



**DIMENSION REDUCTION METHODS FOR
NON-STATIONARY MULTIVARIATE TIME SERIES**

Fayed A. Alshammri

Department of Mathematics and Statistics

University of Strathclyde

Glasgow

August 21, 2020

Acknowledgement

First, I would like to express my thanks and great gratitude to Allah who gave me the ability to carry out and complete my research.

Second, I am thankful to the government of Saudi Arabia represented by the Ministry of Higher Education and the Saudi Electronic University for their scholarship program and endless support.

A special thanks to my supervisor Dr Jiazhu Pan for his support and guidance in every aspect during my studies at the University of Strathclyde.

Also, I would like to extend my thanks to my parents, brothers and sisters for their encouragement to complete my studies. A special thank goes to my wife Fatimah and my son Abdulaziz for their love, support and being great companions during my time in the United Kingdom.

Abstract

In this thesis, we extend the principal component analysis (PCA) to account for both stationary and non-stationary time series data. The dimension reduction methods we propose will employ a moving cross-covariance matrix of data, which can be updated as we move in time. We show that the moving cross-covariance matrix can extract dynamic dependence among variables of both stationary and non-stationary series.

The first two methods we propose can be considered as a generalization of dynamic principal component analysis (DPCA) of Ku et al. (1995) to the non-stationary case. The first method will apply eigenanalysis on the moving cross-covariance matrix of the extended data vector that can be formed by including lagged series into the original data vector. The second method is different from the first one, where we apply eigenanalysis on a quadratic-order of the moving cross-covariance matrix of the extended data vector. In order to optimize the results of our methods, we will propose a new criterion to determine the optimal number of principal components to retain. Additionally, we are going to introduce the moving cross-correlation function that can be used to evaluate the correlations between non-stationary variables. The third dimension reduction method that we introduce will generalize the principal component analysis for time series (TS-PCA) of Chang et al. (2018) to non-stationary series. This method seeks a linear transformation such that the transformed series is segmented into uncorrelated subseries with lower dimensions that can be separately analysed as they are not correlated. The latter method will account for high-dimensional time series.

Theoretical properties of the proposed methods show the consistency of the used estimators. All methods prove their abilities to dimension reduction of both stationary and non-stationary series based on simulated and real data sets.

Contents

1	Introduction	1
1.1	Background	1
1.2	Overview of the Study	4
2	Preliminaries	6
2.1	Multivariate Time Series	6
2.2	Stationarity of a Multivariate Time Series	8
2.3	Principal Component Analysis	11
3	Moving Dynamic Principal Component Analysis	16
3.1	Methodology	16
3.1.1	Optimizing MDPCA's Results	21
3.1.1.1	Choosing Optimal Number of Lags	22
3.1.1.2	Retained Component Criterion (RCC)	25
3.1.2	MDPCA Calculation Procedure	27
3.1.3	Evaluating Dynamic Relationships among MDPCs	28
3.2	Theoretical Properties	36
3.3	Simulations and Real Data Examples	40
3.3.1	Simulations	40
3.3.2	Real Data Examples	81
3.4	Summary and Remarks	90
4	Quadratic Moving Dynamic Principal Component Analysis	92
4.1	Methodology	92
4.1.1	Optimizing the Results of QMDPCA	95

4.1.2	QMDPCA Calculation Procedure	98
4.1.3	Evaluating Dynamic Relationships	98
4.2	Theoretical Properties	99
4.3	Simulations and Real Data Examples	101
4.3.1	Simulations	102
4.3.2	Real Data Examples	152
4.4	Summary and Remarks	160
5	Generalized Principal Component Analysis for Time Series	163
5.1	Methodology	163
5.2	Columns' Rearrangement	168
5.2.1	The Moving Cross-Correlation Function	169
5.2.2	The Maximum Moving Cross-Correlation Method	170
5.3	Theoretical Properties	173
5.3.1	Asymptotic Results (m is fixed)	174
5.3.2	Asymptotic Results ($m = o(W^c)$)	176
5.3.3	Asymptotic Results ($\log m = o(W^c)$)	181
5.4	Simulations and Real Data Examples	183
5.4.1	Simulations	183
5.4.2	Real Data Examples	201
5.5	Summary and Remarks	208
6	Conclusions and Further Work	210
6.1	Conclusions	210
6.2	Further Work	213
	Bibliography	214

