.3.3 Estimated H of foreign exchange in different regions, generated by Matlab 58

6.4.1 Countries with estimated H of bond index > 0.75, generated by Matlab 59

6.4.2 Countries with estimated H of bond index close to 0.75, generated by Matlab 59

6.4.3 Countries with estimated H of bond index > 0.75 in all 3 estimation methods, generated by Matlab 60

6.4.4 Countries for estimated H of bond indexes all over the world, generated by Matlab 60

6.4.5 Countries with estimated H of bond index close to 0.5, generated by Matlab .. 63

6.4.6 Estimated H of bond index in different categories, generated by Matlab 63

List of Figures

2.5.2 Annual transaction costs vs annual tracking error for all tested rebalancing methods in all market, taken from [Ian Carmichael (2009)[9]] 23

2.5.3 Annual transaction costs vs annual tracking error for all tested rebalancing methods in bull market, taken from [Ian Carmichael (2009)[9]] 24

2.5.4 Annual transaction costs vs annual tracking error for all tested rebalancing methods in bear market, taken from [Ian Carmichael (2009)[9]] 24

Part I

Chapter 1

Introduction to Portfolio Rebalancing

Portfolio rebalancing plays an important role in the overall portfolio management. The traditional portfolio rebalancing target for each asset class is stated as a percent of the total value of the fund. Every time when the asset proportions drifted away from its benchmark weight, we rebalance them back. To accommodate disparities, most asset allocation policies include acceptable ranges.

1.1 Traditional Portfolio Rebalancing

At the beginning part of William F. Sharpe (2009)[1], it briefly introduces the traditional portfolio rebalancing method. Specifically, the traditional portfolio rebalancing method is to let dollar amounts invested in assets be X_1, X_2, ... X_n, then the initial value of the portfolio and proportion of every assets in the portfolio are:

$$V_0 = \sum_{i=1}^{n} X_i$$

$$\left(X_1/V_0, X_2/V_0, \ldots\ldots, X_n/V_0 \right)$$

Now a period has passed and we write the value-relative for asset i (the ratio of ending value to beginning value) as k_i, then new dollar value of each asset, ending value of the portfolio, and the proportion of every assets in the portfolio are:

$$\left(k_1X_1, k_2X_2, \ldots\ldots, k_nX_n\right)$$

$$V_1 = \sum_{i=1}^{n} k_i X_i$$

$$\left({k_1X_1}/{V_1}, {k_2X_2}/{V_1}, \ldots\ldots, {k_nX_n}/{V_1} \right)$$

We denote the value-relative change for portfolio as K_p, then:

$$K_p \equiv {V_1}/{V_0}$$

We let D_1, D_2, ... D_n denote dollar amounts of asset purchase (if positive) or sold (if negative) in order to fully rebalance. Then we should have the formulas:

$$k_iX_i + {D_i}/{V_1} = {X_i}/{V_0}$$

So that: $D_i = \left(K_p - k_i\right) X_i$

In this way, it keeps the asset proportions remain the same.

This is known as the traditional portfolio rebalancing.

1.2 Reason of Rebalancing

In Andre F. Perold and William F. Sharpe (1988)[2], it calls the traditional portfolio rebalancing method as the “constant mix” method. In this paper, it compares the “constant mix” with the normal buy and hold strategy, which explains why we do portfolio rebalancing and how to be benefit from the traditional portfolio rebalancing method.

We can see from an example in Andre F. Perold and William F. Sharpe (1988)[2] as below (it assumes the transaction cost is zero here):

Table 1.1.1

Case	Stock Market	Value of Stock	Value of Bills	Value of Assets	Percentage in Stocks
Initial	100	60.00	40.00	100.00	60.0%
After change	90	54.00	40.00	94.00	57.4
After rebalancing	90	56.40	37.60	94.00	60.0
After change	100	62.67	37.60	100.27	62.5
After rebalancing	100	60.16	40.11	100.27	60.0

It assumes the portfolio is a 60/40 constant mix in stock and bond, where the portfolio keeps 60% of its value invested in the stock market and 40% of its value invested in the bond market. In another ward, by assuming the total value of the portfolio is 100, the investor puts 60 in stocks and 40 in bonds.

Now one time has passed and when the stock market value drops from 100 to 90, the value of stock drops to 54 while the value of bond is still 40 in the portfolio. The total value of the portfolio drops to 94 now. In order to keep 60% in stock and 40% in bond, the fund manager should sell 2.4 units of bonds to buy 2.4 units of stocks. As a result, the portfolio values become 56.4 in stock and 37.6 in bond.

Now another period has passed and the stock market then rises from 90 back to 100. As a result, the value of stock rises from 56.4 to 62.67 while the value of bond is still 37.6 in the portfolio. But as we can see from the table 1.1.1 above, the value of portfolio rises to 100.27 while now the stock market and bond market are both 100 as the beginning. Therefore, we can see the "constant mix" method (which is the traditional portfolio rebalancing method) makes 0.27 gain without transaction costs.

Consequently, in Andre F. Perold and William F. Sharpe (1988)[2] it points out that the "constant mix" method gains from reversals. Specifically, the "constant mix" method tends to be superior if markets are characterized more by reversals than by trends while buy-and-hold policy tends to be superior if there is a major move in one direction. Therefore, rebalancing adds values to the return of portfolio when there are more reversals.

What is more, at the beginning part of Christopher Donohue and Kenneth Yip (2003)[3], it explains another reason why we do rebalancing. By comparing several basic rebalancing methods with the buy and hold strategy, Christopher Donohue and Kenneth Yip (2003)[3] shows as in the table 1.1.2 below:

Table 1.1.2

	Buy and Hold	Rebalance Quarterly	Rebalance Annually	Rebalance Outside 5% Range
Annual Turnover (%)	0.000%	6.559%	2.846%	3.032%
Annual Transaction Cost (%)	0.000%	0.045%	0.020%	0.020%
Annual Excess Return (less Transaction Costs)	8.80%	8.58%	8.47%	8.69%
Risk	10.26%	9.64%	9.66%	9.65%
Sharpe Ratio	0.8576	0.8899	0.8768	0.901

We can see that although never rebalancing (Buy and Hold strategy) outperforms all the other strategies in return, it has, for example, the lowest Shape Ratio.

Therefore, in terms of maximizing Sharpe Ratio (or minimize total risk), rebalancing increase a portfolio's Sharpe ratio by reducing volatility or increasing returns.

Chapter 2

Main Approaching Methods and Comparison

Here we introduce the commonly used portfolio rebalancing methods in the financial market and provide a comparison among those methods.

2.1 Calendar Method

The calendar method is the most straightforward rebalancing method. This method simply rebalances the asset ratios fully back to its benchmark ratios at certain time period such as monthly, quarterly, or annually. In Martin Leibowitz and Anthony Bova (2011)[4], it compares the calendar methods based on annually, quarterly, and monthly time benchmarks.

In Martin Leibowitz and Anthony Bova (2011)[4], it did an empirical analysis based on historical study for a 60/40 portfolio (60% in equity and 40% in bond) over 1960 - 2009 period under the zero transaction costs for monthly calendar rebalancing vs annually calendar rebalancing, and quarterly calendar rebalancing vs annually calendar rebalancing.

Summaries provided in Martin Leibowitz and Anthony Bova (2011)[4] are as follows:

Table 2.1.1

Monthly Rebalancing Advantage vs. Equity Return Buckets: 1960-2009

Market Return Range	AR	MR	MR Adv	# of Observations
< -20%	-15.6%	-16.5%	-0.9%	3
-20% to -10%	-5.3%	-5.4%	-0.1%	3
-10% to 0%	-1.3%	-1.3%	0.0%	6
0% to 10%	4.9%	5.2%	0.3%	9
10% to 20%	10.8%	10.9%	0.0%	11
> 20%	19.0%	18.8%	-0.2%	18
Overall Average	8.7%	8.6%	-0.1%	50

Table 2.1.2

Beta-Based and Quarterly Rebalancing Strategies

	Advantage vs AR		
Market Return Range	MR	QR	Beta-Based (0.55-0.65)
< -20%	-1.2%	-0.6%	-0.5%
-20% to -10%	0.0%	0.0%	0.2%
-10% to 0%	0.1%	0.1%	0.2%
0% to 10%	0.3%	0.3%	0.3%
10% to 20%	0.0%	0.0%	0.0%
> 20%	-0.2%	-0.1%	0.0%
Overall Average	-0.1%	0.0%	0.0%

We can see that from table 2.1.1, with zero transaction cost, the annual rebalancing performed better than the monthly rebalancing during 1960-2009 period. Since monthly rebalancing trades much more frequently than the annual rebalancing, on a net-cost basis the annual rebalancing should perform much better than the monthly rebalancing.

Similarly in table 2.1.2, by calculating advantage vs annual rebalancing under the zero transaction cost, we can see that monthly rebalancing has an average advantage -0.1% while that of quarterly rebalancing is 0.0%. Since monthly rebalancing and quarterly rebalancing both trade much more frequently than the annual rebalancing, the annual rebalancing should outperform significantly both monthly rebalancing and quarterly rebalancing.

Therefore, in Martin Leibowitz and Anthony Bova (2011)[4], it suggests that annual rebalancing outperforms both monthly rebalancing and quarterly rebalancing.

2.2 Traditional Two Bands Method

This method is first introduced in Hayne E. Leland (1996)[5]. It assumes transaction costs are proportional to the dollar amounts traded. It defines S - "stocks", B - "bonds", and assumes they both follow log random walks:

$$dS(t)/S = \mu_S dt + \sigma_S dZ_S(t)$$

$$dB(t)/B = \mu_B dt + \sigma_B dZ_B(t)$$

Here dZ_S and dZ_B are the increments of Wiener processes with correlation ρ.

Then the stochastic process w(t) = S(t)/B(t) is:

$$dw(t)/w = (\mu_S - \mu_B + \sigma_B^2 - \rho\sigma_B\sigma_S)dt + \sigma_S dZ_S(t) - \sigma_B dZ_B(t)$$

We assume the target ratio is w*, then the dollar equivalent cost over a time interval dt may be approximated by:

$$L = \lambda(w(t) - w^*)^2 dt$$

where λ is a constant representing the cost per unit of tracking error

Let k_S and k_B be transactions cost per dollar of stocks and bonds traded and let δC, δS and δB be total costs, change in dollar value of stocks, and change in dollar value of bonds:

$$\delta B = -\delta S$$

$$\delta C = k_S \left|\delta S\right| + k_B \left|\delta B\right| = (k_S + k_B)\left|\delta S\right|$$

Since $\delta w = (1+w)\delta S / B$ and W = B + S and $\delta c = \delta C / W$, we have:

$$\delta C / \left|\delta w\right| = B(k_S + k_B) / (1+w)$$

$$\delta c / \left|\delta w\right| = (k_S + k_B) / (1+w)^2$$

Cost Function

Let V(w(t); w_{min}, w_{max}) be the cost function associated a trading strategy characterized by no trading whenever $w(\tau) + \delta w(\tau) \in [w_{\min}, w_{\max}]$:

$$V(w(t); w_{min}\ w_{max}) = E[\int_{\tau=t}^{\infty} e^{-r(\tau-t)} \lambda (w(\tau)_{-} - w^*)^2 d\tau + PV\{transaction\cos ts\}]$$

(1) When $w(\tau) + \delta w(\tau) \in [w_{\min}, w_{\max}]$:

$$V(w(t); w_{min}\ w_{max}) = -\int_{\tau=t}^{t+dt} \lambda (w(t) - w^*)^2 d\tau + E[e^{-rdt} V(w(t) + dw(t); w_{min}\ w_{max})]$$

(2) When $w(\tau) + \delta w(\tau) < w_{\min}$ or $w(\tau) + \delta w(\tau) > w_{\max}$:

$$V(w(t); w_{min}\ w_{max}) = V(w_{min}; w_{min}\ w_{max}) + \delta c^{*} / |\delta w| (w_{min} - w(t\ )$$

$$V(w(t); w_{min}\ w_{max}) = V(w_{\max}\ w_{min}\ w_{max}) + \delta c^{**} / |\delta w| (w\ t) - w_{\max})$$

Therefore, we got the boundary conditions:

$$V(w_{min}; w_{min}\ w_{max}) = -\delta c^{*} / |\delta w| = -(k_S + k_B)(1 + w_{min})^2$$

$$V(w_{\max}; w_{min}\ w_{max}) = \delta c^{**} / |\delta w| = (k_S + k_B)(1 + w_{\max})^2$$

Here $V_k(*; *, *)$ is the derivative of V with respect to its k^{th} argument.

By expanding the expectation term, we have:

$$awV_1(w; w_{min}\ w_{max}) + 0.5bw^2 V_{11}(w\ w_{min}\ w_{max}) + \lambda(w - w^{*})^2 - rV(w\ w_{min}, w_{max}) = 0$$

where $a = \mu_S - \mu_B + \sigma_B^2 - \rho\sigma_B\sigma_S$, $b = \sigma_S^2 + \sigma_B^2 - 2\rho\sigma_B\sigma_S$.

The solution of above differential equation is:

$$V(w; w_{min}\ w_{max}) = \lambda\left[\frac{w^2}{r - 2a - b} - \frac{2ww^{*}}{r - a} + \frac{w^{*2}}{r}\right] + C_1 w^x + C_2 w^y$$

where x = (-2a + b - sqrt{$(2a-b)^2$ + 8br})/2b, y = (-2a + b + sqrt{$(2a-b)^2$ + 8br})/2b.

Here C_1 and C_2 are determined by the boundary conditions and w_{min} & w_{max}.

Therefore, by calculating the rebalancing bands base on the differential equations above, we can get "no trade" interval [w_{min}^{*}, w_{max}^{*}] where no trades are required when ratio of asset values moves within the interval and asset proportions should be adjusted back to the nearest edge of the interval when asset ratio moves outside the no-trade interval.

In Christopher Donohue and Kenneth Yip (2003)[3], it improves above method in approximating the edges of the no-trade region.

Since the no-trade region represents the area surrounding the target ratios, the marginal decline in expected future transaction and tracking error costs is less than or equal to marginal increase in current transaction costs. They let $J(\omega,\beta)$ denote the present value of expected future transaction and tracking error costs as the proportion of wealth currently in risky asset w and rebalancing strategy β. Then the objective is to find strategy β to minimize $J(\omega,\beta)$:

(1) To accept the current level of tracking error, the cost is $J(\omega,\beta)$.

(2) To reduce w to ϖ by reallocating funds from risky asset to risk-free asset, the cost is $\kappa(\omega-\varpi)+J(\varpi,\beta)$.

Therefore, (1) is the optimal action if and only if:

$$J(\omega,\beta)\le\kappa(\omega-\varpi)+J(\varpi,\beta)$$

Further, if the optimal action is to trade (option (2)), the trade should move w to ϖ such that:

$$J(\omega,\beta)-J(\varpi,\beta)=\kappa(\omega-\varpi)$$

Or equivalently:

$$J_{\omega}(\omega, \beta) = \kappa$$

Here $J_{\omega}(\omega, \beta)$ is the partial derivative of $J(\omega, \beta)$ with respect to w.

In Christopher Donohue and Kenneth Yip (2003)[3], it compares optimal rebalancing with Periodic (equal tracking error, daily, weekly, monthly, quarterly, semiannually, annually), Volatility (1%, 2%, 3%, 4%, 5%, 10%, 15%, buy-and-hold), Equal Probability (x = 5%, 20%, 35%, 50%), and Active Risk (thresholds of 0.01%, 0.03%, 0.05%, 0.1% from target-weight portfolio) respectively. The output is listed in tables 2.2.1 below:

Tables 2.2.1

Rebalancing Strategy	Number of Rebalances Per Year	Annual Turnover	Annual Tracking Error	Annual Transaction Cost	Expected Utility	Rank	Annual Return	Annual Return Error	Std. Deviation of Annual Return	Sharpe Ratio	Rank
Optimal	33.65	5.99	0.0054	0.0621	-0.3121	1	12.953	0.0130	8.652	1.497	1
Periodic - Equal Tracking Error	2.00	11.54	0.0054	0.1197	-0.3152	9	13.025	0.0130	8.914	1.161	10
Daily	252.00	126.90	0.0005	1.1782	-0.3534	23	11.621	0.0116	8.546	1.360	22
Weekly	560.40	56.91	0.0011	0.5762	-0.3294	21	12.412	0.0124	8.615	1.441	16
Monthly	12.00	27.99	0.0022	0.2855	-0.3199	18	12.759	0.0128	8.676	1.471	9
Quarterly	4.00	16.35	0.0039	0.1690	-0.3163	14	12.929	0.1290	8.788	1.471	8
Semi-Annually	2.00	11.68	0.0055	0.1232	0.3154	11	13.030	0.0130	8.955	1.455	13
Annually	1.00	8.51	0.0081	0.0936	0.3153	10	13.181	0.3120	9.287	1.419	18
1% Range	24.44	44.42	0.1100	0.4503	-0.3254	19	12.550	0.0126	8.264	1.455	12
2% Range	6.56	23.14	0.0020	0.2355	-0.3183	16	12.809	0.0128	8.670	1.477	4
3% Range	3.02	15.65	0.0030	0.1605	-0.3161	13	12.908	0.0129	8.722	1.480	2
4% Range	1.75	11.94	0.0040	0.1231	-0.3149	8	12.976	0.0130	8.776	1.479	3
5% Range	1.15	9.71	0.0049	0.1013	-0.3144	6	13.030	0.0130	8.840	1.474	7
10% Range	0.34	5.33	0.0094	0.0564	-0.3142	5	13.294	0.0133	9.223	1.441	15
15% Range	0.17	3.83	0.0135	0.0458	-0.3164	15	13.565	0.0136	9.630	1.409	20
Busy and Hold	0.00	0.00	0.0354	0.0000	-0.3820	24	14.960	0.0150	12.456	1.201	24
Equal Probability (5%)	33.42	51.74	0.0009	0.5239	-0.3277	20	12.469	0.0125	8.618	1.447	14
Equal Probability (20%)	2.37	13.82	0.0035	0.1421	-0.3156	12	12.943	0.0129	8.780	4.474	6
Equal Probability (35%)	0.84	8.36	0.0060	0.0883	-0.3139	4	13.110	0.0131	8.989	1.458	11
Equal Probability (50%)	0.45	6.22	0.0085	0.0682	-0.3135	3	13.266	0.0133	9.208	1.441	17
Active Risk (0.10%)	28.85	12.07	0.0039	0.1246	-0.3144	7	12.999	0.0130	8.813	1.475	5
Active Risk (0.30%)	3.84	4.51	0.0112	0.0511	-0.3130	2	13.492	0.0135	9.516	1.418	19
Active Risk (0.50%)	1.58	2.95	0.0173	0.0382	-0.3195	17	13.906	0.0139	10.199	1.364	21
Active Risk (1.0%)	0.37	1.34	0.2800	0.0241	-0.3481	22	14.561	0.1460	11.450	1.268	23

Asset Parameters

$\mu_1 = 11.49$
$\mu_2 = 14.39$
$\mu_3 = 23.21$
$r_f = 5.00$

$$v = \begin{bmatrix} 0.0205 & 0.0062 & 0.0090 \\ 0.0062 & 0.0184 & 0.0085 \\ 0.0090 & 0.0085 & 0.0383 \end{bmatrix}$$

$\omega_1 = 13.55\%$
$\omega_2 = 26.61\%$
$\omega_3 = 26.86\%$
$\omega_4 = 32.98\%$

$(k^+_1, k^-_1) = (0.01, 0.01)$
$(k^+_2, k^-_2) = (0.01, 0.01)$
$(k^+_3, k^-_3) = (0.01, 0.01)$
$(k^+_f, k^-_f) = (0.00, 0.00)$

Rebalancing Strategy	Number of Rebalances Per Year	Annual Turnover	Annual Tracking Error	Annual Transaction Cost	Expected Utility	Rank	Annual Return	Annual Return Error	Std. Deviation of Annual Return	Sharpe Ratio	Rank
Optimal	60.78	13.76	0.1780	0.1470	-0.4373	1	11.149	0.0111	16.480	0.677	1
Periodic - Equal Tracking Error	2.80	27.72	0.0178	0.2901	-0.4401	6	11.183	0.0112	16.787	0.666	6
Daily	252.00	266.34	0.0020	2.7329	-0.5497	23	8.493	0.0085	16.079	0.528	24
Weekly	50.40	119.28	0.0043	1.2354	-0.4747	21	10.110	0.0101	16.334	0.619	21
Monthly	12.00	58.39	0.0087	0.6098	-0.4471	12	10.821	0.0108	16.499	0.656	13
Quarterly	4.00	33.84	0.0151	0.3579	-0.4393	4	11.141	0.0111	16.697	0.667	4
Semi-Annually	2.00	23.83	0.0214	0.2570	-0.4413	9	11.271	0.0113	16.974	0.664	9
Annually	1.00	16.89	0.0304	0.1893	-0.4493	14	11.465	0.0115	17.552	0.653	16
1% Range	81.55	167.40	0.0025	1.7271	-0.4977	22	9.578	0.0096	16.243	0.590	22
2% Range	22.35	87.76	0.0047	0.9112	-0.4601	16	10.456	0.0105	16.393	0.638	18
3% Range	10.26	59.48	0.0068	0.6207	-0.4478	13	10.777	0.0108	16.479	0.654	15
4% Range	5.86	44.89	0.0090	0.4694	-0.4421	10	10.946	0.0109	16.533	0.662	11
5% Range	3.77	35.93	0.0112	0.3764	-0.4396	5	11.044	0.0110	16.602	0.665	8
10% Range	0.97	18.08	0.0219	0.1964	-0.4402	8	11.324	0.0113	16.954	0.668	2
15% Range	0.43	11.81	0.0321	0.1241	-0.4523	15	11.449	0.0114	17.345	0.660	12
Busy and Hold	0.00	0.00	0.0934	0.0000	-0.6367	24	12.498	0.0125	22.156	0.564	23
Equal Probability (5%)	30.76	102.79	0.0041	1.0655	-0.4666	19	10.293	0.0103	16.357	0.629	20
Equal Probability (20%)	2.09	26.61	0.0154	0.2820	-0.4384	3	11.170	0.1120	16.763	0.666	5
Equal Probability (35%)	0.69	15.20	0.0263	0.1683	-0.4448	11	11.381	0.0114	17.152	0.664	10
Equal Probability (50%)	0.34	10.42	0.0366	0.1222	-0.4603	17	11.528	0.0115	17.607	0.655	14
Active Risk (0.10%)	305.65	88.58	0.0046	0.9196	-0.4604	18	10.451	0.0105	16.401	0.637	19
Active Risk (0.30%)	36.92	29.06	0.0134	0.3072	-0.4373	2	11.174	0.0112	16.741	0.667	3
Active Risk (0.50%)	13.18	16.99	0.0222	0.1840	-0.4401	7	11.355	0.0114	17.060	0.666	7
Active Risk (1.0%)	3.08	7.76	0.0426	0.0937	-0.4689	20	11.709	0.0117	18.119	0.646	17

Asset Parameters

$m_1 = 19.00$, $m_2 = 14.00$, $m_3 = 12.00$, $r_f = 2.00$

$$v = \begin{bmatrix} 0.1600 & 0.0000 & 0.0000 \\ 0.0000 & 0.1000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0900 \end{bmatrix}$$

$w_1 = 23.61\%$, $w_2 = 26.67\%$, $w_3 = 24.69\%$, $w_f = 25.03\%$

$(k'_1, k_1) = (0.01, 0.01)$, $(k'_2, k_2) = (0.01, 0.01)$, $(k'_3, k_3) = (0.01, 0.01)$, $(k'_f, k_f) = (0.00, 0.00)$

Rebalancing Strategy	Number of Rebalances Per Year	Annual Turnover	Annual Tracking Error	Annual Transaction Cost	Expected Utility	Rank	Annual Return	Annual Return Error	Std. Deviation of Annual Return	Sharpe Ratio	Rank
Optimal	54.12	10.49	0.0088	0.1566	-0.2985	1	14.401	0.0144	12.042	1.196	1
Periodic - Equal Tracking Error	2.80	19.90	0.0088	0.3299	-0.3011	8	14.324	0.0143	12.160	1.178	5
Daily	252.00	188.42	0.0010	3.0446	-0.3868	22	11.185	0.0112	11.683	0.957	24
Weekly	50.40	84.42	0.0021	1.3777	-0.3270	18	13.084	0.0131	11.888	1.101	20
Monthly	12.00	41.36	0.0043	0.6810	-0.3060	12	13.901	0.0139	12.001	1.158	16
Quarterly	4.00	24.00	0.0074	0.3999	-0.3008	6	14.255	0.0143	12.114	1.177	6
Semi-Annually	2.00	16.96	0.0105	0.2881	-0.3031	9	14.409	0.0144	12.260	1.175	8
Annually	1.00	12.08	0.0150	0.2133	-0.3115	15	14.575	0.0146	12.529	1.163	14
1% Range	43.19	86.21	0.0017	1.3956	-0.3275	19	13.058	0.0131	11.882	1.099	21
2% Range	11.53	44.57	0.0032	0.7266	-0.3069	13	13.825	0.0138	11.982	1.154	17
3% Range	5.28	30.20	0.0047	0.4929	-0.3008	5	14.106	0.0141	12.044	1.171	11
4% Range	3.01	22.76	0.0063	0.3724	-0.2989	4	14.255	0.0143	12.098	1.178	4
5% Range	1.94	18.22	0.0078	0.2998	-0.2989	3	14.345	0.0143	12.158	1.180	2
10% Range	0.51	9.24	0.0152	0.1591	-0.3104	14	14.626	0.0146	12.476	1.172	10
15% Range	0.23	6.13	0.0216	0.1113	0.3314	21	14.833	0.0148	12.805	1.158	15
Busy and Hold	0.00	0.00	0.0497	0.0000	-0.5249	24	15.573	0.0156	14.905	1.045	23
Equal Probability (5%)	30.02	7195.00	0.0020	1.1726	-0.3203	17	13.314	0.0133	11.913	1.118	19
Equal Probability (20%)	2.06	18.75	0.0075	0.3101	-0.2989	2	14.332	0.0143	12.157	1.179	3
Equal Probability (35%)	0.70	10.84	0.0129	0.1848	-0.3054	11	14.558	0.0146	12.385	1.175	7
Equal Probability (50%)	0.35	7.58	0.0180	0.1340	0.3185	16	14.719	0.0147	12.631	1.165	13
Active Risk (0.10%)	79.63	30.93	0.0045	0.5047	-0.3010	7	14.096	0.0141	12.057	1.169	12
Active Risk (0.30%)	9.14	10.26	0.0132	0.1737	-0.3045	10	14.641	0.0146	12.480	1.173	9
Active Risk (0.50%)	3.22	5.85	0.0214	0.1051	-0.3298	20	14.877	0.0149	12.934	1.150	18
Active Risk (1.0%)	0.60	2.18	0.037554	0.0490	-0.420734	23	15.312	0.0153	13.982	1.095	22

Asset Parameters

$m_1 = 16.32$, $m_2 = 17.92$, $m_3 = 18.98$, $r_f = 6.00$

$$v = \begin{bmatrix} 0.0515 & 0.0000 & 0.0000 \\ 0.0000 & 0.0621 & 0.0000 \\ 0.0000 & 0.0000 & 0.0608 \end{bmatrix}$$

$w_1 = 23.74\%$, $w_2 = 22.99\%$, $w_3 = 27.85\%$, $w_c = 25.42\%$

$(k'_1, k_1) = (0.03, 0.03)$, $(k'_2, k_2) = (0.01, 0.01)$, $(k'_3, k_3) = (0.01, 0.01)$, $(k'_f, k_f) = (0.00, 0.00)$

We can see from tables 2.2.1, it concludes that strategies that use a no-trade region constructed as a function of individual asset transaction costs and investor risk aversion offer promise for finding the proper balance between minimizing transaction costs and tracking error.

2.3 Dynamic Programming Method

In Sun, Fan, Chen, Schouwenaars, and Albota (2004)[6], it first introduces the dynamic programming method. It presents an approach explicitly weighs transaction costs and portfolio tracking error.

In that paper, it defines w_t as state, u_t as policy, and n_t as state uncertainty. The state transition is:

$$w_{t+1} = h(w_t, u_t, n_t)$$

Therefore, the cost function is written recursively as:

$$J_t(w_t) = E[G(w_t, u_t, n_t) + J_{t+1}(w_{t+1})]$$

Here J_t is the so-called cost-to-go function, G is the cost for current period. Therefore, the cost at any given period is the expected cost from t to t+1 along with the expected cost from t+1 onwards.

At time t, the optimal strategy is to choose u_t such that the cost is minimized:

$$J_t^*(w_t) = \min_{u_t} E[G(w_t, u_t, n_t) + J_{t+1}(w_{t+1})]$$

Assuming convergence, this recursion approaches a fixed point such that:

$$J_t^*(w) = J_{t+1}^*(w) = J^*(w)$$

Therefore, we could use value iteration technique to determine the cost-to-go values. By randomly choosing an arbitrary set of cost-to-go values $J_t(w)$, we repeatedly apply above minimization function to obtain cost-to-go values successively. After a sufficient number of iterations, we will approach a steady-state, which should converge the optimal $J^*_t(w)$.

The decision to rebalance should be based on a consideration of 3 costs:

(1) Tracking error associated with any deviation in our portfolio from the optimal portfolio

(2) Trading costs associated with buying or selling any assets during rebalancing

(3) Expected future cost from next month onwards

Therefore, to apply dynamic programming, we write the cost function as:

$$E[G(w_t, u_t, n_t)] = \tau(u_t) + \varepsilon(w_t + u_t)$$

Here $\tau(u_t)$ is the associated trading cost, $\varepsilon(.)$ represents the sub-optimality cost. $\varepsilon(w_t + u_t) = 0$ when $w_t + u_t = w^*$.

For a portfolio w to go to another portfolio w', we assume for asset i we pay a transaction cost of c_i per dollar to buy or sell so that:

$$\tau(w', w) = c^T \left| w' - w \right|$$

For any portfolio weights w, we express the expected utility as $U(\mu^T w, w^T \Lambda w)$. Then there exists a risk-free rate $r_{CE}(w)$ where $U(r_{CE}, 0) = U(\mu^T w, w^T \Lambda w)$. We denote $r_{CE}(w)$ as certainty equivalent return for weights w.

Under the certainty equivalence approach, the tracking error has the cost function:

$$\varepsilon(w) = r_{CE}(w^*) - r_{CE}(w)$$

Therefore, by using above method, that paper compares it with other portfolio rebalancing methods as below:

Tables 2.3.1

	(a)	(b)	(c)	(d)
	Trading Cost (bps)	Suboptimality Cost (bps)	Aggregate Cost (bps)	Utility Shortfall (utils x 10^4)
Ideal	0.00	0.00	0.00	0.00
Optimal DP	1.57	0.46	2.03	2.15
No Trading	0.00	4.74	4.74	4.56
5% Tolerance	3.68	0.13	3.81	3.65
Monthly	12.92	0.00	12.92	12.94
Quarterly	7.46	0.05	7.51	7.46
Annual	3.71	0.26	3.97	4.05

		(a) Trading Cost (bps)	(b) Suboptimality Cost (bps)	(c) Aggregate Cost (bps)	(d) Utility Shortfall (utils x 10^4)
Quadratic $\alpha = 1.5$	Ideal	0.00	0.00	0.00	0.00
	Optimal DP	3.97	1.49	5.47	5.23
	No Trading	0.00	30.18	30.18	32.89
	5% Tolerance	7.29	0.70	7.99	7.88
	Monthly	23.67	0.00	23.67	23.73
	Quarterly	13.69	0.28	13.96	14.21
	Annual	6.84	1.55	8.39	8.43
Power	Ideal	0.00	0.00	0.00	0.00
	Optimal DP	3.39	1.15	4.54	4.32
	No Trading	0.00	25.99	25.99	26.04
	5% Tolerance	5.19	0.82	6.01	5.95
	Monthly	20.05	0.00	20.05	19.96
	Quarterly	11.59	0.18	11.78	11.87
	Annual	5.81	1.02	6.84	6.80
Log Wealth	Ideal	0.00	0.00	0.00	0.00
	Optimal DP	4.80	1.93	6.72	6.57
	No Trading	0.00	30.52	30.52	32.01
	5% Tolerance	11.95	0.43	12.38	12.72
	Monthly	28.15	0.00	28.15	28.19
	Quarterly	16.27	0.40	16.67	16.80
	Annual	8.05	2.17	10.22	10.65

According to tables 2.3.1, it concludes that this optimal dynamic programming method provides gains over the best of the traditional techniques of rebalancing.

In Kritzman, Myrgren, and Page (2009)[7], it is doing similar thing but it re-defines the cost function as:

$$J_t(X_t, X_{t-1}) = CEC_t + TC_t + J_{t+1}(X_{t+1}, X_t)$$

Here CEC_t is the Certainty Equivalent Cost and TC_t is the Transaction Costs. Then the current period $J_t(X_t, X_{t-1})$ is a function of the current CEC, TC, and future costs $J_{t+1}(X_{t+1}, X_t)$.

Specifically, it assumes investors have log-wealth utility and the expected utility E(U) is written as the weighted sum of the n security expected returns under m scenarios, each associated with p probability:

$$E(U)=\sum_{i=1}^{m} p_i In(1+\sum_{j=1}^{n} X_j \mu_{ij}) = pIn(1+\mu X')$$

Where: $\mu = \begin{pmatrix} \mu_{11} & \cdots & \cdots \\ \vdots & \ddots & \vdots \\ \mu_{m1} & \cdots & \cdots \end{pmatrix}$

By letting $X^{opt} = [X_1^{opt},...,X_n^{opt}]$ denote the optimal portfolio weights and E(U) is maximized when X = X^{opt} as E(U*), it quantifies the loss in expected utility as Certainty Equivalent Cost (CEC):

$$CEC = e^{E(U^*)} - e^{E(U)}$$

And it defines the Transaction Costs (TC) at period t are:

$$TC_t = \sum_{i=1}^{n} C_j \left| X_{jt} - X_{jt-1} \right|$$

Then in Kritzman, Myrgren, and Page (2009)[7] it improves the method in Sun, Fan, Chen, Schouwenaars, and Albota (2004)[6] by replacing $J_{t+1}(X_{t+1}, X_t)$ as above by the quadratic function below:

$$Q = \sum_{i=1}^{n} d(X_i - X^{opt})^2$$

As a result, the cost function becomes:

$$J_t(X_t, X_{t-1}) = CEC_t + TC_t + Q_t$$

In that paper, it also provides empirical results as below:

Table 2.3.2

Table 5. Performance Comparison – Total Costs (bps)

Rebalancing Strategy	Two Assets	Three Assets	Four Assets	Five Assets	Ten Assets	Twenty Five Assets	Fifty Assets	Hundred Assets
Dynamic Programming	6.31	6.66	7.33	8.76	NA	NA	NA	NA
MvD Heuristic	6.90	7.03	7.58	8.61	25.57	20.38	17.92	12.46
0.25% Bands	15.19	17.01	19.81	21.37	41.93	42.96	41.53	26.88
0.50% Bands	14.11	15.75	17.81	18.92	41.73	38.42	31.15	21.82
0.75% bands	12.80	14.09	15.32	16.27	40.05	32.95	31.46	25.02
1% Bands	11.54	12.52	13.15	14.13	37.71	31.95	36.74	29.47
2% Bands	8.73	9.20	9.79	10.73	41.94	48.59	66.96	39.33
3% Bands	8.51	8.66	10.14	11.43	61.29	73.78	89.03	41.54
4% Bands	9.46	9.52	12.08	13.78	88.49	93.23	98.55	41.96
5% Bands	11.20	11.21	14.80	16.77	120.19	106.38	102.38	42.03
Monthly	15.65	17.25	20.07	21.85	41.92	42.92	43.34	39.75
Quarterly	11.05	11.86	13.51	14.76	45.17	34.32	33.12	26.54
Semi-annually	11.13	11.53	12.67	13.95	69.97	40.75	37.33	24.41

We can see in table 2.3.2, it shows that by using this method it performs similar to the dynamic programming in Sun, Fan, Chen, Schouwenaars, and Albota (2004)[6] and performs substantially better than other heuristic methods.

2.4 Masters Method

In Seth J. Masters (2003)[8], it introduces a very intuitive way to calculate the rebalancing bands.

2.4.1 Trigger Bands

It lets K represent the investor's risk tolerance, and lets Delta be the deviation from target allocation, then the benefit of rebalancing is:

$$\frac{(Tracking\ Error)^2}{2K}\Delta^2$$

Here the tracking error is a measure of how closely a portfolio follows the benchmarks, where it could be defined as the standard deviation of the difference between the portfolio and the benchmark returns.

Let C be the total two-way cost of rebalancing, then the cost of rebalancing is:

$$C\Delta$$

To sum up, the net benefit of rebalancing is:

$$\frac{(Tracking\ Error)^2}{2K}\Delta^2 - C\Delta$$

Therefore, the trigger point T for asset class i should be:

$$\frac{(Tracking\ Error_i)^2}{2K}T_i^2 - C_iT_i = 0$$

We get:

$$T_i = \frac{2KC_i}{(Tracking\ Error_i)^2}$$

We can see two points about the triggering points:

(1) The higher risk tolerance K, the higher trigger point.

(2) The more expensive to trade an asset, the higher trigger point.