List of Tables

2.1	The cross-correlation matrix of the nine variables in Example 2.3.1.	13
3.1	Correlated variables of the simulated data with eight variables in Example 3.1.2 based on different visualization methods	35
3.2	A comparison between the results of MDPCA with different combinations of W and l in Example 3.3.1.	42
3.3	Standard deviations of the absolute value of the moving cross-correlations between two MDPCs with different combinations of W and l in Example 3.3.1.	44
3.4	A comparison between the results of MDPCA with different sizes of T in Example 3.3.1.	46
3.5	Standard deviations of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.1.	48
3.6	A comparison between the results of MDPCA with different combinations of W and l in Example 3.3.2.	53
3.7	Standard deviations of the absolute value of the moving cross-correlations between three MDPCs with different combinations of W and l in Example 3.3.2 - Part 1.	55
3.8	Standard deviations of the absolute value of the moving cross-correlations between three MDPCs with different combinations of W and l in Example 3.3.2 - Part 2.	56
3.9	A comparison between the results of MDPCA with different sizes of T in Example 3.3.2.	57

3.10	Standard deviations of the absolute value of the moving cross-correlations between three MDPCs with different sizes of T in Example 3.3.2.	59
3.11	A comparison between the results of MDPCA with different combinations of W and l in Example 3.3.3.	64
3.12	Standard deviations of the absolute value of the moving cross-correlations between two MDPCs with different options of W and l in Example 3.3.3.	66
3.13	A comparison between the results of MDPCA with different sizes of T in Example 3.3.3.	68
3.14	Standard deviations of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.3.	70
3.15	A comparison between the results of MDPCA with different combinations of W and l in Example 3.3.4.	72
3.16	Standard deviations of the absolute value of the moving cross-correlations between two MDPCs with different combinations of W and l in Example 3.3.4.	75
3.17	A comparison between the results of MDPCA with different sizes of T in Example 3.3.4.	77
3.18	Standard deviations of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.4.	79
3.19	Names and labels of the 17 US companies in Example 3.3.6.	86
4.1	A comparison between QMDPCA and DPCA with different sizes of l applied to the simulated stationary series in Example 4.3.1.	104
4.2	Standard deviations of the absolute value of the acf function between two components with different sizes of l in Example 4.3.1.	106
4.3	Comparison between the results of QMDPCA with different sizes of T in Example 4.3.1.	107
4.4	Standard deviations of the absolute value of the acf function between two QMDPCs with different sizes of T in Example 4.3.1.	109
4.5	A comparison between QMDPCA and MDPCA with $W = 101$ and different sizes of l in Example 4.3.2 where two components are retained.	114

4.6	Standard deviations of the absolute value of the moving cross-correlation between two components with $W = 101$ and different sizes of l in Example 4.3.2.	116
4.7	A comparison between the results of QMDPCA with different combinations of W and l in Example 4.3.2.	118
4.8	Standard deviations of the absolute value of the moving cross-correlations between two components with different combinations of W and l in Example 4.3.2.	121
4.9	A comparison between the results of QMDPCA with different sizes of T in Example 4.3.2.	123
4.10	Standard deviations of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.2. . .	125
4.11	A comparison between the results of QMDPCA with different combinations of W and l in Example 4.3.3.	130
4.12	Standard deviations of the absolute value of the moving cross-correlations between two components with different combinations of W and l in Example 4.3.3.	133
4.13	A comparison between the results of QMDPCA with different sizes of T in Example 4.3.3.	134
4.14	Standard deviations of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.3. . .	137
4.15	A comparison between the results of QMDPCA with different combinations of W and l in Example 4.3.4.	143
4.16	Standard deviations of the absolute value of the moving cross-correlations between two QMDPCs with different combinations of W and l in Example 4.3.4.	146
4.17	A comparison between the results of QMDPCA with different sizes of T in Example 4.3.4.	148
4.18	Standard deviations of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.4. . .	150
4.19	Names and labels of the 18 US companies in Example 4.3.6.	156

5.1	Correlated variables of the simulated data of eight variables in Example 5.2.1 based on different methods.	172
5.2	Percentages of correct segmentation of 500 replications of the simulated series in Example 5.4.1.	184
5.3	Percentages of correct segmentation of 500 replications of the simulated series in Example 5.4.2.	188
5.4	Percentages of correct segmentation of 500 replications of the simulated series in Example 5.4.3.	194
5.5	Number of subseries after applying GTS-PCA and TS-PCA to the randomly selected replica in Example 5.4.4.	200
5.6	Names and labels of the 22 US companies in Example 5.4.6.	204
5.7	Summary results of the transformed segmentation of the 22 US companies in Example 5.4.6 after applying GTS-PCA with $W = 101$. . .	208