We let σ_i^2 be the volatility of the asset to be rebalanced; σ_j^2 be the volatility of the rest of the portfolio; and ρ_{ij} be the correlation between the asset and the rest of portfolio. Then the tracking error can be calculated by:

$$\left(Tracking\ Error_i\right)^2 = \sigma_i{}^2 + \sigma_j{}^2 - 2\rho_{ij}\sigma_i\sigma_j$$

As a result, the trigger point is:

$$T_i = \frac{2KC_i}{\left(\sigma_i{}^2 + \sigma_j{}^2 - 2\rho_{ij}\sigma_i\sigma_j\right)}$$

2.4.2 Level of Rebalancing

For the optimal rebalancing, marginal benefit should be equal to marginal cost.

Therefore, we should rebalance as far as the incremental move still yields a marginal net benefit and stop when the marginal benefit of rebalancing is equal to marginal cost:

$$\frac{\delta \text{Net Rebalancing Benefit}}{\delta\Delta} = 0$$

Then we get:

$$\Delta = \frac{KC}{\left(Tracking\ Error\right)^2}$$

This is exactly half of the trigger point T.

To sum up, for Masters method, we calculate the triggering points as the formula before, then every time we rebalance back to the midpoint between and triggering point and the benchmark.

2.5 Comparison among Traditional Methods

In order to determine the best portfolio rebalancing method, we need to find out the most optimal trade-off between tracking error and transaction costs. In Ian Carmichael (2009)[9], it compares most of the portfolio rebalancing methods in the financial market.

The methods it takes into consideration are:

Table 2.5.1

method name	method brief
Calendar	Portfolio is rebalanced at a pre-determined frequency. (e.g. annually)
Common in Practice - 'Complete'	All asset classes have identical no-trading intervals that are fixed and do not change with time. - To rebalance all assets back to target allocation
Common in Practice - 'Selective'	All asset classes have identical no-trading intervals that are fixed and do not change with time. - To only rebalance the asset class in breach of its rebalancing interval back to target allocaton
Asymmetrical	It has asymmetrical no-trading intervals around target allocation where upper bound significantly wider than lower bound
Leland (1996) (Traditional Two Bands)	Optimal no-trade intervals are dependent on 'relative asset class volatility', 'cost per unit of tracking error', and 'transaction cost'. It requires asset classes to be only rebalanced back to the boundary of the no-trade interval.

Masters (2003)	In contrast to Leland, Masters found it optimal to rebalance back only to half-way between the edge of the no-trading interval and the target allocation.
Dynamic programming	It uses dynamic programming to determine optimal rebalancing rules.

However, in order to be useful in the financial market, it should satisfy the criteria below:

(1) Must be unambiguous and simple to follow

(2) Must be practical to both individual and institutional investors (no infinite infinitesimal changes)

(3) Must work in a world with transaction fees

Therefore, based on the criteria above, only 'Calendar', 'Common - Complete', 'Common - Selective', 'Asymmetrical', and 'Masters' methods are satisfied.

In Ian Carmichael (2009)[9], it applies bootstrapping simulation method to above rebalancing methods. It assumes the transaction costs are 1% if VIX >= 35 and 0.2% if VIX < 35 and one week delay is built into the model. In this way, this paper summarizes the performance annual transaction cost (%) vs annual tracking error (%) in all market, in bull market, and in bear market.

The performance curves are shown in Ian Carmichael (2009)[9] as below:

Figure 2.5.2 - in all market

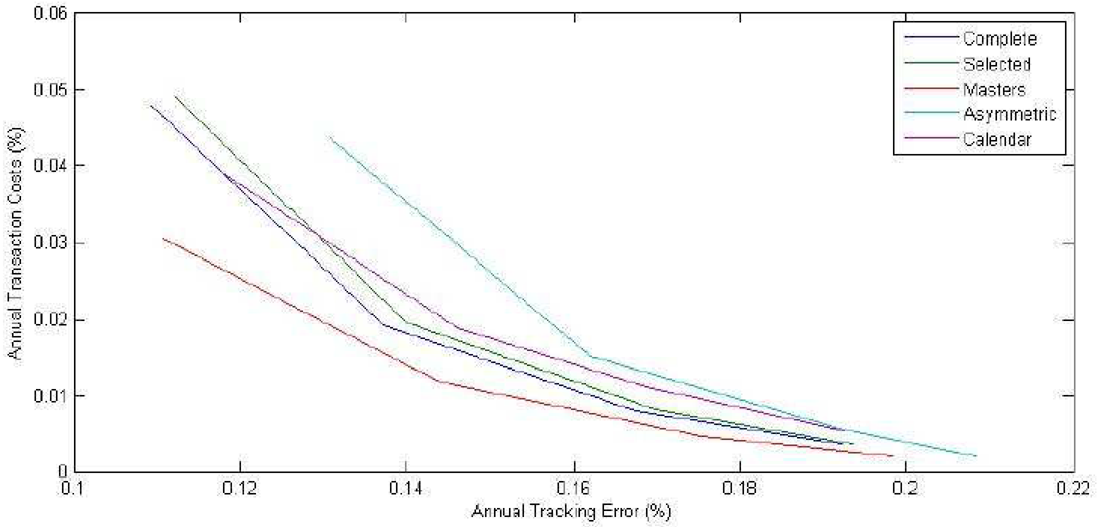


Figure 2.5.3 - in bull market

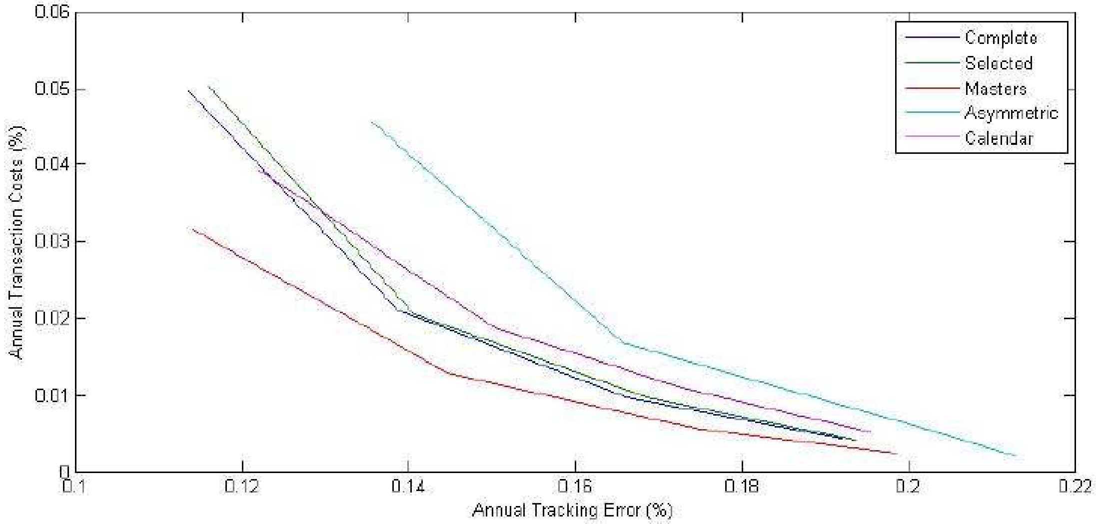
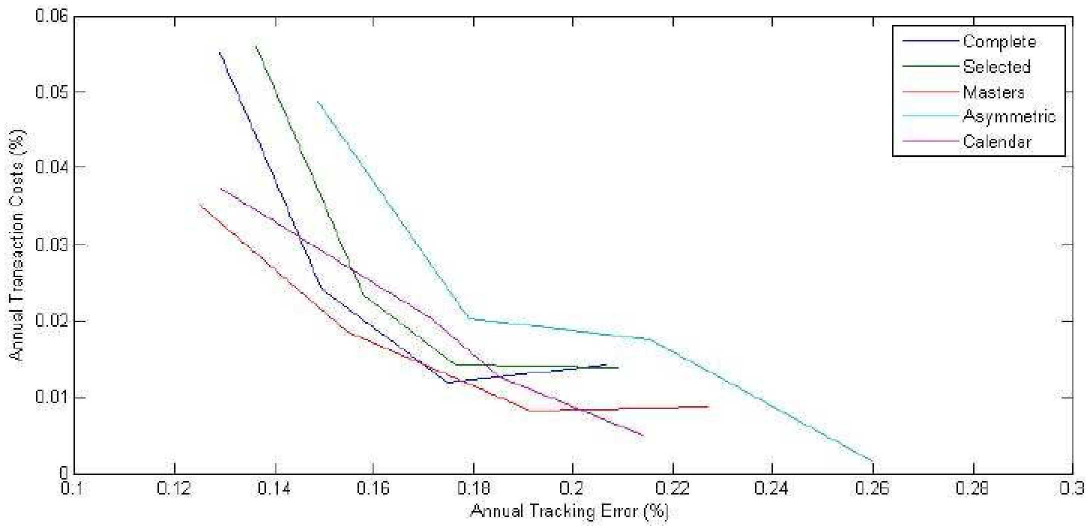


Figure 2.5.4 - in bear market



Therefore, according to figure 2.5.2, figure 2.5.3, and figure 2.5.4 from Ian Carmichael (2009)[9] above, we can see that no matter in which market environment, in bull market, in bear market, or dropping the transaction costs' dependence on market volatility, Masters method is found to provide the most optimal trade-off between tracking error and associated transaction costs where:

(1) It incurs a significant smaller transaction cost per rebalance

(2) It requires only a marginal increase in rebalancing frequency

Chapter 3

New Portfolio Rebalancing Method

In the previous chapters, we describe the traditional portfolio rebalancing methods. However, all previous methods are basing on fixed benchmarks. But when market moves dramatically, fixed benchmarks might not capture the market feature changes very well. Therefore, in this chapter, we are going to introduce a new portfolio rebalancing method with changing benchmarks.

3.1 New Idea in Portfolio Rebalancing

In the introduction part, we describe the traditional portfolio rebalancing. As we discussed before, buying stocks when they fall and selling stocks as they rise strategy capitalizes on reversals and normally provide lower total risk and higher Sharpe Ratio.

However, in William F. Sharpe (2009)[1], it points out that it is impossible for all the investors to follow the traditional portfolio rebalancing method. It illustrates this with a simple example. It assumes that there are totally four investors where they follow an asset allocation policy with positive proportions of four asset classes but proportions differ. Now a period has passed, and the assets have performed differently. In the table below the assets are numbered in terms of their performance ($k_1 > k_2 > k_3 > k_4$):

Table 3.1.1

Asset Allocation Trades for Four Investors

Assets in decreasing order of return	*Investor A*	*Investor B*	*Investor C*	*Investor D*	*Number of Sellers*	*Number of Buyers*	*Net Number of Sellers*
1	-	-	-	-	4	0	4
2	-	-	-	+	3	1	2
3	+	+	-	+	1	3	-2
4	+	+	+	+	0	4	-4

Here a minus sign indicates an asset to be sold and a plus sign one to be purchased. The final columns show the number of investors wishing to sell an asset, the number wishing to buy and the difference between the two.

We can see that since every investor holds the best-performing asset, every investor's portfolio return will be below its return. Hence, every investor will wish to sell shares of asset 1. Conversely, every investor's portfolio return will be greater than the return of the worst-performing asset, so every investor will wish to buy shares of asset 4. As a result, there are only sellers for asset 1 and only buyers for asset 4 so that no trades will incur for these two assets. Therefore, it is impossible for all the investors to rebalance base on the traditional rebalancing method.

In order to solve this problem, in William F. Sharpe (2009)[1] it introduces the an adaptive asset allocation method which is macro-consistent in the sense that it is possible for all investors to follow. Specifically, it lets benchmark ratios change with market change.

It lets $V_{im,t}$ and $V_{im,0}$ be total outstanding market values of asset i at time t and time 0, lets $X_{im,0}$ and $X_{im,t}$ be the proportion of values in the market for asset i at time 0 and time t, and lets the proportion invested in asset i at time 0 and time t be $X_{if,0}$ and $X_{if,t}$. Then the base market asset allocations are:

$$X_{m,0} = (X_{1m,0}, X_{2m,0,} \ldots, X_{nm,0})$$

The base portfolio asset allocations are:

$$X_{f,0} = (X_{1f,0}, X_{2f,0,} \ldots, X_{nf,0})$$

Then it defines the benchmark portfolio asset allocations at time t are:

$$(13) \quad X_{if,t} = \frac{X_{if,0}\left(V_{im,t} / V_{im,0}\right)}{\sum_i X_{if,0}\left(V_{im,t} / V_{im,0}\right)}$$

Since total market value of asset i at time 0 / t equal its proportion of total value $X_{im,0}$ / $X_{im,t}$ times total value of all assets at time $V_{m,0}$ / $V_{m,t}$, therefore:

$$(14) \quad \begin{aligned} V_{im,0} &= X_{im,0} V_{m,0} \\ V_{im,t} &= X_{im,t} V_{m,t} \end{aligned}$$

Put (14) into (13) we have:

$$(15) \quad X_{if,t} = \frac{X_{if,0}\left(X_{im,t} / X_{im,0}\right)}{\sum_i X_{if,0}\left(X_{im,t} / X_{im,0}\right)}$$

In this way, the ratio of the fund's proportion in stocks to that of the market varied. To convert an existing balanced fund to an adaptive one, the stated policy $X_{f,0}$ need only be augmented by the "normal" market conditions $X_{m,0}$.

What is more, if the fund has a base allocation changing in every time period, we can just let $X_{ib,t}$ represent the “base” allocation for time t specified in current policy. This replaces the constant allocation given by $X_{if,0}$ in the formula:

$$(17) \quad X_{if,t} = \frac{X_{ib,t}\left(X_{im,t} / X_{im,0}\right)}{\sum_i X_{ib,t}\left(X_{im,t} / X_{im,0}\right)}$$

3.2 New Portfolio Rebalancing Method

In William F. Sharpe (2009)[1], it provides a way to be consist with the market change. This kind of idea makes sense because fixed benchmark ratios tend to under-weigh the market changing when there is a big move in one asset. Therefore, changing benchmark ratios according to the market change makes contribution to minimize the market changing affection in portfolio management. However, if we strictly follow this idea, no rebalancing will happen after the portfolio asset allocation ratios change to the benchmark ratio. Because according to William F. Sharpe (2009)[1], the benchmark ratios at time t are:

$$(15) \quad X_{if,t} = \frac{X_{if,0}\left(X_{im,t} / X_{im,0}\right)}{\sum_i X_{if,0}\left(X_{im,t} / X_{im,0}\right)}$$

If the portfolio asset ratios $Y_{if,0}$ equal to the benchmark asset ratios $X_{if,0}$ at time 0, we can see at time t the portfolio asset ratios $Y_{if,t}$ will be:

$$Y_{if,t} = \frac{Y_{if,0}\left(X_{im,t}/X_{im,0}\right)}{\sum_{i=1}^{n} Y_{if,0}\left(X_{im,t}/X_{im,0}\right)} = \frac{X_{if,0}\left(X_{im,t}/X_{im,0}\right)}{\sum_{i=1}^{n} X_{if,0}\left(X_{im,t}/X_{im,0}\right)} = X_{if,t}$$

Therefore, if we follow the adaptive method strictly, once the portfolio ratios change to the benchmark ratios, no further rebalancing will happen.

In order to prevent this kind of issue but also take the market changing effects into consideration, we define control parameters k_i to all the assets. Here k_i is a market sensitivity parameter for asset i.

We define the benchmark ratios at time t to be:

$$X_{if,t} = \frac{X_{if,0}\left(X_{im,t}/X_{im,0}\right)^{k_i}}{\sum_{i=1}^{n} X_{if,0}\left(X_{im,t}/X_{im,0}\right)^{k_i}}$$

Here $0 <= k_i <= 1$.

We can see when $k_i = 0$:

$$X_{if,t} = \frac{X_{if,0}}{\sum_{i=1}^{n} X_{if,0}} = X_{if,0}$$

This is the traditional portfolio rebalancing case.

And when $k_i = 1$:

$$X_{if,t} = \frac{X_{if,0}\left(X_{im,t} \big/ X_{im,0} \right)}{\sum_{i=1}^{n} X_{if,0}\left(X_{im,t} \big/ X_{im,0} \right)}$$

This is the adaptive portfolio rebalancing case.

Since Ian Carmichael (2009)[9], it finds out that no matter in any market conditions, in bull market, and in bear market, Masters method is found to provide the most optimal trade-off between tracking error and associated transaction costs. We try to utilize the ideas from Masters method but we make our improvement, for Masters method is based on the traditional fixed rebalancing benchmark.

Here we combine the idea of Masters' method with our new adaptive portfolio rebalancing benchmark ratios, where we apply the Masters portfolio rebalancing method to the moving adaptive benchmark ratios as we define above.

At the beginning, we let the market asset allocation ratios at time 0 be:

$$X_{m,0} = \left(X_{1m,0}, X_{2m,0}, \ldots\ldots, X_{nm,0} \right)$$

And we define our benchmark portfolio asset allocation ratios at time 0 as:

$$X_{f,0} = \left(X_{1f,0}, X_{2f,0}, \ldots\ldots, X_{nf,0} \right)$$

Also, our portfolio holding of asset 1, 2, ..., n at time 0 are:

$$Y_{f,0} = \left(Y_{1f,0}, Y_{2f,0}, \ldots\ldots, Y_{nf,0}\right)$$

We also define the market sensitivity parameter for asset i as k_i where $0 <= k_i <= 1$.

When one period has passed at time t, we assume the market ratio for asset i changes from $X_{im,0}$ to $X_{im,t}$, for every $1<= i <= n$, then from our benchmark changing method, the benchmark portfolio asset allocation ratios at time t becomes:

$$X_{if,t} = \frac{X_{if,0}\left(X_{im,t} / X_{im,0}\right)^{k_i}}{\sum_{i=1}^{n} X_{if,0}\left(X_{im,t} / X_{im,0}\right)^{k_i}}$$

Since in Masters' paper, it defines the trigger point as :

$$T_i = \frac{2KC_i}{\left(Tracking\ Error_i\right)^2}$$

Then we can set $U_{if,t}$ and $L_{if,t}$, the upper band and lower band for triggering rebalancing for asset i, as:

$$U_{if,t} = X_{if,t} + \frac{2KC_i}{\left(TrackingError_i\right)^2}$$

$$L_{if,t} = X_{if,t} - \frac{2KC_i}{\left(TrackingError_i\right)^2}$$

Since portfolio holding of asset i at time 0 is $Y_{if,0}$, with the market changing at time t it becomes:

$$Y_{if,t} = Y_{if,0}\left(X_{im,t} \middle/ X_{im,0} \right)$$

Base on our portfolio rebalancing method:

When $U_{if,t} \le Y_{if,t} \le L_{if,t}$, it keeps unchanged:

$$Y_{if,t} = Y_{if,t}$$

Otherwise, while $Y_{if,t} > U_{if,t}$, we do the rebalancing and change $Y_{if,t}$ to:

$$Y_{if,t} = X_{if,t} + \frac{KC_i}{(TrackingError_i)^2}$$

While $Y_{if,t} < L_{if,t}$, we do the rebalancing and change $Y_{if,t}$ to:

$$Y_{if,t} = X_{if,t} - \frac{KC_i}{(TrackingError_i)^2}$$

Specifically, we move the portfolio ratios to the midpoints of benchmark ratios and triggering bands.