List of Figures

2.1	The daily stock prices of the companies Apple and Microsoft from November 2013 to December 2017.	7
2.2	Eigenvalues' plot after applying PCA on the nine variables in Example 2.3.1.	14
3.1	A time series plot of the simulated series with three variables in Example 3.1.1.	18
3.2	An eigenvalues' plot of the simulated series with three variables in Example 3.1.1 after applying DPCA $l = 1$	19
3.3	Moving cross-correlation plots with $W = 101$ of the DPCs of the simulated series with three variables in Example 3.1.1 after applying DPCA with $l = 1$	19
3.4	Eigenvalues' plots after applying MDPCA to a series of five variables with three static and two dynamic relations.	24
3.5	A time series plot of the simulated series with eight variables in Example 3.1.2.	31
3.6	Sample cross-correlation plots of the simulated series with eight variables in Example 3.1.2 - Part 1.	32
3.7	Sample cross-correlation plots of the simulated series with eight variables in Example 3.1.2 - Part 2.	32
3.8	Sample cross-correlation plots of the simulated series with eight variables in Example 3.1.2 - Part 3.	33
3.9	Sample cross-correlation plots of the simulated series with eight variables in Example 3.1.2 - Part 4.	33

3.10 Sample moving cross-correlation plots with a window size 101 of the simulated series with eight variables in Example 3.1.2 - part 1. 34

3.11 Sample moving cross-correlation plots with a window size 101 of the simulated series with eight variables in Example 3.1.2 - part 2. 34

3.12 Sample moving cross-correlation plots with a window size 101 of the simulated series with eight variables in Example 3.1.2 - part 3. 35

3.13 Sample moving cross-correlation plots with a window size 101 of the simulated series with eight variables in Example 3.1.2 - part 4. 35

3.14 Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.1 [Left: MDPCA ($W = 101$, $l = 1$), Right: MDPCA($W = 101$, $l = 5$)]. 42

3.15 Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.1 [Left: MDPCA ($W = 201$, $l = 1$), Right: MDPCA($W = 201$, $l = 5$)]. 43

3.16 Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.1 [Left: MDPCA ($W = 301$, $l = 1$), Right: MDPCA($W = 301$, $l = 5$)]. 43

3.17 Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.1 [Left: $T = 200$, Right: $T = 400$]. 46

3.18 Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.1 [Left: $T = 600$, Right: $T = 800$]. 47

3.19 Eigenvalues' plots after applying MDPCA with $W = 101$ and different sizes of l to the simulated series with three variables in Example 3.3.1. 49

3.20 Top: An eigenvalues' plot of the simulated series with three variables in Example 3.3.1 after applying MDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot. 50

3.21 Sample moving cross-correlation plots of the MDPCs of the simulated series with three variables in Example 3.3.1 after applying MDPCA with $W = 101$ and $l = 1$ 50

3.22	A time series plot of the simulated series with six variables in Example 3.3.2.	52
3.23	Plots of the mean of the absolute value of the moving cross-correlations between three MDPCs in Example 3.3.2 [Left: MDPCA ($W = 101$, $l = 1$), Right: MDPCA($W = 101$, $l = 5$)].	53
3.24	Plots of the mean of the absolute value of the moving cross-correlations between three MDPCs in Example 3.3.2 [Left: MDPCA ($W = 201$, $l = 1$), Right: MDPCA($W = 201$, $l = 5$)].	54
3.25	Plots of the mean of the absolute value of the moving cross-correlations between three MDPCs in Example 3.3.2 [Left: MDPCA ($W = 301$, $l = 1$), Right: MDPCA($W = 301$, $l = 5$)].	54
3.26	Plots of the mean of the absolute value of the moving cross-correlations between three MDPCs with different sizes of T in Example 3.3.2 [Left: $T = 600$, Right: $T = 800$].	58
3.27	Plots of the mean of the absolute value of the moving cross-correlations between three MDPCs with different sizes of T in Example 3.3.2 [Left: $T = 1000$, Right: $T = 1200$].	58
3.28	Eigenvalues' plots after applying MDPCA with $W = 101$ and different sizes of l to the simulated series with six variables in Example 3.3.2.	60
3.29	Top: An eigenvalues' plot of the simulated series with six variables in Example 3.3.2 after applying MDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot.	61
3.30	Sample moving cross-correlation plots of the MDPCs of the simulated series in Example 3.3.2 after MDPCA with $W = 101$ and $l = 1$ is applied.	61
3.31	A time series plot of the simulated series with ten variables in Example 3.3.3.	63
3.32	Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.3 [Left: MDPCA ($W = 101$, $l = 1$), Right: MDPCA($W = 101$, $l = 5$)].	64

3.33	Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.3 [Left: MDPCA ($W = 201$, $l = 1$), Right: MDPCA($W = 201$, $l = 5$)].	65
3.34	Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.3 [Left: MDPCA ($W = 301$, $l = 1$), Right: MDPCA($W = 301$, $l = 5$)].	65
3.35	Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.3 [Left: $T = 200$, Right: $T = 400$].	68
3.36	Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.3 [Left: $T = 600$, Right: $T = 800$].	69
3.37	Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.4 [Left: MDPCA ($W = 101$, $l = 1$), Right: MDPCA($W = 101$, $l = 5$)].	73
3.38	Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.4 [Left: MDPCA ($W = 201$, $l = 1$), Right: MDPCA($W = 201$, $l = 5$)].	73
3.39	Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.4 [Left: MDPCA ($W = 301$, $l = 1$), Right: MDPCA($W = 301$, $l = 5$)].	74
3.40	Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.4 [Left: $T = 200$, Right: $T = 400$].	77
3.41	Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.4 [Left: $T = 600$, Right: $T = 800$].	78
3.42	The daily stock prices of the ten US companies from November 2013 to December 2017 in Example 3.3.5.	81

3.43	Sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the ten US companies before MDPCA is applied in Example 3.3.5 - Part 1.	82
3.44	Sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the ten US companies before MDPCA is applied in Example 3.3.5 - Part 2.	82
3.45	Sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the ten US companies before MDPCA is applied in Example 3.3.5 - Part 3.	83
3.46	Sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the ten US companies before MDPCA is applied in Example 3.3.5 - Part 4.	83
3.47	Eigenvalues' plots after applying MDPCA with $W = 101$ to the ten US companies in Example 3.3.5.	84
3.48	Top: An eigenvalues' plot of the ten US companies in Example 3.3.5 after applying MDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot.	85
3.49	Sample moving cross-correlation plots of the three MDPCs of the ten US companies after MDPCA with $W=101$ and $l=1$ is applied in Example 3.3.5.	85
3.50	A time series plot of the daily stock prices of the 17 US companies in Example 3.3.6- Part 1	86
3.51	A time series plot of the daily stock prices of the 17 US companies in Example 3.3.6- Part 2	87
3.52	Last 36 sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the 17 US companies before MDPCA is applied in Example 3.3.6.	88
3.53	Eigenvalues' plots after applying MDPCA with $W = 101$ to the 17 US companies in Example 3.3.6.	88
3.54	Top: An eigenvalues' plot of the 17 US companies in Example 3.3.6 after applying MDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot.	89

3.55	Sample moving cross-correlation plots of the MDPCs of the 17 US companies in Example 3.3.6 after MDPCA with $W=101$ and $l=1$ is applied.	89
4.1	A time series plot of the randomly selected simulated series with three variables in Example 4.3.1.	103
4.2	Plots of the mean of the absolute value of the moving cross-correlations between two components in Example 4.3.1 [Left: QMDPCA ($l = 1$), Right: DPCA($l = 1$)].	105
4.3	Plots of the mean of the absolute value of the moving cross-correlations between two components in Example 4.3.1 [Left: QMDPCA ($l = 5$), Right: DPCA($l = 5$)].	105
4.4	Plots of the mean of the absolute value of the acf function between two QMDPCs with different sizes of T in Example 4.3.1 [Left: $T = 200$, Right: $T = 400$].	108
4.5	Plots of the mean of the absolute value of the acf function between two QMDPCs with different sizes of T in Example 4.3.1 [Left: $T = 600$, Right: $T = 800$].	108
4.6	Eigenvalues' plots after applying QMDPCA with $W = N$ and different sizes of l to the simulated series with three variables in Example 4.3.1.	111
4.7	Top: An eigenvalues' plot of the simulated series with three variables in Example 4.3.1 after applying QMDPCA with $W = N$ and $l = 1$. Bottom: RCC plot.	111
4.8	Sample moving cross-correlation plots of the retained two QMDPCs of the simulated series in Example 4.3.1.	112
4.9	A time series plot of the simulated non-stationary series with five variables in Example 4.3.2.	113
4.10	Plots of the mean of the absolute value of the moving cross-correlations between two components in Example 4.3.2 [Left: QMDPCA ($W = 101, l = 1$), Right: MDPCA($W = 101, l = 1$)].	115