3.3 Back-Testing

Here we did a back-testing for testing this new portfolio rebalancing method. For comparison purpose, we also calculate the results from Calendar Methods, Traditional Two Bands Methods, Asymmetrical Two Bands Methods, and Masters Method. The same as the reason in Ian Carmichael (2009)[9], since the differential equations and Dynamic Programming Method are impractical to implement and very difficult to calculate, they are excluded in the discussion in this paper. We calculate the fixed bands from 1% to 15% for rebalancing back to bands, back to benchmarks, and back to the midpoints of bands and benchmarks respectively so that it should include the bands calculated by Leland method.

In order to more accurately mimic the financial market, here we define the fund's structure, rebalancing criteria, starting situation, and market data all based on a hypothetical multi-clients institutional funds.

We consider the institutional fund with 5 clients, each has assets in equity, fixed income, cash, private equity, and infrastructure respectively. Since the private equity and infrastructure are highly illiquid, we use a combination of equity and fixed income to approximate.

Tables 3.3.1

	Client A		
Asset Class	**Asset**	**Benchmark**	**Benchmark**
Equity	51%	55%	MSCI World
Fixed Income	29%	30%	Barc Agg Bond Index
Cash	2%	1%	91 Day T-Bill
Illiquid - Private Equity	8%	6%	100% MSCI World
Illiquid - Infrastructure	10%	8%	50% MSCI World 50% Barc Agg Bond Index
Total	**100%**	**100%**	

Current Market Value $200,000,000

	Client B		
Asset Class	**Asset**	**Benchmark**	**Benchmark**

Equity	30%	25%	MSCI World
Fixed Income	40%	30%	Barc Agg Bond Index
Cash	2%	5%	91 Day T-Bill
Illiquid - Private Equity	10%	16%	100% MSCI World
Illiquid - Infrastructure	18%	24%	75% MSCI World 25% Barc Agg Bond Index
Total	**100%**	**100%**	

Current Market Value $100,000,000

	Client C		
Asset Class	**Asset**	**Benchmark**	**Benchmark**
Equity	62%	60%	MSCI World
Fixed Income	7%	10%	Barc Agg Bond Index
Cash	3%	5%	91 Day T-Bill
Illiquid - Private Equity	13%	13%	100% MSCI World
Illiquid - Infrastructure	15%	13%	75% MSCI World 25% Barc Agg Bond Index
Total	**100%**	**100%**	

Current Market Value $10,000,000

	Client D1		
Asset Class	**Asset**	**Benchmark**	**Benchmark**
Equity	53%	50%	MSCI World
Fixed Income	37%	40%	Barc Agg Bond Index
Cash	5%	5%	91 Day T-Bill
Illiquid - Private Equity	2%	3%	100% MSCI World
Illiquid - Infrastructure	3%	2%	50% MSCI World 50% Barc Agg Bond Index
Total	**100%**	**100%**	

Current Market Value $10,000,000

	Client D2		
Asset Class	**Asset**	**Benchmark**	**Benchmark**
Equity	51%	50%	MSCI World
Fixed Income	39%	40%	Barc Agg Bond Index
Cash	1%	5%	91 Day T-Bill
Illiquid - Private Equity	6%	3%	100% MSCI World
Illiquid - Infrastructure	3%	2%	50% MSCI World 50% Barc Agg Bond

			Index
Total	**100%**	**100%**	

Current Market Value $3,000,000

Therefore when we need to rebalance the private equity or infrastructure, we rebalance the equivalent amount of equity and fixed income according to their public benchmark defined above (private investment benchmarking is not within the scope of this paper). What is more, when doing the rebalancing, we use the cash account first once it has enough money (greater than minimum requirement and the benchmark ratio for cash account). There is no transaction cost for cash. The transaction cost for equity is 10 bps and that for fixed income is 5 bps. Moreover, when doing the rebalancing, preference is given to internal cross trades (between clients) first before buying or selling in the market. The first 30 internal trades are free. After the first 30 trades, the transaction cost is 1 bps.

Base on the portfolio rebalancing platform we developed in Matlab (sample codes are introduced in the end), for fixed band with fully rebalancing (rebalance to the benchmark exactly) from 1% to 15%, we can see the return of the fund and returns of each client is:

Table 3.3.2

Return Rate	Total	Client A	Client B	Client C	Client D1	Client D2
Fixed Band, Fully Rebalancing, 1%	0.6081	0.6068	0.6195	0.4135	0.6886	0.6886
Fixed Band, Fully Rebalancing, 2%	0.6211	0.6241	0.6275	0.4166	0.6804	0.6938
Fixed Band, Fully Rebalancing, 3%	0.6089	0.6083	0.6166	0.4360	0.6936	0.6840
Fixed Band, Fully Rebalancing, 4%	0.6023	0.5958	0.6241	0.3914	0.6923	0.7116

Fixed Band, Fully Rebalancing, 5%	0.6284	0.6234	0.6516	0.4272	0.6762	0.6989
Fixed Band, Fully Rebalancing, 6%	0.6161	0.6201	0.6167	0.3725	0.7456	0.7081
Fixed Band, Fully Rebalancing, 7%	0.6325	0.6208	0.6719	0.3832	0.6954	0.7115
Fixed Band, Fully Rebalancing, 8%	0.6127	0.6059	0.6361	0.3954	0.6978	0.7191
Fixed Band, Fully Rebalancing, 9%	0.6951	0.7147	0.6804	0.4072	0.7223	0.7550
Fixed Band, Fully Rebalancing, 10%	0.6520	0.6689	0.6343	0.4186	0.6960	0.7457
Fixed Band, Fully Rebalancing, 11%	0.6873	0.7038	0.6766	0.4333	0.7143	0.7022
Fixed Band, Fully Rebalancing, 12%	0.6237	0.6533	0.5773	0.3543	0.7369	0.7148
Fixed Band, Fully Rebalancing, 13%	0.6417	0.6746	0.6011	0.3543	0.6463	0.7430
Fixed Band, Fully Rebalancing, 14%	0.6748	0.7105	0.6370	0.3544	0.6718	0.6301
Fixed Band, Fully Rebalancing, 15%	0.6124	0.5790	0.6971	0.3544	0.6818	0.6411

For fixed band with half rebalancing (rebalance to the midpoint of benchmark and triggering band) from 1% to 15%, we can see the return of the fund and returns of each client is:

Table 3.3.3

Return Rate	Total	Client A	Client B	Client C	Client D1	Client D2
Fixed Band, Half Rebalancing, 1%	0.6135	0.6142	0.6214	0.4121	0.6954	0.6957
Fixed Band, Half Rebalancing, 2%	0.6187	0.6158	0.6351	0.4132	0.6994	0.6851
Fixed Band, Half Rebalancing, 3%	0.6178	0.6089	0.6448	0.4146	0.7066	0.6939
Fixed Band, Half Rebalancing, 4%	0.6356	0.6372	0.6457	0.4059	0.7092	0.7077
Fixed Band, Half Rebalancing, 5%	0.6416	0.6425	0.6543	0.4083	0.7073	0.7114
Fixed Band, Half Rebalancing, 6%	0.6443	0.6444	0.6609	0.3790	0.7248	0.6997
Fixed Band, Half Rebalancing, 7%	0.6526	0.6556	0.6623	0.4032	0.7218	0.7309
Fixed Band, Half Rebalancing, 8%	0.6361	0.6478	0.6271	0.3810	0.7126	0.7567
Fixed Band, Half Rebalancing, 9%	0.6290	0.6128	0.6715	0.3872	0.7394	0.7353
Fixed Band, Half Rebalancing, 10%	0.6343	0.6301	0.6572	0.3925	0.6922	0.7671
Fixed Band, Half Rebalancing, 11%	0.6322	0.6062	0.6977	0.4000	0.7068	0.7029
Fixed Band, Half Rebalancing, 12%	0.6341	0.6294	0.6593	0.3554	0.7336	0.7085
Fixed Band, Half Rebalancing, 13%	0.6627	0.6645	0.6859	0.3554	0.6790	0.7430
Fixed Band, Half Rebalancing, 14%	0.6185	0.6089	0.6525	0.3554	0.7140	0.6796
Fixed Band, Half Rebalancing, 15%	0.6419	0.6319	0.6796	0.3554	0.7282	0.7187

Fox fixed band with marginal rebalancing (rebalance to the nearest triggering band) from 1% to 15%, we can see the return of the fund and returns of each client is:

Table 3.3.4

Return Rate	Total	Client A	Client B	Client C	Client D1	Client D2
Fixed Band, Marginal Rebalancing, 1%	0.6157	0.6146	0.6283	0.4086	0.6968	0.6922
Fixed Band, Marginal Rebalancing, 2%	0.6165	0.6127	0.6337	0.4107	0.7031	0.6902
Fixed Band, Marginal Rebalancing, 3%	0.6279	0.6249	0.6458	0.4021	0.7097	0.7054
Fixed Band, Marginal Rebalancing, 4%	0.6403	0.6376	0.6597	0.3913	0.7247	0.7178
Fixed Band, Marginal Rebalancing, 5%	0.6453	0.6453	0.6593	0.3903	0.7344	0.7317
Fixed Band, Marginal Rebalancing, 6%	0.6451	0.6477	0.6528	0.3892	0.7417	0.7418
Fixed Band, Marginal Rebalancing, 7%	0.6409	0.6412	0.6536	0.3862	0.7328	0.7405
Fixed Band, Marginal Rebalancing, 8%	0.6371	0.6315	0.6636	0.3813	0.7145	0.7286
Fixed Band, Marginal Rebalancing, 9%	0.6421	0.6363	0.6722	0.3752	0.7024	0.7167
Fixed Band, Marginal Rebalancing, 10%	0.6476	0.6401	0.6828	0.3685	0.7022	0.7183
Fixed Band, Marginal Rebalancing, 11%	0.6512	0.6429	0.6898	0.3596	0.7038	0.7190
Fixed Band, Marginal Rebalancing, 12%	0.6545	0.6447	0.6971	0.3562	0.7038	0.7189
Fixed Band, Marginal Rebalancing, 13%	0.6575	0.6457	0.7054	0.3562	0.6972	0.7180
Fixed Band, Marginal Rebalancing, 14%	0.6590	0.6460	0.7103	0.3562	0.6920	0.7163
Fixed Band, Marginal Rebalancing, 15%	0.6596	0.6448	0.7152	0.3562	0.6865	0.7138

For asymmetrical band with fully rebalancing (rebalance to the benchmark exactly) from 1%, 2% to 6%, 7%, we can see the return of the fund and returns of each client is:

Table 3.3.5

Return Rate	Total	Client A	Client B	Client C	Client D1	Client D2
Asymmetrical,Fully Rebalancing, 1%, 2%	0.6110	0.6108	0.6199	0.4123	0.6971	0.6959
Asymmetrical,Fully Rebalancing, 1%, 3%	0.6107	0.6081	0.6227	0.4196	0.7042	0.7044
Asymmetrical,Fully Rebalancing, 1%, 4%	0.6088	0.6060	0.6213	0.4191	0.7013	0.7014
Asymmetrical,Fully Rebalancing, 1%, 5%	0.6069	0.6039	0.6199	0.4185	0.6983	0.6985
Asymmetrical,Fully Rebalancing, 1%, 6%	0.6051	0.6019	0.6185	0.4179	0.6954	0.6956
Asymmetrical,Fully Rebalancing, 1%, 7%	0.6032	0.5998	0.6171	0.4174	0.6924	0.6926
Asymmetrical,Fully Rebalancing, 2%, 3%	0.6105	0.6037	0.6349	0.4186	0.6755	0.6708
Asymmetrical,Fully Rebalancing, 2%, 4%	0.6245	0.6179	0.6504	0.4283	0.6746	0.6846
Asymmetrical,Fully Rebalancing, 2%, 5%	0.6157	0.6165	0.6246	0.4396	0.6694	0.6784
Asymmetrical,Fully Rebalancing, 2%, 6%	0.6171	0.6151	0.6320	0.4391	0.6682	0.6771

Asymmetrical,Fully Rebalancing, 2%, 7%	0.6172	0.6137	0.6353	0.4387	0.6669	0.6758
Asymmetrical,Fully Rebalancing, 3%, 4%	0.6028	0.5964	0.6227	0.4238	0.6858	0.6851
Asymmetrical,Fully Rebalancing, 3%, 5%	0.5994	0.5926	0.6206	0.4036	0.6950	0.6852
Asymmetrical,Fully Rebalancing, 3%, 6%	0.6052	0.5917	0.6413	0.4231	0.6769	0.6740
Asymmetrical,Fully Rebalancing, 3%, 7%	0.6052	0.5908	0.6440	0.4138	0.6785	0.6730
Asymmetrical,Fully Rebalancing, 4%, 5%	0.5911	0.5866	0.6107	0.3738	0.6769	0.6781
Asymmetrical,Fully Rebalancing, 4%, 6%	0.5889	0.5872	0.6014	0.3778	0.6728	0.7127
Asymmetrical,Fully Rebalancing, 4%, 7%	0.5967	0.5865	0.6254	0.3777	0.7058	0.6860
Asymmetrical,Fully Rebalancing, 5%, 6%	0.6051	0.6162	0.5966	0.3888	0.6677	0.6575
Asymmetrical,Fully Rebalancing, 5%, 7%	0.6096	0.6237	0.5941	0.3963	0.6832	0.6607
Asymmetrical,Fully Rebalancing, 6%, 7%	0.6135	0.6065	0.6401	0.3676	0.7078	0.7024

For asymmetrical band with half rebalancing (rebalance to the midpoint of benchmark and triggering band) from 1%, 2% to 6%, 7%, we can see the return of the fund and returns of each client is:

Table 3.3.6

Return Rate	Total	Client A	Client B	Client C	Client D1	Client D2
Asymmetrical,Half Rebalancing, 1%, 2%	0.6173	0.6128	0.6363	0.4126	0.6973	0.6939
Asymmetrical,Half Rebalancing, 1%, 3%	0.6164	0.6135	0.6322	0.4139	0.6950	0.6918
Asymmetrical,Half Rebalancing, 1%, 4%	0.6158	0.6119	0.6338	0.4135	0.6929	0.6896
Asymmetrical,Half Rebalancing, 1%, 5%	0.6144	0.6103	0.6329	0.4131	0.6908	0.6874
Asymmetrical,Half Rebalancing, 1%, 6%	0.6129	0.6086	0.6317	0.4126	0.6888	0.6852
Asymmetrical,Half Rebalancing, 1%, 7%	0.6114	0.6070	0.6306	0.4122	0.6867	0.6830
Asymmetrical,Half Rebalancing, 2%, 3%	0.6149	0.6071	0.6397	0.4171	0.6974	0.6917
Asymmetrical,Half Rebalancing, 2%, 4%	0.6238	0.6124	0.6588	0.4179	0.6920	0.6798
Asymmetrical,Half Rebalancing, 2%, 5%	0.6217	0.6151	0.6461	0.4221	0.6915	0.6817
Asymmetrical,Half Rebalancing, 2%, 6%	0.6220	0.6141	0.6493	0.4219	0.6904	0.6804
Asymmetrical,Half Rebalancing, 2%, 7%	0.6226	0.6130	0.6534	0.4216	0.6894	0.6792
Asymmetrical,Half Rebalancing, 3%, 4%	0.6099	0.6103	0.6181	0.4171	0.6907	0.6851
Asymmetrical,Half Rebalancing, 3%, 5%	0.6165	0.6095	0.6417	0.4131	0.6885	0.6860
Asymmetrical,Half Rebalancing, 3%, 6%	0.6162	0.6087	0.6412	0.4171	0.6951	0.6850
Asymmetrical,Half Rebalancing, 3%, 7%	0.6155	0.6080	0.6404	0.4169	0.6943	0.6840
Asymmetrical,Half Rebalancing, 4%, 5%	0.6219	0.6268	0.6265	0.3898	0.6888	0.6910
Asymmetrical,Half Rebalancing, 4%, 6%	0.6264	0.6305	0.6315	0.3920	0.7070	0.6951
Asymmetrical,Half Rebalancing, 4%, 7%	0.6265	0.6298	0.6336	0.3919	0.7031	0.6944
Asymmetrical,Half Rebalancing, 5%, 6%	0.6292	0.6273	0.6481	0.3916	0.6999	0.6850

Asymmetrical,Half Rebalancing, 5%, 7%	0.6285	0.6266	0.6453	0.3916	0.7162	0.6867
Asymmetrical,Half Rebalancing, 6%, 7%	0.6277	0.6298	0.6391	0.3752	0.6987	0.7123

For asymmetrical band with marginal rebalancing (rebalance to the nearest triggering band) from 1%, 2% to 6%, 7%, we can see the return of the fund and returns of each client is:

Table 3.3.7

Return Rate	Total	Client A	Client B	Client C	Client D1	Client D2
Asymmetrical,Marginal Rebalancing, 1%, 2%	0.6175	0.6164	0.6294	0.4140	0.7034	0.6877
Asymmetrical,Marginal Rebalancing, 1%, 3%	0.6175	0.6145	0.6328	0.4162	0.7012	0.6955
Asymmetrical,Marginal Rebalancing, 1%, 4%	0.6205	0.6132	0.6454	0.4158	0.6995	0.6939
Asymmetrical,Marginal Rebalancing, 1%, 5%	0.6195	0.6119	0.6451	0.4155	0.6978	0.6922
Asymmetrical,Marginal Rebalancing, 1%, 6%	0.6183	0.6106	0.6442	0.4151	0.6961	0.6905
Asymmetrical,Marginal Rebalancing, 1%, 7%	0.6172	0.6093	0.6432	0.4148	0.6944	0.6888
Asymmetrical,Marginal Rebalancing, 2%, 3%	0.6129	0.6075	0.6338	0.4081	0.6939	0.6890
Asymmetrical,Marginal Rebalancing, 2%, 4%	0.6173	0.6055	0.6508	0.4096	0.7018	0.6919
Asymmetrical,Marginal Rebalancing, 2%, 5%	0.6177	0.6047	0.6536	0.4127	0.7047	0.6876
Asymmetrical,Marginal Rebalancing, 2%, 6%	0.6143	0.6038	0.6442	0.4126	0.7052	0.6866
Asymmetrical,Marginal Rebalancing, 2%, 7%	0.6205	0.6030	0.6660	0.4124	0.7043	0.6856
Asymmetrical,Marginal Rebalancing, 3%, 4%	0.6206	0.6155	0.6424	0.3993	0.7047	0.6932
Asymmetrical,Marginal Rebalancing, 3%, 5%	0.6205	0.6148	0.6426	0.3916	0.7162	0.7085
Asymmetrical,Marginal Rebalancing, 3%, 6%	0.6219	0.6141	0.6488	0.3915	0.7128	0.7089
Asymmetrical,Marginal Rebalancing, 3%, 7%	0.6247	0.6134	0.6590	0.3915	0.7166	0.7081
Asymmetrical,Marginal Rebalancing, 4%, 5%	0.6306	0.6260	0.6534	0.3886	0.7162	0.7021
Asymmetrical,Marginal Rebalancing, 4%, 6%	0.6323	0.6254	0.6592	0.3886	0.7182	0.7168
Asymmetrical,Marginal Rebalancing, 4%, 7%	0.6250	0.6248	0.6366	0.3885	0.7216	0.7193
Asymmetrical,Marginal Rebalancing, 5%, 6%	0.6343	0.6325	0.6515	0.3875	0.7199	0.7186
Asymmetrical,Marginal Rebalancing, 5%, 7%	0.6336	0.6320	0.6494	0.3875	0.7297	0.7200
Asymmetrical,Marginal Rebalancing, 6%, 7%	0.6298	0.6344	0.6330	0.3864	0.7200	0.7210

For calendar method with monthly, quarterly, semi-annually, and annually rebalancing, we can see the return of the fund and returns of each client is:

Table 3.3.8

Return Rate	Total	Client A	Client B	Client C	Client D1	Client D2
Calendar, Monthly Rebalancing	0.5949	0.5952	0.6036	0.4004	0.6726	0.6726
Calendar, Quarterly Rebalancing	0.6281	0.6296	0.6351	0.4193	0.7134	0.7134
Calendar, Semi-annually Rebalancing	0.5447	0.5426	0.5573	0.3741	0.6109	0.6109
Calendar, Annually Rebalancing	0.5652	0.5648	0.5755	0.3793	0.6344	0.6344

For Masters method, we can see the portfolio returns are:

Table 3.3.9

Return Rate	Total	Client A	Client B	Client C	Client D1	Client D2
Marsters' Method	0.7151	0.6310	0.9016	0.4271	0.7942	0.7998

For our new portfolio rebalancing method, we can see the return of the fund and returns of each client is:

Table 3.3.10

Return Rate	Total	Client A	Client B	Client C	Client D1	Client D2
Our Method	0.7301	0.6631	0.9016	0.4271	0.6572	0.7365

All in all, we can see that from the tables above our new portfolio rebalancing method performs the best among all the methods.

Part II

Chapter 4

Modeling with Fractional Brownian Motion

Mathematical Finance was seriously developed with the help of the theory of self-similar (long-memory, long-range dependent, fractional) processes. On the one hand, according to A.N.Kolmogorov (1940)[10], this theory was first initiated. On the other hand by H. Hurst (1951)[11], it gave an experimental motivation/background for this theory. Hurst introduced a special parameter H which was interpreted later as a quantitative characteristics of long-range dependence of the the process. The Hurst experimental results were theoretically developed by B.B. Mandelbrott in many directions (including finance) and publications (see, for instance his book "The Fractional Geometry and Nature", San-Francisco, W.H.Freemann, 1982). He also proposed to call H as the Hurst parameter. By the way, the processes studied by A.N.Kolmogorov as the Wiener Spirals later became another name as Fractional Brownian Motions. A comprehensive theory as well as financial and statistical applications of these processes can be found in the book by Yu.Mishura "Stochastic calculus for fractional Brownian motion and related processes", Springer, Lecture Notes in Mathematics 1929 (2008).

Here we give an introduction to fractional Brownian motion and Mixed Brownian and

Fractional Brownian Model.

4.1 Tradition fractional Brownian motion

A fractional Brownian motion is a generalization of Brownian motion without independent increments. It is a continuous-time Gaussian process $B_t^H, t \in [0,T]$, which:

(i) It starts at 0:

$$B_0^H = 0$$

(ii) It has expectation 0 for all t in [0, T]:

$$E[B_t^H] = 0, t \in \Re^+$$

(iii) It has the co-variance functions as:

$$E[B_t^H B_s^H] = \tfrac{1}{2}(|t|^{2H} + |s|^{2H} - |t-s|^{2H})$$

Here H is called Hurst parameter where:

$$H \in \Re^+, H \in (0,1)$$

Value of H determines what kind of process the fractional Brownian motion is:

If H = 1/2, the process is Brownian motion or Wiener process.

If H > 1/2, the increments of the process are positively correlated.

If H < 1/2, the increments of the process are negatively correlated.

The increment process is known as fractional Gaussian noise:

$$X(t) = B_{t+1}^{H} - B_{t}^{H}$$

4.2 Mixed Brownian and Fractional Brownian Model

In the financial market, we assume $\{W_t, t \geq 0\}$ is independent Brownian motion and $\{B_t^H, t \geq 0\}$ is fractional Brownian motion with Hurst index H > 1/2.

Then the financial market includes 2 assets:

(1) non-risky asset:

$$S_t^0 = S_0^0 e^{rt}, t \geq 0, S_0^0 > 0$$

(2) risky asset governed by the linear combination of W and B^H:

$$S_t = S_0 e^{\mu t + \sigma_1 W_t + \sigma_2 B_t^H}, t \geq 0, S_0 > 0$$

Here:

$r > 0$ is the constant risk free rate.

$\mu \in \Re$ is the drift coefficient.

$\sigma_1 > 0$ is the volatility for the standard Brownian motion W.

$\sigma_2 > 0$ is the volatility for the fractional Brownian motion B^H.

Then the discounted price process has the form:

$$X_t = S_t e^{-rt} = S_0 e^{(\mu - r)t + \sigma_1 (W_t + \sigma B_t^H)}$$

Chapter 5

Why We Need H > 3/4?

A long-range dependence phenomena is observed in many fields. From the point of view of the theory of stochastic processes, one of the most convenient stochastic models to describe the long-range dependence is the fractional Brownian motion with the Hurst parameter H > 1/2. However, the it has a disadvantage that it might admit arbitrage opportunity for a wide class of self-financing strategies.

In Melnikov & Mishura (2011)[12], it points out that mixed financial model that includes both Brownian and fractional Brownian components could avoid this problem. More importantly, it provides approaches that for H in (3/4, 1), the linear combination of a Wiener process and a fractional Brownian motion with H in (3/4, 1) is a semi-martingale with respect to the natural filtration. Therefore we could treat such market as a standard semi-martingale one.

According to Melnikov & Mishura (2011)[12], for fixed T > 0, we consider the market on the interval [0, T], we denote the mixed process $M_t^{H,\sigma} = W_t + \sigma B_t^H, t \in [0,T]$ and filtration $F_t^M = \sigma\{M_u^{H,\sigma}, 0 \le u \le t\}$.

In Melnikov & Mishura (2011)[12], it develops following 6 properties:

1. The mixed process $M_t^{H,\sigma} = W_t + \sigma B_t^H$ where t is in [0, T] is equivalent in measure to a Brownian motion if and only if H is in (3/4, 1).

2. For H is in (3/4, 1), there exists a unique real-valued Volterra kernel [illegible] $T]^2)$ such that (t is in [0, T]):

$$B_t := M_t^{H,\sigma} - \int_0^t \int_0^s \tilde{} \text{[illegible]} {}_u^{H,\sigma} ds$$

3. The representation below is unique:

$$\sigma^2 H(2H-1)(t-s)^{2H-2} = r_\sigma(t,s) + \int_0^s r_\sigma(t,x) r_\sigma(s,x) dx, 0 \le s < t \le T.$$

If $\tilde{L}$ is a Brownian motion on (Ω, F, P) and $m \in L_2([0,T]^2)$ is a real-valued Volterra Kernel such that

$$M_t^{H,\sigma} := \tilde{L}_t \text{[illegible]} \int_0^t \int_0^s \text{[illegible]} {}_u \text{[illegible]}$$

4. As a consequence of Property 3, the process $\sigma_1 W + \sigma_2 B^H$ is a semi-martingale with respect to its natural filtration.

5. Let B_t be a a Brownian motion on probability space (Ω, F, P) and $k \in L_2([0,T]^2)$ be a real-valued Volterra Kernel, $a \in L_2([0,T])$. Then:

$$E\exp\left(\int_0^t\int_0^s k(s,u)dB_u dB_s - \tfrac{1}{2}\int_0^t\left(\int_0^s k(s,u)dB_u\right)^2 ds\right) = 1.$$

6. If we consider the following class of strategies

$S = \{\Psi = (\Psi^{(1)}, \Psi^{(2)})\}$: $\psi^{(1)}$ and $\psi^{(2)}$ are $\bar{F}^M - predictable$,

$\int_0^T \left|\psi_u^{(1)}\right| du < \infty, \int_0^T \left|\psi_u^{(2)}\right|^2 du < \infty$ a.s.,

Then the discounted capital $V_t = V_0 + \int_0^t \psi_u^{(2)} dX_u, t \in [0,T]$ satisfies the condition

$\inf_{t\in[0,T]} V_t \geq 0$ a.s.

Therefore we can see it is very important that H is > 3/4. When H is in (3/4, 1), the objective and the martingale measures coincide we obtain simple formulas to compute the solution of efficient hedging problem. Therefore, it is very important for us to find out H in (3/4, 1) in the real market.

Chapter 6

Estimate H all over the World

Since when the Hurst parameter > 3/4, we have such a lot interesting properties and beautiful results. Here we take a look at the estimated H values for equity, fixed income, and foreign exchange markets all over the world, respectively.

6.1 Estimate Hurst Parameter H

Since a fractional Brownian motion is a continuous-time Gaussian process depending on the Hurst parameter 0 < H < 1, the fractional Brownian motion is self-similar in distribution and the variance of the increments is given by:

$$Var(B_t^H - B_s^H) = v * |t-s|^{2H}$$

where here v is a positive constant.

The special form of the variance of the increments suggests various ways to estimate the parameter H. In the Book <Theory and Applications of Long-Range Dependence>[13], it explores many methods.

There are 3 commonly used methods to estimate the Hurst parameter H. The first two methods are based on second order discrete derivative, where the second one is wavelet-based[14]. The third estimate is based on the linear regression in log-log plot, of the variance of detail versus level[15]. Since first two methods give similar results with smaller dispersion than the third one, we focus on the estimated H from the first two methods.

6.2 Empirical Results for World Equity Market

We downloaded the stock index data for countries all over the world since January, 2000 from Bloomberg. Then base on the estimation platform we developed in the Matlab (sample codes are provided in the end), we can see the estimated H of countries' stock indexes which is > 0.75 are:

Table 6.2.1

Excellent:	Method 1	Method 2
Vietnam	0.8165	0.7919
Slovenia	0.8529	0.8441
Malta	0.7953	0.868
Ukraine	0.8013	0.81
Nigeria	0.8713	0.9143
Kenya	0.8607	0.9194
Morocco	0.7534	0.7569
Qatar	0.7632	0.7765
Palestine	0.8090	0.8839

We find that Vietnam, Slovenia, Malta, Ukraine, Nigeria, Kenya, Morocco, Qatar, Palestine have estimated H > 0.75 in both methods.

Table 6.2.2

Good:	Method 1	Method 2

Chile	0.7298	0.7587
Laos	0.7026	0.7292
Tunisia	0.7348	0.719
Oman	0.7153	0.7335
Lebanon	0.7379	0.717
Jordan	0.7846	0.7351

According to table 6.2.2 above, we also find that Chile, Jordan, Oman, Lebanon, Laos, Tunisia, Serbia have estimated H > 0.7 in both methods or > 0.75 in one method and close to 0.7 in another method.

Therefore, above countries can be considered as having H > 0.75 or close to 0.75 in their stock indexes.

Specifically, if we use the average of estimated H from both estimation methods as the our estimated H, the distribution of the estimated H of stock indexes for all the countries are:

Table 6.2.3

	0.75<=H<1
Vietnam	0.8042
Slovenia	0.8485
Serbia	0.9948
Malta	0.8317
Ukraine	0.8057
Nigeria	0.8928
Kenya	0.8900
Morocco	0.7552
Jordan	0.7598
Qatar	0.7698
Palestine	0.8465

0.7<=H<0.75	
Chile	0.7442
Mongolia	0.7351
Laos	0.7159
Tunisia	0.7269
Oman	0.7244
Lebanon	0.7274
0.65<=H<0.7	
Sri Lanka	0.6679
Philippines	0.6749
Egypt	0.6620
Bahrain	0.6649
0.6<=H<0.65	
Colombia	0.6105
Venezuela	0.6290
UK2	0.6027
Iceland	0.6176
Greece	0.6321
Pakistan	0.6312
Indonesia	0.6287
India	0.6141
Malaysia	0.6236
Croatia	0.6015
Lithuania	0.6186
Bulgaria	0.6135
Kuwait	0.6422
0.55<=H<0.6	
Mexico	0.5930
Peru	0.5977
Portugal	0.5616
Ireland	0.5639
Belgium	0.5714
Denmark	0.5666
Austria	0.5542
Poland	0.5698
Czech Republic	0.5835
China3	0.5572
New Zealand	0.5863
Japan1	0.5564
Estonia	0.5877

Cyprus	0.5595
Russia2	0.5595
Hungary	0.5825
Romania	0.5709
Ghana	0.5924
Namibia	0.5707
Mauritius	0.5660
Saudi Arabia	0.5998
0.5<=H<0.55	
Canada	0.5025
USA3	0.5104
Brazil	0.5059
Bloomberg Euro	0.5103
EUROPE 600	0.5094
Europe 350	0.5030
UK3	0.5038
Germany	0.5028
France2	0.5064
Spain	0.5259
Switzerland	0.5496
Luxembourg	0.5229
Finland	0.5133
Norway	0.5062
Taiwan	0.5399
South Korea	0.5361
Thailand	0.5184
Singapore	0.5170
South Africa1	0.5369
South Africa2	0.5451
United Arab	0.5122
Israel	0.5077
0.45<=H<0.5	
Jamaica	0.4743
Eurotop 100	0.4929
UK1	0.4940
France1	0.4908
Italy	0.4594
Netherland	0.4989
Sweden	0.4838
Australia	0.4603

Japan2		0.4729
Hong Kong		0.4977
China1		0.4979
China2		0.4801
Slovakia		0.4745
Latvia		0.4921
Turkey		0.4578
Russia1		0.4964
Botswana		0.4516
	0.4<=H<0.45	
USA1		0.4305
USA2		0.4236
USA4		0.4356
USA5		0.4342
Argentina		0.4489
Bangladesh		0.4036
	0.35<=H<0.4	
Bermuda		0.3801
	0.3<=H<0.35	
Panama		0.3346
	0<=H<0.3	
Costa Rica		0.1584
Kazakhstan		0.2963
Tanzania		0.2843

What is more, since when H = 0.5, it is the Wiener process. Therefore when the estimated H is closed to 0.5, it could be considered as Wiener process. By Matlab output, if we use the average of estimated H from both estimation methods as the our estimated H, the countries with estimated H closed to 0.5 are:

Table 6.2.4

	H that closes to 0.5	
Canada		0.5025
USA3		0.5104
Brazil		0.5059
Bloomberg Euro		0.5103

Eurotop 100	0.4929
EUROPE 600	0.5094
Europe 350	0.5030
UK1	0.4940
UK3	0.5038
Germany	0.5028
France1	0.4908
France2	0.5064
Netherland	0.4989
Luxembourg	0.5229
Finland	0.5133
Norway	0.5062
Sweden	0.4838
Thailand	0.5184
Singapore	0.5170
Hong Kong	0.4977
China1	0.4979
China2	0.4801
Latvia	0.4921
Russia1	0.4964
United Arab	0.5122
Israel	0.5077

Then we compare countries in different regions, if we use the average of estimated H from both estimation methods as the our estimated H, the estimated H of them are:

Table 6.2.5

Estimated H:	Average	Standard Deviation
North America	0.4561	0.0393
South America	0.4979	0.1638
West Europe	0.5332	0.0441
Asia	0.5782	0.1001
East Europe	0.6881	0.247
Africa	0.6280	0.183
Middle East	0.6655	0.1109

We can see that countries in East Europe has the highest Average Estimated H (which is close to 0.7) but the variance is also big. The countries in Middle East also has the Average Estimated H close to 0.7 but the variance is modest. On the other hand, the countries in North America and West Europe both have Estimated H close to 0.5 and have very low variance. This means that stock indexes in those regions are close to Wiener processes.

Moreover, we also compare the H values for stock indexes of countries for developed vs developing countries. if we use the average of estimated H from both estimation methods as the our estimated H, The estimated H of them are:

Table 6.2.6

Estimated H:	Average	Standard Deviation
Developed Country	0.5411	0.0874
Developing Country	0.6252	0.1942

We can see that the developing countries have higher estimated H comparing to developed countries. On the other hand, they also have higher variance for estimated H comparing to developed countries.

6.3 Empirical Results for World Foreign Exchange Market

We downloaded the foreign exchange data for major countries all over the world since January, 2000 from Bloomberg. After running the Matlab programs, we find that in the foreign exchange market there are no currency with estimated Hurst parameter $H > 0.75$ or even close to 0.7 in both estimation methods. Therefore, no countries should be considered as having $H > 0.75$ or close to 0.75 in their foreign exchange rates.

Specifically, if we use the average of estimated H from both estimation methods as the our estimated H, the distribution of estimated H of foreign exchange for all the countries are:

Table 6.3.1

0.6<=H<0.65	
Argentina	0.6297
0.55<=H<0.6	
South Korea	0.5553
India	0.5824
Hong Kong	0.5685
Ukraine	0.5727
Romania	0.5502
Chile	0.5706
Israel	0.5520
0.5<=H<0.55	
Sweden	0.5059
UK	0.5400
Europe	0.5047
Taiwan	0.5369
Singapore	0.5001
Turkey	0.5222
Poland	0.5465
Hungary	0.5341
Venezuela	0.5355
Colombia	0.5354
Brazil	0.5453
Saudi Arab	0.5182
0.45<=H<0.5	
Norway	0.4876
Japan	0.4591
Denmark	0.4770
Switzerland	0.4572
New Zealand	0.4941

Australia	0.4545
Japan	0.4591
Thailand	0.4980
Philippine	0.4819
Malaysia	0.4962
Indonesia	0.4685
China	0.4717
South Africa	0.4856
Russia	0.4729
Iceland	0.4748
Czech Republic	0.5000
Bulgaria	0.4888
Peru	0.4761
Morocco	0.4658
Iran	0.4869
0.4<=H<0.45	
Canada	0.4344
Mexica	0.4367
0.35<=H<0.4	
Syria	0.3844
Egypt	0.3660
0<=H<0.3	
Qatar	0.1252
Oman	0.1995
Lebanon	0.1603
Kuwait	0.1993
Jordan	0.1783
Bahrain	0.0886
United Arab	0.2173

What is more, since when H = 0.5, it is the Wiener process. Therefore when the estimated H is closed to 0.5, it could be considered as Wiener process. Then if we use the average of estimated H from both estimation methods as the our estimated H, the countries with estimated H closed to 0.5 are:

Table 6.3.2

H that closes to 0.5	
Sweden	0.5059
Norway	0.4876
Denmark	0.4770
New Zealand	0.4941
Europe	0.5047
Thailand	0.4980
Singapore	0.5001
Philippine	0.4819
Malaysia	0.4962
South Africa	0.4856
Turkey	0.5222
Czech Republic	0.5000
Bulgaria	0.4888
Peru	0.4761
Saudi Arab	0.5182
Iran	0.4869

Then we compare the countries in different regions, if we use the average of estimated H from both estimation methods as the our estimated H, the estimated H of them are:

Table 6.3.3

Estimated H:	Average	Standard Deviation
G10	0.4814	0.0313
Asia	0.5108	0.0429
East Europe Africa	0.5148	0.0352
Latin America	0.5328	0.0625
Middle East	0.3032	0.1637

We can see that the average estimated H for exchange rates for G10, Asian, East Europe, African, and Latin American countries are all close to 0.5, which means they close to Wiener process. For Middle East countries, they have an average estimated H which is closed to 0.3.