4.11 Plots of the mean of the absolute value of the moving cross-correlations between two components in Example 4.3.2 [Left: QMDPCA ($W = 101, l = 5$), Right: MDPKA($W = 101, l = 5$)]. 115

4.12 Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.2 [Left: QMDPCA ($W = 101, l = 1$), Right: QMDPCA($W = 101, l = 5$)]. 119

4.13 Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.2 [Left: QMDPCA ($W = 201, l = 1$), Right: QMDPCA($W = 201, l = 5$)]. 119

4.14 Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.2 [Left: QMDPCA ($W = 301, l = 1$), Right: QMDPCA($W = 301, l = 5$)]. 120

4.15 Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.2 [Left: $T = 200$, Right: $T = 400$]. 123

4.16 Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.2 [Left: $T = 600$, Right: $T = 800$]. 124

4.17 Eigenvalues' plots after applying QMDPCA with $W = 101$ and different sizes of l to the simulated series with five variables in Example 4.3.2. 127

4.18 Top: An eigenvalues' plot of the simulated series with five variables in Example 4.3.2 after applying QMDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot. 127

4.19 Sample moving cross-correlation plots of the retained two QMDPCs of the simulated series in Example 4.3.2. 128

4.20 A time series plot of the simulated non-stationary series with ten variables in Example 4.3.3. 129

4.21 Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.3 [Left: QMDPCA ($W = 101, l = 1$), Right: QMDPCA($W = 101, l = 5$)]. 131

4.22 Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.3 [Left: QMDPCA ($W = 201$, $l = 1$), Right: QMDPCA($W = 201$, $l = 5$)]. 131

4.23 Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.3 [Left: QMDPCA ($W = 301$, $l = 1$), Right: QMDPCA($W = 301$, $l = 5$)]. 132

4.24 Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.3 [Left: $T = 200$, Right: $T = 400$]. 135

4.25 Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.3 [Left: $T = 600$, Right: $T = 800$]. 136

4.26 Eigenvalues' plots after applying QMDPCA with $W = 101$ and different sizes of l to the simulated series with ten variables in Example 4.3.3. 139

4.27 Top: An eigenvalues' plot of the simulated series with ten variables in Example 4.3.3 after applying QMDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot. 140

4.28 Sample moving cross-correlation plots of the retained two QMDPCs of the simulated series in Example 4.3.3. 140

4.29 A time series plot of the simulated non-stationary series with 15 variables in Example 4.3.4 - Part 1. 142

4.30 A time series plot of the simulated non-stationary series with 15 variables in Example 4.3.4 - Part 2. 142

4.31 Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.4 [Left: QMDPCA ($W = 101$, $l = 1$), Right: QMDPCA($W = 101$, $l = 5$)]. 144

4.32 Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.4 [Left: QMDPCA ($W = 201$, $l = 1$), Right: QMDPCA($W = 201$, $l = 5$)]. 144

4.33	Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.4 [Left: QMDPCA ($W = 301$, $l = 1$), Right: QMDPCA($W = 301$, $l = 5$)].	145
4.34	Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.4 [Left: $T = 200$, Right: $T = 400$].	148
4.35	Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.4 [Left: $T = 600$, Right: $T = 800$].	149
4.36	A time series plot of the daily stock prices of the ten US companies in Example 4.3.5	153
4.37	First 36 sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the ten US companies in Example 4.3.5 before QMDPCA is applied.	153
4.38	Eigenvalues' plots after applying QMDPCA with $W = 101$ to the ten US companies in Example 4.3.5.	154
4.39	Top: An eigenvalues' plot of the ten US companies in Example 4.3.5 after applying QMDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot.	155
4.40	Sample moving cross-correlation plots of the three QMDPCs of the ten US companies in Example 4.3.5 after QMDPCA with $W=101$ and $l=1$ is applied.	155
4.41	A time series plot of the daily stock prices of the 18 US companies in Example 4.3.6 - Part 1.	157
4.42	A time series plot of the daily stock prices of the 18 US companies in Example 4.3.6 - Part 2.	157
4.43	Last 36 sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the 18 US companies in Example 4.3.6 before QMDPCA is applied.	158
4.44	Eigenvalues' plots after applying QMDPCA with $W = 101$ to the 18 US companies in Example 4.3.6.	159