6.4 Empirical Results for World Fixed Income Market

We downloaded the bond index data for the countries all over the world since January, 2000 from Bloomberg. By running Matlab program, we have:

Table 6.4.1

Excellent:	Method 1	Method 2
Portugal	0.7593	0.8342
Greece	0.7630	0.8180
Colombia	0.8262	0.8171
Mexico	0.7565	0.7633
Indonesia	0.8925	0.9179
Philippine	0.7724	0.8216

We find out that Portugal, Greece, Colombia, Mexico, Indonesia, Philippine have estimated H > 0.75 in both methods.

Table 6.4.2

Good:	Method 1	Method 2
Italy	0.7065	0.7327
Spain	0.7442	0.8025
Ireland	0.7289	0.7351
Belgium	0.7234	0.7612
Norway	0.7169	0.7069
Egypt	0.7161	0.7712
Romania	0.8534	0.7471
Peru	0.7643	0.7450
Malaysia	0.7700	0.7393

We also find Spain, Romania, Peru, Malaysia, Belgium, Egypt, Italy, Ireland have estimated H > 0.7 in both methods or > 0.75 in one method and > 0.7 in another method.

What is more, although we know that 3rd method is not a very good method to estimate H, from the Matlab output we have:

Table 6.4.3

Wonderful:	Method 1	Method 2	Method 3
Indonesia	0.8925	0.9179	0.7812
Philippine	0.7724	0.8216	0.8402

For Indonesia and Philippine, all 3 methods indicate they have estimated Hurst parameter $H > 0.75$ by table 6.4.3.

Therefore, above countries can be considered as having $H > 0.75$ or close to 0.75 in their bond indexes while Indonesia and Philippine should be considered with strong certainty.

Specifically, if we use the average of estimated H from both estimation methods as the our estimated H, the distribution of estimated H of bond indexes for all the countries are:

Table 6.4.4

	0.75<=H<1
Spain	0.7734
Portugal	0.7967
Greece	0.7905
Romania	0.8002
Colombia	0.8216
Mexico	0.7599
Peru	0.7547

Indonesia	0.9052
Malaysia	0.7546
Philippine	0.7970
0.7<=H<0.75	
Italy	0.7196
Ireland	0.7320
Belgium	0.7423
Norway	0.7119
Egypt	0.7437
India	0.7094
0.65<=H<0.7	
Croatia	0.6592
Hungary	0.6556
Nigeria	0.6734
SEK Corp	0.6660
0.6<=H<0.65	
Slovenia	0.6457
Sweden	0.6262
Singapore	0.6102
Poland	0.6005
Dominican	0.6215
CHF Corp	0.6107
NOK Corp	0.6100
0.55<=H<0.6	
Eurozone	0.5808
France	0.5544
Austria	0.5863
UK	0.5620
Denmark	0.5881
New Zealand	0.5518
Loc EM	0.5732
Russia	0.5800
Turkey	0.5678
LATAM	0.5941
0.5<=H<0.55	
Canada	0.5135
Germany	0.5101
Netherland	0.5188
Luxembourg	0.5071
Finland	0.5072

Switzerland	0.5458
Japan	0.5140
EMEA Local	0.5085
Czech Republic	0.5009
Latvia	0.5228
South Africa	0.5332
Serbia	0.5224
Brazil	0.5044
South Korea	0.5207
Thailand	0.5466
CAD Corp	0.5042
GBP Corp	0.5100
0.45<=H<0.5	
US2	0.4697
US3	0.4678
US4	0.4773
Pacific	0.4716
Hong Kong	0.4743
Israel	0.4552
Kenya	0.4571
Chile	0.4883
Jamaica	0.4607
Asia	0.4516
USD Corp	0.4578
EUR Corp	0.4786
0.4<=H<0.45	
US1	0.4445
Slovakia	0.4472
Australia	0.4210
Bulgaria	0.4010
Lithuania	0.4232
Taiwan	0.4271
JPY Corp	0.4265
AUD Corp	0.4106
0<=H<0.3	
China	0.1522

What is more, since when H = 0.5, it is the Wiener process. Therefore when the estimated H is closed to 0.5, it could be considered as Wiener process. Then if we use the average of estimated H from both estimation methods as the our estimated H, the countries with estimated H closed to 0.5 are:

Table 6.4.5

H that closes to 0.5	
Canada	0.5135
US4	0.4773
Germany	0.5101
Netherland	0.5188
Luxembourg	0.5071
Finland	0.5072
Japan	0.5140
EMEA Local	0.5085
Czech Republic	0.5009
Latvia	0.5228
Serbia	0.5224
Brazil	0.5044
Chile	0.4883
South Korea	0.5207
CAD Corp	0.5042
EUR Corp	0.4786
GBP Corp	0.5100

Then we examine the average bond indexes in different categories, if we use the average of estimated H from both estimation methods as the our estimated H, the estimated H of them are:

Table 6.4.6

Estimated H:	Average	Standard Deviation
North American	0.4746	0.0250
Eurozone	0.6275	0.1212
Other European	0.6068	0.0661
Asia Pacific	0.5071	0.0670
Local Emerging	0.5842	0.1517

Corporate	0.5194	0.0896

We can see that for Asia Pacific bonds (only include developed countries), the average estimated Hurst parameter H are very close to 0.5 and have very low standard deviation, which means they close to Wiener process. This is also true for and North American bonds and Corporate bonds. However, both Euro-zone bonds and emerging market bonds have higher average estimated H which are close to 0.6.

Part III

Chapter 7

Conclusion

7.1 Conclusion of Part I

Base on the result of the back-testing we did in Chapter 3, our portfolio rebalancing method was shown performed the best. Therefore, this method makes its contribution by responding the market changes and it should be considered as a new method in portfolio rebalancing.

7.2 Conclusion of Part II

Base on the empirical results we got in Chapter 6, in equity market, the stock indexes of Vietnam, Slovenia, Malta, Ukraine, Nigeria, Kenya, Morocco, Qatar, Palestine could be treated as a standard semi-martingale with mixed financial model that includes both Brownian and fractional Brownian components in Melnikov & Mishura (2011)[1] while those of countries in North America and West Europe both could be considered as Wiener process. Stock indexes of developing countries have higher estimated H comparing to those of developed countries.

In foreign exchange market, there is no country's currency could be treated as a standard semi-martingale with mixed financial model that includes both Brownian and fractional Brownian components in Melnikov & Mishura (2011)[1] but exchange rates for G10, Asian, East Europe, African, and Latin American countries all could be considered as Wiener process.

In fixed income market, the bond indexes of Portugal, Greece, Colombia, Mexico, Indonesia, Philippine could be treated as a standard semi-martingale with mixed financial model that includes both Brownian and fractional Brownian components in Melnikov & Mishura (2011)[1] while those of the developed countries in Asia Pacific region, the countries in North America and Corporate bond indexes could be considered as Wiener process.

Bibliography

(1) William F. Sharpe (2010). Adaptive Asset Allocation Policies. *Financial Analysts Journal,* Vol. 66, Number 3, May/June 2010: pp. 45-59.

(2) Andre F. Perold and William F. Sharpe (1988). Dynamic Strategies for Asset Allocation. *Financial Analysts Journal*, January/February 1988: pp. 16-27.

(3) Christopher Donohue and Kenneth Yip (2003). Optimal Portfolio Rebalancing with Transaction Costs. *Journal of Portfolio Management*, Summer 2003, Vol. 29, No. 4: pp. 49-63.

(4) Martin Leibowitz and Anthony Bova (2011). Portfolio Strategy - Slow Rebalancing: A 50-Year History. *Morgan Stanley Research*, January 19, 2011.

(5) Hayne E. Leland (1996). Optimal Asset Rebalancing in the Presence of Transactions Costs. *IBER Working Paper* RPF-261, August 1996.

(6) Sun, Fan, Chen, Schouwenaars, and Albota (2004). Optimal Rebalancing Strategy Using Dynamic Programming for Institutional Portfolios. *SSRN Electronic Journal*, 12/2004; DOI: 10.2139/ssrn.639284. (Revised version pubished at *Journal of Portfolio Management*, Winter 2006)

(7) Kritzman, Myrgren, and Page (2009). Optimal Rebalancing: A Scalable Solution. *Journal Of Investment Management*, First Quarter 2009.

(8) Seth J. Masters (2003). Rebalancing. *Journal of Portfolio Management*, Spring 2003, Vol. 29, No. 3: pp. 52-57.

(9) Ian Carmichael (2009). Rebalancing - A Comparison of Methods. *MLC Investments Limited Research Report*, May 2009

(10) A.N.Kolmogorov (1940). Wienerische Spiralen und einige undere interessante Kurven in Hilbertschen Raum. *Doklady Akademii Nauk*, Vol.26, 2, 115-118.

(11) H. Hurst (1951). Long-term storage capacity of reservoirs. *Transactions of American Society of Civil Engineers*, Vol.116, 770-808.

(12) Alexander Melnikov and Yuliya Mishura (2011). On pricing and hedging in financial markets with long-range dependence. *Math Finan Econ* (2011) 5:29-46.

(13) Jean-Marc Bardet, Gabriel Lang, Georges Oppenheim, Anne Philippe, Stilian Stoev, Murad S. Taqqu. Semi-Parametric Estimation of the Long-Range Dependence Parameter: A Survey. *Theory and Applications of Long-Range Dependence*,

pp.557-577.

(14) Jacques Istas and Gabriel Lang. Quadratic variations and estimation of the local Hölder index of a Gaussian process. *Ann. Inst. Poincaré*, 33, pp. 407–436.

(15) Patrick Flandrin. wavelet analysis and synthesis of fractional brownian motion. *ieee trans. on inf. th.*, 38, pp. 910–917.

Appendix 1. Sample Matlab Code and Output for Chapter 3

(For full Matlab codes, please contact Jiayin Kang: jiayin2@ualberta.ca)

```
%% (1) Traditional Rebalancing Method with Fixed Band
%% Benchmark Weight Parameter
% To store the benchmark weight for each client:
% Column: 1 - Equity, 2 - Fixed Income, 3 - Cash, 4 - PE, 5 - Infrastructure:
benchmark_A = [0.55 0.3 0.01 0.06 0.08];

benchmark_B = [0.25 0.3 0.05 0.16 0.24];

benchmark_C = [0.6 0.1 0.05 0.125 0.125];

benchmark_D1 = [0.5 0.4 0.05 0.03 0.02];
benchmark_D2 = [0.5 0.4 0.05 0.03 0.02];


%% Starting Weight & Market Value
% To input the starting weight and starting market value for each client:
% Column: 1 - Equity, 2 - Fixed Income, 3 - Cash, 4 - PE, 5 - Infrastructure:
start_A = [0.51 0.29 0.02 0.08 0.1];
marketvalue_start_A = 200000000;

start_B = [0.3 0.4 0.02 0.1 0.18];
marketvalue_start_B = 100000000;

start_C = [0.62 0.07 0.03 0.13 0.15];
marketvalue_start_C = 10000000;

start_D1 = [0.53 0.37 0.05 0.02 0.03];
marketvalue_start_D1 = 10000000;

start_D2 = [0.51 0.39 0.01 0.06 0.03];
marketvalue_start_D2 = 3000000;


%% Cost Parameter
% To store the transaction cost parameters of each asset for both external
% and internal:

equity_transaction_percentage = 0.001;
fixedincome_transaction_percentage = 0.0005;
```

```
cash_transaction_percentage = 0;

PE_transaction_percentage = equity_transaction_percentage;
infrastruture_transaction_percentage_A =
0.5*equity_transaction_percentage +
0.5*fixedincome_transaction_percentage;
infrastruture_transaction_percentage_B =
0.75*equity_transaction_percentage +
0.25*fixedincome_transaction_percentage;
infrastruture_transaction_percentage_C =
0.75*equity_transaction_percentage +
0.25*fixedincome_transaction_percentage;
infrastruture_transaction_percentage_D1 =
0.5*equity_transaction_percentage +
0.5*fixedincome_transaction_percentage;
infrastruture_transaction_percentage_D2 =
0.5*equity_transaction_percentage +
0.5*fixedincome_transaction_percentage;

internal_transaction_percentage = 0.0001;

internalcost_limitnumber = 30;


%% Import Data
% We first change the data into 'csv' form and put them in "Data" file.
% Since the date data in the 1st column is not readable, we change them
% into number form and store in the 3rd column.
% To import the data from "Data" for future use:

equity_data = importdata('C:\Users\Alan Kang\Desktop\Paper
Writing\Rebalancing Codes\Data1\MSCI World.csv');
bond_data = importdata('C:\Users\Alan Kang\Desktop\Paper
Writing\Rebalancing Codes\Data1\Barc Agg Bond Index.csv');
cash_data = importdata('C:\Users\Alan Kang\Desktop\Paper
Writing\Rebalancing Codes\Data1\T-Bill.csv');

% Since we cannot change the date data (in 1st column) into proper form,
we
% change them into number in 3rd column and use them as proxy for date
% here:
equity_date = equity_data(:,3);
equity_price = equity_data(:,2);
```

```
bond_date_old = bond_data(:, 3);
bond_price_old = bond_data(:, 2);

cash_date_old = cash_data(:, 3);
cash_price_old = cash_data(:, 2);


%% Get the number of days for data
% To calculate the number of days in data file:
d = length(equity_date);


%% Sort Data
% Since bond and cash data are missing for many days, we sort the data
% according to equity data and fill missing parts with previous ones.

[bond_date bond_price] = BA_sort_data(equity_date, equity_price,
bond_date_old, bond_price_old);
[cash_date cash_price] = BA_sort_data(equity_date, equity_price,
cash_date_old, cash_price_old);


%% Private Equity & Infrastructure
% For illiquid assets, we use their banking system (base on eqiuty and
bond
% prices) to calculate their prices:

PE_date = equity_date;
PE_price = equity_price;

infrastructure_date = equity_date;
infrastructure_price_A = (equity_price + bond_price)/2;
infrastructure_price_B = (3*equity_price + bond_price)/4;
infrastructure_price_C = (3*equity_price + bond_price)/4;
infrastructure_price_D1 = (equity_price + bond_price)/2;
infrastructure_price_D2 = (equity_price + bond_price)/2;


%% Rebalancing Record
% To record the amounts of rebalancing of each asset in market value for
% each client at each day:
% 0: no trade, x: buy x market value, -x: sell x market value
rebalancing_A = zeros(d, 5);
rebalancing_B = zeros(d, 5);
```

```
rebalancing_C = zeros(d, 5);
rebalancing_D1 = zeros(d, 5);
rebalancing_D2 = zeros(d, 5);
% To record the total amounts of rebalancing of each asset in market value
% at each day:
rebalancing_total = zeros(d, 5);


%% Transaction Cost
% To record the total transaction cost at each day:
transaction_cost_total = zeros(d, 1);

% To record the total amounts of transaction of each asset (in internal
and
% external) for each client at each day:
% First 5 columns are internal transactions; Second 5 columns are external
% transactions.
transaction_A = zeros(d, 10);
transaction_B = zeros(d, 10);
transaction_C = zeros(d, 10);
transaction_D1 = zeros(d, 10);
transaction_D2 = zeros(d, 10);

% To record the transaction cost of each asset for each client each day:
transaction_cost_A = zeros(d, 5);
transaction_cost_B = zeros(d, 5);
transaction_cost_C = zeros(d, 5);
transaction_cost_D1 = zeros(d, 5);
transaction_cost_D2 = zeros(d, 5);

% To record the current number of internal trades.
internal_number = zeros(d, 1);

% Since the first 30 internal transactions are free, we record them in
% the credit account here. Then when calculating the total transaction
% costs, we deduct them from total cost:
trade_credit_A = zeros(d, 5);
trade_credit_B = zeros(d, 5);
trade_credit_C = zeros(d, 5);
trade_credit_D1 = zeros(d, 5);
trade_credit_D2 = zeros(d, 5);


%% Traditinal Rebalancing Method with Fixed Band, Fully Rebalance
```

```
% Set the target rebalancing band width percentage:
n1 = 15;
MyRecord = zeros(n1, 6);
x = 0;
for a1 = 1:n1
x = x + 1;
rebalancing_benchmark = a1/100;
% Set the level away from benchmark weight when we rebalance back to target
% benchmark weight (this is a parameter for not fully rebalancing use):
rebalancing_diff = 0;


%% Cash Requirement
% The index number for cash asset:
cash_mark = 3;
% To set the minimum percentage for cash value for each client:
cash_min = 0;


%% Rebalancing Test
% To record the market values and percentage weights of each asset for
each
% client at each day:
% First variable: market value
% Second variable: percentage inside this client
% Columns: 1 - Equity, 2 - Fixed Income, 3 - Cash, 4 - PE, 5 - Infrastructure
marketvalue_A = zeros(d, 5);
percentage_A = zeros(d, 5);

marketvalue_B = zeros(d, 5);
percentage_B = zeros(d, 5);

marketvalue_C = zeros(d, 5);
percentage_C = zeros(d, 5);

marketvalue_D1 = zeros(d, 5);
percentage_D1 = zeros(d, 5);

marketvalue_D2 = zeros(d, 5);
percentage_D2 = zeros(d, 5);

% To input the 1st day's market values and percentage weights of each asset
% for each client as our starting point:
marketvalue_A(1, :) = start_A(1, :)*marketvalue_start_A;
```

```
marketvalue_B(1, :) = start_B(1, :)*marketvalue_start_B;
marketvalue_C(1, :) = start_C(1, :)*marketvalue_start_C;
marketvalue_D1(1, :) = start_D1(1, :)*marketvalue_start_D1;
marketvalue_D2(1, :) = start_D2(1, :)*marketvalue_start_D2;

percentage_A(1, :) = start_A;
percentage_B(1, :) = start_B;
percentage_C(1, :) = start_C;
percentage_D1(1, :) = start_D1;
percentage_D2(1, :) = start_D2;

% Starting from the 2nd day, we do the rebalancing:
% We do the rebalancing at the beginning of each day base on previous day's
% close prices and information. Then we update at the end of that day base
% on the market changes this day in order to get the information for the
% rebalancing for the next day.
for i = 2:d
% To check whether this client's portfolio needs to be rebalanced:
% 'y_': 1 - need to rebalance, 0 - not need to
    y_A = BB2_check(percentage_A(i-1, :), benchmark_A,
rebalancing_benchmark);
    y_B = BB2_check(percentage_B(i-1, :), benchmark_B,
rebalancing_benchmark);
    y_C = BB2_check(percentage_C(i-1, :), benchmark_C,
rebalancing_benchmark);
    y_D1 = BB2_check(percentage_D1(i-1, :), benchmark_D1,
rebalancing_benchmark);
    y_D2 = BB2_check(percentage_D2(i-1, :), benchmark_D2,
rebalancing_benchmark);

% Rebalance until Satisfying Benchmark
% 'percentage_' records the portfolio weights that they should be
% 'rebalancing_' records the transaction that they should have

% For "BB_rebalancing_weight" function, every time we rebalance the asset
% with highest amount of weight difference away from benchmark. Then we
% check whether it satisfies the requirement or not. If not, we rebalance
% again base on this philosophy until it satisfies:
% After everything is done, we also record the market values for each asset
% after rebalancing.

% At the beginning, it is yesterday's wieght.
    percentage_A(i, :) = percentage_A(i-1, :);
    while y_A == 1
```

```
% When it is not OK, we rebalance 1 unsatisfied asset each time.
        [percentage_A(i, :), change] =
BB_rebalancing_weight(percentage_A(i, :), benchmark_A,
rebalancing_benchmark, rebalancing_diff, cash_mark, cash_min);
        rebalancing_A(i, :) = rebalancing_A(i, :) +
change.*sum(marketvalue_A(i-1, :));
        y_A = BB2_check(percentage_A(i, :), benchmark_A,
rebalancing_benchmark);
    end
    marketvalue_A(i, :) =
percentage_A(i, :).*sum(marketvalue_A(i-1, :));

% Similar logic is applied to client B, C, D1, D2:

    percentage_B(i, :) = percentage_B(i-1, :);
    while y_B == 1
        [percentage_B(i, :), change] =
BB_rebalancing_weight(percentage_B(i, :), benchmark_B,
rebalancing_benchmark, rebalancing_diff, cash_mark, cash_min);
        rebalancing_B(i, :) = rebalancing_B(i, :) +
change.*sum(marketvalue_B(i-1, :));
        y_B = BB2_check(percentage_B(i, :), benchmark_B,
rebalancing_benchmark);
    end
    marketvalue_B(i, :) =
percentage_B(i, :).*sum(marketvalue_B(i-1, :));

    percentage_C(i, :) = percentage_C(i-1, :);
    while y_C == 1
        [percentage_C(i, :), change] =
BB_rebalancing_weight(percentage_C(i, :), benchmark_C,
rebalancing_benchmark, rebalancing_diff, cash_mark, cash_min);
        rebalancing_C(i, :) = rebalancing_C(i, :) +
change.*sum(marketvalue_C(i-1, :));
        y_C = BB2_check(percentage_C(i, :), benchmark_C,
rebalancing_benchmark);
    end
    marketvalue_C(i, :) =
percentage_C(i, :).*sum(marketvalue_C(i-1, :));

    percentage_D1(i, :) = percentage_D1(i-1, :);
    while y_D1 == 1
        [percentage_D1(i, :), change] =
BB_rebalancing_weight(percentage_D1(i, :), benchmark_D1,
```

```
rebalancing_benchmark, rebalancing_diff, cash_mark, cash_min);
        rebalancing_D1(i, :) = rebalancing_D1(i, :) +
change.*sum(marketvalue_D1(i-1, :));
        y_D1 = BB2_check(percentage_D1(i, :), benchmark_D1,
rebalancing_benchmark);
    end
    marketvalue_D1(i, :) =
percentage_D1(i, :).*sum(marketvalue_D1(i-1, :));

    percentage_D2(i, :) = percentage_D2(i-1, :);
    while y_D2 == 1
        [percentage_D2(i, :), change] =
BB_rebalancing_weight(percentage_D2(i, :), benchmark_D2,
rebalancing_benchmark, rebalancing_diff, cash_mark, cash_min);
        rebalancing_D2(i, :) = rebalancing_D2(i, :) +
change.*sum(marketvalue_D2(i-1, :));
        y_D2 = BB2_check(percentage_D2(i, :), benchmark_D2,
rebalancing_benchmark);
    end
    marketvalue_D2(i, :) =
percentage_D2(i, :).*sum(marketvalue_D2(i-1, :));

% Fullfill Cash Requirement (make sure it satisfies 1% cash requirement)
% After rebalancing, we check whether the minimum cash requirement is
% fulfilled or not. Since we calculate the updated portfolio weights they
% should be, then calculate transaction costs to change from previous ones
% to these ones. To calculate in this way is OK.
    [percentage_A(i, :), marketvalue_A(i, :), rebalancing_A(i, :)] =
BB3_check_cash(percentage_A(i, :), benchmark_A, marketvalue_A(i, :),
rebalancing_A(i, :), cash_mark, cash_min);
    [percentage_B(i, :), marketvalue_B(i, :), rebalancing_B(i, :)] =
BB3_check_cash(percentage_B(i, :), benchmark_B, marketvalue_B(i, :),
rebalancing_B(i, :), cash_mark, cash_min);
    [percentage_C(i, :), marketvalue_C(i, :), rebalancing_C(i, :)] =
BB3_check_cash(percentage_C(i, :), benchmark_C, marketvalue_C(i, :),
rebalancing_C(i, :), cash_mark, cash_min);
    [percentage_D1(i, :), marketvalue_D1(i, :), rebalancing_D1(i, :)] =
BB3_check_cash(percentage_D1(i, :), benchmark_D1, marketvalue_D1(i, :),
rebalancing_D1(i, :), cash_mark, cash_min);
    [percentage_D2(i, :), marketvalue_D2(i, :), rebalancing_D2(i, :)] =
BB3_check_cash(percentage_D2(i, :), benchmark_D2, marketvalue_D2(i, :),
rebalancing_D2(i, :), cash_mark, cash_min);

% Calculate Total Transaction for Each Product Everyday
```

```
% The reason to calculate this is because we need to determine the net
% transaction for each asset. That is the amount which we have to buy
% externally. After getting this, we can calculate the minimum transaction
% cost from original portfolio weights to current ones, which should be
the
% transaction costs here.
    rebalancing_total(i, :) = rebalancing_A(i, :) + rebalancing_B(i, :)
+ rebalancing_C(i, :) + rebalancing_D1(i, :) + rebalancing_D2(i, :);

% Divide Rebalancing into Internal and External
% Here we extend the rebalancing data for each client into internal and
% external trades and calculate the trades eligible for free internal
% trades base on our 30 free internal trade condition and store them in
% 'credit_temp'.
    [transaction_A(i, :), transaction_B(i, :), transaction_C(i, :),
transaction_D1(i, :), transaction_D2(i, :), internal_number(i),
credit_temp] = BC_transaction_cost(rebalancing_total(i, :),
rebalancing_A(i, :), rebalancing_B(i, :), rebalancing_C(i, :),
rebalancing_D1(i, :), rebalancing_D2(i, :), internal_number(i-1),
internalcost_limitnumber);
    trade_credit_A(i, :) = credit_temp(1, :);
    trade_credit_B(i, :) = credit_temp(2, :);
    trade_credit_C(i, :) = credit_temp(3, :);
    trade_credit_D1(i, :) = credit_temp(4, :);
    trade_credit_D2(i, :) = credit_temp(5, :);

% Calculate Transaction Cost
% Base on internal trades, external trades, and free internal trades, we
% calculate the optimal transaction cost that should be.
    transaction_cost_A(i, :) = abs(transaction_A(i, 1:5) -
trade_credit_A(i, :)).*[internal_transaction_percentage
internal_transaction_percentage cash_transaction_percentage
internal_transaction_percentage internal_transaction_percentage] +
abs(transaction_A(i, 6:10)).*[equity_transaction_percentage
fixedincome_transaction_percentage cash_transaction_percentage
PE_transaction_percentage infrastruture_transaction_percentage_A];
    transaction_cost_B(i, :) = abs(transaction_B(i, 1:5) -
trade_credit_B(i, :)).*[internal_transaction_percentage
internal_transaction_percentage cash_transaction_percentage
internal_transaction_percentage internal_transaction_percentage] +
abs(transaction_B(i, 6:10)).*[equity_transaction_percentage
fixedincome_transaction_percentage cash_transaction_percentage
PE_transaction_percentage infrastruture_transaction_percentage_B];
    transaction_cost_C(i, :) = abs(transaction_C(i, 1:5) -
```

```
trade_credit_C(i, :)).*[internal_transaction_percentage
internal_transaction_percentage cash_transaction_percentage
internal_transaction_percentage internal_transaction_percentage] +
abs(transaction_C(i, 6:10)).*[equity_transaction_percentage
fixedincome_transaction_percentage cash_transaction_percentage
PE_transaction_percentage infrastruture_transaction_percentage_C];
   transaction_cost_D1(i, :) = abs(transaction_D1(i, 1:5) -
trade_credit_D1(i, :)).*[internal_transaction_percentage
internal_transaction_percentage cash_transaction_percentage
internal_transaction_percentage internal_transaction_percentage] +
abs(transaction_D1(i, 6:10)).*[equity_transaction_percentage
fixedincome_transaction_percentage cash_transaction_percentage
PE_transaction_percentage infrastruture_transaction_percentage_D1];
   transaction_cost_D2(i, :) = abs(transaction_D2(i, 1:5) -
trade_credit_D2(i, :)).*[internal_transaction_percentage
internal_transaction_percentage cash_transaction_percentage
internal_transaction_percentage internal_transaction_percentage] +
abs(transaction_D2(i, 6:10)).*[equity_transaction_percentage
fixedincome_transaction_percentage cash_transaction_percentage
PE_transaction_percentage infrastruture_transaction_percentage_D2];

% Total Cost at Each Day
% Sum of the transaction cost for each client is the total cost this day.
   transaction_cost_total(i) = sum(transaction_cost_A(i, :)) +
sum(transaction_cost_B(i, :)) + sum(transaction_cost_C(i, :)) +
sum(transaction_cost_D1(i, :)) + sum(transaction_cost_D2(i, :));

% Update Market Value due to Market Changes
% Since we do all the transaction at the beginning of each day base on
% previous day's data, at the end of this day we need to update the new
% market values and portfolio weights for each client base on the changes
% in this day. Then the beginning of next day we can do next step base
on
% this information.
   marketvalue_A(i, :) =
marketvalue_A(i, :).*[equity_price(i)/equity_price(i-1)
bond_price(i)/bond_price(i-1) cash_price(i)/cash_price(i-1)
PE_price(i)/PE_price(i-1)
infrastructure_price_A(i)/infrastructure_price_A(i-1)];
   marketvalue_B(i, :) =
marketvalue_B(i, :).*[equity_price(i)/equity_price(i-1)
bond_price(i)/bond_price(i-1) cash_price(i)/cash_price(i-1)
PE_price(i)/PE_price(i-1)
infrastructure_price_B(i)/infrastructure_price_B(i-1)];
```

```
    marketvalue_C(i, :) =
marketvalue_C(i, :).*[equity_price(i)/equity_price(i-1)
bond_price(i)/bond_price(i-1) cash_price(i)/cash_price(i-1)
PE_price(i)/PE_price(i-1)
infrastructure_price_C(i)/infrastructure_price_C(i-1)];
    marketvalue_D1(i, :) =
marketvalue_D1(i, :).*[equity_price(i)/equity_price(i-1)
bond_price(i)/bond_price(i-1) cash_price(i)/cash_price(i-1)
PE_price(i)/PE_price(i-1)
infrastructure_price_D1(i)/infrastructure_price_D1(i-1)];
    marketvalue_D2(i, :) =
marketvalue_D2(i, :).*[equity_price(i)/equity_price(i-1)
bond_price(i)/bond_price(i-1) cash_price(i)/cash_price(i-1)
PE_price(i)/PE_price(i-1)
infrastructure_price_D2(i)/infrastructure_price_D2(i-1)];

    percentage_A(i, :) = [marketvalue_A(i, 1) marketvalue_A(i, 2)
marketvalue_A(i, 3) marketvalue_A(i, 4) marketvalue_A(i,
5)]/sum(marketvalue_A(i, :));
    percentage_B(i, :) = [marketvalue_B(i, 1) marketvalue_B(i, 2)
marketvalue_B(i, 3) marketvalue_B(i, 4) marketvalue_B(i,
5)]/sum(marketvalue_B(i, :));
    percentage_C(i, :) = [marketvalue_C(i, 1) marketvalue_C(i, 2)
marketvalue_C(i, 3) marketvalue_C(i, 4) marketvalue_C(i,
5)]/sum(marketvalue_C(i, :));
    percentage_D1(i, :) = [marketvalue_D1(i, 1) marketvalue_D1(i, 2)
marketvalue_D1(i, 3) marketvalue_D1(i, 4) marketvalue_D1(i,
5)]/sum(marketvalue_D1(i, :));
    percentage_D2(i, :) = [marketvalue_D2(i, 1) marketvalue_D2(i, 2)
marketvalue_D2(i, 3) marketvalue_D2(i, 4) marketvalue_D2(i,
5)]/sum(marketvalue_D2(i, :));

end


%% Transaction Costs
TransactionCost_A = sum(transaction_cost_A(:, 1)) +
sum(transaction_cost_A(:, 2)) + sum(transaction_cost_A(:, 3)) +
sum(transaction_cost_A(:, 4)) + sum(transaction_cost_A(:, 5));
TransactionCost_B = sum(transaction_cost_B(:, 1)) +
sum(transaction_cost_B(:, 2)) + sum(transaction_cost_B(:, 3)) +
sum(transaction_cost_B(:, 4)) + sum(transaction_cost_B(:, 5));
TransactionCost_C = sum(transaction_cost_C(:, 1)) +
sum(transaction_cost_C(:, 2)) + sum(transaction_cost_C(:, 3)) +
```

```
sum(transaction_cost_C(:, 4)) + sum(transaction_cost_C(:, 5));
TransactionCost_D1 = sum(transaction_cost_D1(:, 1)) +
sum(transaction_cost_D1(:, 2)) + sum(transaction_cost_D1(:, 3)) +
sum(transaction_cost_D1(:, 4)) + sum(transaction_cost_D1(:, 5));
TransactionCost_D2 = sum(transaction_cost_D2(:, 1)) +
sum(transaction_cost_D2(:, 2)) + sum(transaction_cost_D2(:, 3)) +
sum(transaction_cost_D2(:, 4)) + sum(transaction_cost_D2(:, 5));

TransactionCost_Total = TransactionCost_A + TransactionCost_B +
TransactionCost_C + TransactionCost_D1 + TransactionCost_D2;


%% Portfolio Final Returns
Return_Value_A = sum(marketvalue_A(d, :)) - sum(marketvalue_A(1, :)) -
TransactionCost_A;
Return_Value_B = sum(marketvalue_B(d, :)) - sum(marketvalue_B(1, :)) -
TransactionCost_B;
Return_Value_C = sum(marketvalue_C(d, :)) - sum(marketvalue_C(1, :)) -
TransactionCost_C;
Return_Value_D1 = sum(marketvalue_D1(d, :)) - sum(marketvalue_D1(1, :))
- TransactionCost_D1;
Return_Value_D2 = sum(marketvalue_D2(d, :)) - sum(marketvalue_D2(1, :))
- TransactionCost_D2;

Return_Value_Total = Return_Value_A + Return_Value_B + Return_Value_C +
Return_Value_D1 + Return_Value_D2;


Return_Percentage_A = (sum(marketvalue_A(d, :)) -
sum(marketvalue_A(1, :)) - TransactionCost_A)/sum(marketvalue_A(1, :));
Return_Percentage_B = (sum(marketvalue_B(d, :)) -
sum(marketvalue_B(1, :)) - TransactionCost_B)/sum(marketvalue_B(1, :));
Return_Percentage_C = (sum(marketvalue_C(d, :)) -
sum(marketvalue_C(1, :)) - TransactionCost_C)/sum(marketvalue_C(1, :));
Return_Percentage_D1 = (sum(marketvalue_D1(d, :)) -
sum(marketvalue_D1(1, :)) -
TransactionCost_D1)/sum(marketvalue_D1(1, :));
Return_Percentage_D2 = (sum(marketvalue_D2(d, :)) -
sum(marketvalue_D2(1, :)) -
TransactionCost_D2)/sum(marketvalue_D2(1, :));

Return_Percentage_Total = Return_Value_Total/(sum(marketvalue_A(1, :))
+ sum(marketvalue_B(1, :)) + sum(marketvalue_C(1, :)) +
sum(marketvalue_D1(1, :)) + sum(marketvalue_D2(1, :)));
```

```
%% Output Section
% Output the Transaction Cost
fprintf('The transaction cost for Client A is: ')
fprintf('\n')
TransactionCost_A
fprintf('\n')

fprintf('The transaction cost for Client B is: ')
fprintf('\n')
TransactionCost_B
fprintf('\n')

fprintf('The transaction cost for Client C is: ')
fprintf('\n')
TransactionCost_C
fprintf('\n')

fprintf('The transaction cost for Client D1 is: ')
fprintf('\n')
TransactionCost_D1
fprintf('\n')

fprintf('The transaction cost for Client D2 is: ')
fprintf('\n')
TransactionCost_D2
fprintf('\n')

fprintf('The total transaction cost is: ')
fprintf('\n')
TransactionCost_Total
fprintf('\n')

fprintf('\n')
fprintf('\n')


% Output the Portfolio Return
fprintf('The portfolio profit value for Client A is: ')
fprintf('\n')
Return_Value_A
fprintf('\n')
```

```
fprintf('The portfolio profit value for Client B is: ')
fprintf('\n')
Return_Value_B
fprintf('\n')

fprintf('The portfolio profit value for Client C is: ')
fprintf('\n')
Return_Value_C
fprintf('\n')

fprintf('The portfolio profit value for Client D1 is: ')
fprintf('\n')
Return_Value_D1
fprintf('\n')

fprintf('The portfolio profit value for Client D2 is: ')
fprintf('\n')
Return_Value_D2
fprintf('\n')

fprintf('The total portfolio profit value is: ')
fprintf('\n')
Return_Value_Total
fprintf('\n')

fprintf('\n')
fprintf('\n')


fprintf('The portfolio return for Client A is: ')
fprintf('\n')
Return_Percentage_A
fprintf('\n')

fprintf('The portfolio return for Client B is: ')
fprintf('\n')
Return_Percentage_B
fprintf('\n')

fprintf('The portfolio return for Client C is: ')
fprintf('\n')
Return_Percentage_C
fprintf('\n')
```

```
fprintf('The portfolio return for Client D1 is: ')
fprintf('\n')
Return_Percentage_D1
fprintf('\n')

fprintf('The portfolio return for Client D2 is: ')
fprintf('\n')
Return_Percentage_D2
fprintf('\n')

fprintf('The total portfolio return is: ')
fprintf('\n')
Return_Percentage_Total
fprintf('\n')

MyRecord(x, :) = [Return_Percentage_Total Return_Percentage_A
Return_Percentage_B Return_Percentage_C Return_Percentage_D1
Return_Percentage_D2];
end
```

Appendix 2. Sample Matlab Code and Output for Chapter 6

(For full Matlab codes, please contact Jiayin Kang: jiayin2@ualberta.ca)

```
%% Set number of indexes:
N = 108;
Data = cell(N, 3);

%% Input data one by one:
% North American Stock Index:
Data{1, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\Canada.csv';
Data{2, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\USA1.csv';
Data{3, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\USA2.csv';
Data{4, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\USA3.csv';
Data{5, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\USA4.csv';
Data{6, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\USA5.csv';

% South American Stock Index:
Data{7, 2} = 'C:\Users\Alan Kang\Documents\MATLAB\fBm\Data\CSV1\Costa
Rica.csv';
Data{8, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\Jamaica.csv';
Data{9, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\Mexico.csv';
Data{10, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\Panama.csv';
Data{11, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\Peru.csv';
Data{12, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\Argentina.csv';
Data{13, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\Bermuda.csv';
Data{14, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\Brazil.csv';
Data{15, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\Chile.csv';
Data{16, 2} = 'C:\Users\Alan
```

```
Kang\Documents\MATLAB\fBm\Data\CSV1\Colombia.csv';
Data{17, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV1\Venezuela.csv';

% West European Stock Index:
Data{18, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Bloomberg Euro.csv';
Data{19, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Eurotop 100.csv';
Data{20, 2} = 'C:\Users\Alan Kang\Documents\MATLAB\fBm\Data\CSV2\EUROPE
600.csv';
Data{21, 2} = 'C:\Users\Alan Kang\Documents\MATLAB\fBm\Data\CSV2\Europe
350.csv';
Data{22, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\UK1.csv';
Data{23, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\UK2.csv';
Data{24, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\UK3.csv';
Data{25, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Germany.csv';
Data{26, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\France1.csv';
Data{27, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\France2.csv';
Data{28, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Spain.csv';
Data{29, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Switzerland.csv';
Data{30, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Italy.csv';
Data{31, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Portugal.csv';
Data{32, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Ireland.csv';
Data{33, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Iceland.csv';
Data{34, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Netherland.csv';
Data{35, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Belgium.csv';
Data{36, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Luxembourg.csv';
Data{37, 2} = 'C:\Users\Alan
```

```
Kang\Documents\MATLAB\fBm\Data\CSV2\Denmark.csv';
Data{38, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Finland.csv';
Data{39, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Norway.csv';
Data{40, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Sweden.csv';
Data{41, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Austria.csv';
Data{42, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Greece.csv';
Data{43, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV2\Poland.csv';
Data{44, 2} = 'C:\Users\Alan Kang\Documents\MATLAB\fBm\Data\CSV2\Czech
Republic.csv';

% Asian Stock Index:
Data{45, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\China3.csv';
Data{46, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\Taiwan.csv';
Data{47, 2} = 'C:\Users\Alan Kang\Documents\MATLAB\fBm\Data\CSV3\South
Korea.csv';
Data{48, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\Australia.csv';
Data{49, 2} = 'C:\Users\Alan Kang\Documents\MATLAB\fBm\Data\CSV3\New
Zealand.csv';
Data{50, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\Pakistan.csv';
Data{51, 2} = 'C:\Users\Alan Kang\Documents\MATLAB\fBm\Data\CSV3\Sri
Lanka.csv';
Data{52, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\Thailand.csv';
Data{53, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\Indonesia.csv';
Data{54, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\India.csv';
Data{55, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\Singapore.csv';
Data{56, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\Malaysia.csv';
Data{57, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\Philippines.csv';
Data{58, 2} = 'C:\Users\Alan
```

```
Kang\Documents\MATLAB\fBm\Data\CSV3\Vietnam.csv';
Data{59, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\Bangladesh.csv';
Data{60, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\Mongolia.csv';
Data{61, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\Laos.csv';
Data{62, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\Japan1.csv';
Data{63, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\Japan2.csv';
Data{64, 2} = 'C:\Users\Alan Kang\Documents\MATLAB\fBm\Data\CSV3\Hong
Kong.csv';
Data{65, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\China1.csv';
Data{66, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV3\China2.csv';

% East European Stock Index:
Data{67, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Slovakia.csv';
Data{68, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Croatia.csv';
Data{69, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Slovenia.csv';
Data{70, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Bosnia.csv';
Data{71, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Serbia.csv';
Data{72, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Montenegro.csv';
Data{73, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Estonia.csv';
Data{74, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Macedonia.csv';
Data{75, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Latvia.csv';
Data{76, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Lithuania.csv';
Data{77, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Bulgaria.csv';
Data{78, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Turkey.csv';
Data{79, 2} = 'C:\Users\Alan
```

```
Kang\Documents\MATLAB\fBm\Data\CSV4\Cyprus.csv';
Data{80, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Malta.csv';
Data{81, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Russia1.csv';
Data{82, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Russia2.csv';
Data{83, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Hungary.csv';
Data{84, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Romania.csv';
Data{85, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Ukraine.csv';
Data{86, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV4\Kazakhstan.csv';

% African Stock Index:
Data{87, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV5\Botswana.csv';
Data{88, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV5\Nigeria.csv';
Data{89, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV5\Tanzania.csv';
Data{90, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV5\Kenya.csv';
Data{91, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV5\Ghana.csv';
Data{92, 2} = 'C:\Users\Alan Kang\Documents\MATLAB\fBm\Data\CSV5\South
Africa1.csv';
Data{93, 2} = 'C:\Users\Alan Kang\Documents\MATLAB\fBm\Data\CSV5\South
Africa2.csv';
Data{94, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV5\Egypt.csv';
Data{95, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV5\Morocco.csv';
Data{96, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV5\Tunisia.csv';
Data{97, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV5\Namibia.csv';

% Middle East Stock Index:
Data{98, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV6\Jordan.csv';
Data{99, 2} = 'C:\Users\Alan
```

```
Kang\Documents\MATLAB\fBm\Data\CSV6\Oman.csv';
Data{100, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV6\Qatar.csv';
Data{101, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV6\United Arab.csv';
Data{102, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV6\Mauritius.csv';
Data{103, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV6\Kuwait.csv';
Data{104, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV6\Israel.csv';
Data{105, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV6\Palestine.csv';
Data{106, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV6\Lebanon.csv';
Data{107, 2} = 'C:\Users\Alan
Kang\Documents\MATLAB\fBm\Data\CSV6\Bahrain.csv';
Data{108, 2} = 'C:\Users\Alan Kang\Documents\MATLAB\fBm\Data\CSV6\Saudi
Arabia.csv';


for i = 1:N
    Data{i, 1} = uigetfile(Data{i, 2});
    Data{i, 3} = csvread(Data{i, 2}, 2, 1);
end

%% Estimate the H parameter:
Hest = zeros(N ,3);
for i = 1:N
    Hest(i,:) = wfbmesti(Data{i, 3});
end


%% Calculate our Chosen Estimated H:
Hest_chosen =  (Hest(:, 1) + Hest(:, 2))/2;


%% Pick up the indexes with H estimation in specific ranges:

% Ranges:
M = 12;

M1 = 1;
```

```
M2 = 0.75;
M3 = 0.7;
M4 = 0.65;
M5 = 0.6;
M6 = 0.55;
M7 = 0.5;
M8 = 0.45;
M9 = 0.4;
M10 = 0.35;
M11 = 0.3;
M12 = 0;

K1 = 0.475;
K2 = 0.525;


% Find out H that 0.75<=H<1:
[index H_value] = AB_filter(Hest_chosen, M2, M1);
if index(1) == 0
    H_estimated1 = cell(1, 2);
    H_estimated1{1, 1} = 'Null';
    H_estimated1{1, 2} = 0;
else
    d = length(index);
    H_estimated1 = cell(d, 2);
    for i = 1:d
        H_estimated1{i, 1} = Data{index(i), 1};
        H_estimated1{i, 2} = H_value(i);
    end
end

% Find out H that 0.7<=H<0.75:
[index H_value] = AB_filter(Hest_chosen, M3, M2);
if index(1) == 0
    H_estimated2 = cell(1, 2);
    H_estimated2{1, 1} = 'Null';
    H_estimated2{1, 2} = 0;
else
    d = length(index);
    H_estimated2 = cell(d, 2);
    for i = 1:d
        H_estimated2{i, 1} = Data{index(i), 1};
        H_estimated2{i, 2} = H_value(i);
    end
```

```
end

% Find out H that 0.65<=H<0.7:
[index H_value] = AB_filter(Hest_chosen, M4, M3);
if index(1) == 0
    H_estimated3 = cell(1, 2);
    H_estimated3{1, 1} = 'Null';
    H_estimated3{1, 2} = 0;
else
    d = length(index);
    H_estimated3 = cell(d, 2);
    for i = 1:d
        H_estimated3{i, 1} = Data{index(i), 1};
        H_estimated3{i, 2} = H_value(i);
    end
end

% Find out H that 0.6<=H<0.65:
[index H_value] = AB_filter(Hest_chosen, M5, M4);
if index(1) == 0
    H_estimated4 = cell(1, 2);
    H_estimated4{1, 1} = 'Null';
    H_estimated4{1, 2} = 0;
else
    d = length(index);
    H_estimated4 = cell(d, 2);
    for i = 1:d
        H_estimated4{i, 1} = Data{index(i), 1};
        H_estimated4{i, 2} = H_value(i);
    end
end

% Find out H that 0.55<=H<0.6:
[index H_value] = AB_filter(Hest_chosen, M6, M5);
if index(1) == 0
    H_estimated5 = cell(1, 2);
    H_estimated5{1, 1} = 'Null';
    H_estimated5{1, 2} = 0;
else
    d = length(index);
    H_estimated5 = cell(d, 2);
    for i = 1:d
        H_estimated5{i, 1} = Data{index(i), 1};
        H_estimated5{i, 2} = H_value(i);
```

```
    end
end

% Find out H that 0.5<=H<0.55:
[index H_value] = AB_filter(Hest_chosen, M7, M6);
if index(1) == 0
    H_estimated6 = cell(1, 2);
    H_estimated6{1, 1} = 'Null';
    H_estimated6{1, 2} = 0;
else
    d = length(index);
    H_estimated6 = cell(d, 2);
    for i = 1:d
        H_estimated6{i, 1} = Data{index(i), 1};
        H_estimated6{i, 2} = H_value(i);
    end
end

% Find out H that 0.45<=H<0.5:
[index H_value] = AB_filter(Hest_chosen, M8, M7);
if index(1) == 0
    H_estimated7 = cell(1, 2);
    H_estimated7{1, 1} = 'Null';
    H_estimated7{1, 2} = 0;
else
    d = length(index);
    H_estimated7 = cell(d, 2);
    for i = 1:d
        H_estimated7{i, 1} = Data{index(i), 1};
        H_estimated7{i, 2} = H_value(i);
    end
end

% Find out H that 0.4<=H<0.45:
[index H_value] = AB_filter(Hest_chosen, M9, M8);
if index(1) == 0
    H_estimated8 = cell(1, 2);
    H_estimated8{1, 1} = 'Null';
    H_estimated8{1, 2} = 0;
else
    d = length(index);
    H_estimated8 = cell(d, 2);
    for i = 1:d
        H_estimated8{i, 1} = Data{index(i), 1};
```

```
        H_estimated8{i, 2} = H_value(i);
    end
end

% Find out H that 0.35<=H<0.4:
[index H_value] = AB_filter(Hest_chosen, M10, M9);
if index(1) == 0
    H_estimated9 = cell(1, 2);
    H_estimated9{1, 1} = 'Null';
    H_estimated9{1, 2} = 0;
else
    d = length(index);
    H_estimated9 = cell(d, 2);
    for i = 1:d
        H_estimated9{i, 1} = Data{index(i), 1};
        H_estimated9{i, 2} = H_value(i);
    end
end

% Find out H that 0.3<=H<0.35:
[index H_value] = AB_filter(Hest_chosen, M11, M10);
if index(1) == 0
    H_estimated10 = cell(1, 2);
    H_estimated10{1, 1} = 'Null';
    H_estimated10{1, 2} = 0;
else
    d = length(index);
    H_estimated10 = cell(d, 2);
    for i = 1:d
        H_estimated10{i, 1} = Data{index(i), 1};
        H_estimated10{i, 2} = H_value(i);
    end
end

% Find out H that 0<=H<0.3:
[index H_value] = AB_filter(Hest_chosen, M12, M11);
if index(1) == 0
    H_estimated11 = cell(1, 2);
    H_estimated11{1, 1} = 'Null';
    H_estimated11{1, 2} = 0;
else
    d = length(index);
    H_estimated11 = cell(d, 2);
    for i = 1:d
```

```
        H_estimated11{i, 1} = Data{index(i), 1};
        H_estimated11{i, 2} = H_value(i);
    end
end


% Find out H that closes to 0.5:
[index H_value] = AB_filter(Hest_chosen, K1, K2);
if index(1) == 0
    H_estimated_K = cell(1, 2);
    H_estimated_K{1, 1} = 'Null';
    H_estimated_K{1, 2} = 0;
else
    d = length(index);
    H_estimated_K = cell(d, 2);
    for i = 1:d
        H_estimated_K{i, 1} = Data{index(i), 1};
        H_estimated_K{i, 2} = H_value(i);
    end
end


%% Divide Data by Region
N1 = 6;
N2 = 17;
N3 = 44;
N4 = 66;
N5 = 86;
N6 = 97;
N7 = 108;


%% Estimate Average H base on regions:

first1 = mean(Hest_chosen(1:N1, 1));
first2 = mean(Hest_chosen((N1+1):N2, 1));
first3 = mean(Hest_chosen((N2+1):N3, 1));
first4 = mean(Hest_chosen((N3+1):N4, 1));
first5 = mean(Hest_chosen((N4+1):N5, 1));
first6 = mean(Hest_chosen((N5+1):N6, 1));
first7 = mean(Hest_chosen((N6+1):N7, 1));

first_sd1 = std(Hest_chosen(1:N1, 1));
first_sd2 = std(Hest_chosen((N1+1):N2, 1));
```

```
first_sd3 = std(Hest_chosen((N2+1):N3, 1));
first_sd4 = std(Hest_chosen((N3+1):N4, 1));
first_sd5 = std(Hest_chosen((N4+1):N5, 1));
first_sd6 = std(Hest_chosen((N5+1):N6, 1));
first_sd7 = std(Hest_chosen((N6+1):N7, 1));


%% Output Section
% Output estimators in specific ranges:
if H_estimated1{1, 2} == 0
    fprintf('There is no H that 0.75<=H<1.\n')
else
    fprintf('The estimated H that 0.75<=H<1 are: ')
    H_estimated1
    fprintf('\n')
end
fprintf('\n')
fprintf('\n')

if H_estimated2{1, 2} == 0
    fprintf('There is no H that 0.7<=H<0.75.\n')
else
    fprintf('The estimated H that 0.7<=H<0.75 are: ')
    H_estimated2
    fprintf('\n')
end
fprintf('\n')
fprintf('\n')

if H_estimated3{1, 2} == 0
    fprintf('There is no H that 0.65<=H<0.7.\n')
else
    fprintf('The estimated H that 0.65<=H<0.7 are: ')
    H_estimated3
    fprintf('\n')
end
fprintf('\n')
fprintf('\n')

if H_estimated4{1, 2} == 0
    fprintf('There is no H that 0.6<=H<0.65.\n')
else
    fprintf('The estimated H that 0.6<=H<0.65 are: ')
    H_estimated4
```

```
    fprintf('\n')
end
fprintf('\n')
fprintf('\n')

if H_estimated5{1, 2} == 0
    fprintf('There is no H that 0.55<=H<0.6.\n')
else
    fprintf('The estimated H that 0.55<=H<0.6 are: ')
    H_estimated5
    fprintf('\n')
end
fprintf('\n')
fprintf('\n')

if H_estimated6{1, 2} == 0
    fprintf('There is no H that 0.5<=H<0.55.\n')
else
    fprintf('The estimated H that 0.5<=H<0.55 are: ')
    H_estimated6
    fprintf('\n')
end
fprintf('\n')
fprintf('\n')

if H_estimated7{1, 2} == 0
    fprintf('There is no H that 0.45<=H<0.5.\n')
else
    fprintf('The estimated H that 0.45<=H<0.5 are: ')
    H_estimated7
    fprintf('\n')
end
fprintf('\n')
fprintf('\n')

if H_estimated8{1, 2} == 0
    fprintf('There is no H that 0.4<=H<0.45.\n')
else
    fprintf('The estimated H that 0.4<=H<0.45 are: ')
    H_estimated8
    fprintf('\n')
end
fprintf('\n')
fprintf('\n')
```

```
if H_estimated9{1, 2} == 0
    fprintf('There is no H that 0.35<=H<0.4.\n')
else
    fprintf('The estimated H that 0.35<=H<0.4 are: ')
    H_estimated9
    fprintf('\n')
end
fprintf('\n')
fprintf('\n')

if H_estimated10{1, 2} == 0
    fprintf('There is no H that 0.3<=H<0.35.\n')
else
    fprintf('The estimated H that 0.3<=H<0.35 are: ')
    H_estimated10
    fprintf('\n')
end
fprintf('\n')
fprintf('\n')

if H_estimated11{1, 2} == 0
    fprintf('There is no H that 0<=H<0.3.\n')
else
    fprintf('The estimated H that 0<=H<0.3 are: ')
    H_estimated11
    fprintf('\n')
end
fprintf('\n')
fprintf('\n')


% Output H that closes to 0.5:
if H_estimated_K{1, 2} == 0
    fprintf('There is no H that closes to 0.5.\n')
else
    fprintf('The H that closes to 0.5 are: ')
    H_estimated_K
    fprintf('\n')
end
fprintf('\n')
fprintf('\n')
```

```
% Output North American Estimated H:
fprintf('The Average Estimated H in North America are: ')
fprintf('\n')
first1
first_sd1
fprintf('\n')
fprintf('\n')

% Output South American Estimated H:
fprintf('The Average Estimated H in South America are: ')
fprintf('\n')
first2
first_sd2
fprintf('\n')
fprintf('\n')

% Output West European Estimated H:
fprintf('The Average Estimated H in West Europe are: ')
fprintf('\n')
first3
first_sd3
fprintf('\n')
fprintf('\n')

% Output Asian Estimated H:
fprintf('The Average Estimated H in Asia are: ')
fprintf('\n')
first4
first_sd4
fprintf('\n')
fprintf('\n')

% Output East European Estimated H:
fprintf('The Average Estimated H in East Europe are: ')
fprintf('\n')
first5
first_sd5
fprintf('\n')
fprintf('\n')

% Output African Estimated H:
fprintf('The Average Estimated H in Africa are: ')
fprintf('\n')
first6
```

```
first_sd6
fprintf('\n')
fprintf('\n')

% Output Middle East Estimated H:
fprintf('The Average Estimated H in Middle East are: ')
fprintf('\n')
first7
first_sd7
fprintf('\n')
fprintf('\n')
```