4.45	Top: An eigenvalues' plot of the 18 US companies in Example 4.3.6 after applying QMDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot.	159
4.46	Sample moving cross-correlation plots of the QMDPCs of the 18 US companies in Example 4.3.6 after QMDPCA with $W=101$ and $l=1$ is applied.	160
5.1	A time series plot of the simulated series of eight variables in Example 5.2.1.	172
5.2	A time series plot of the randomly selected replica in Example 5.4.1.	186
5.3	Sample moving cross-correlation plots with $W = 101$ between the components of the randomly selected replica in Example 5.4.1 before transformation.	186
5.4	Sample moving cross-correlation plots with $W = 101$ between the components of the randomly selected replica in Example 5.4.1 after GTS-PCA with $W = 101$ is applied.	187
5.5	A time series plot of the randomly selected replica in Example 5.4.2.	189
5.6	Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 before transformation - part 1.	189
5.7	Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 before transformation - part 2.	190
5.8	Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 before transformation - part 3.	190
5.9	Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 before transformation - part 4.	191
5.10	Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 after GTS-PCA with $W = 101$ is applied - part 1.	191
5.11	Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 after GTS-PCA with $W = 101$ is applied - part 2.	192

5.12	Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 after GTS-PCA with $W = 101$ is applied - part 3.	192
5.13	Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 after GTS-PCA with $W = 101$ is applied - part 4.	193
5.14	A time series plot of the variables 1 - 10 of the randomly selected replica in Example 5.4.3.	195
5.15	A time series plot of the variables 11 - 20 of the randomly selected replica in Example 5.4.3.	196
5.16	First 50 sample moving cross-correlation plots $W = 101$ of the randomly selected replica in Example 5.4.3 before transformation - part 1.	196
5.17	First 50 sample moving cross-correlation plots $W = 101$ of the randomly selected replica in Example 5.4.3 before transformation - part 2.	197
5.18	Last 50 sample moving cross-correlation plots $W = 101$ of the randomly selected replica in Example 5.4.3 before transformation - part 1.	197
5.19	Last 50 sample moving cross-correlation plots $W = 101$ of the randomly selected replica in Example 5.4.3 before transformation - part 2.	198
5.20	A time series plot of the randomly selected replica in Example 5.4.4.	199
5.21	Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.4 before transformation.	200
5.22	Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.4 after GTS-PCA is applied.	201
5.23	A time series plot of the daily stock prices of the six US companies in Example 5.4.5.	202
5.24	Sample moving cross-correlation plots of the daily stock prices of the six US companies in Example 5.4.5 before GTS-PCA is applied.	203

5.25	Sample moving cross-correlation plots of the daily stock prices of the six US companies in Example 5.4.5 after the GTS-PCA is applied. . .	203
5.26	A time series plot of the daily stock prices of the 22 US companies in Example 5.4.6 - Part 1.	204
5.27	Time series plot of the daily stock prices of the 22 US companies in Example 5.4.6 - Part 2.	205
5.28	Time series plot of the daily stock prices of the 22 US companies in Example 5.4.6 - Part 3.	205
5.29	First 50 sample moving cross-correlation plots of the daily stock prices of the 22 US companies in Example 5.4.6 before GTS-PCA is applied - Part 1.	206
5.30	First 50 sample moving cross-correlation plots of the daily stock prices of the 22 US companies in Example 5.4.6 before GTS-PCA is applied - Part 2.	206
5.31	Last 50 sample moving cross-correlation plots of the daily stock prices of the 22 US companies in Example 5.4.6 before GTS-PCA is applied - Part 1.	207
5.32	Last 50 sample moving cross-correlation plots of the daily stock prices of the 22 US companies in Example 5.4.6 before GTS-PCA is applied - Part 2.	207

Chapter 1

Introduction

1.1 Background

Multivariate time series analysis has many applications as it can account for interrelations between variables. Advanced technology nowadays allows for the collection of multivariate natured data in a wide range of fields, such as economics, industry, healthcare and social networks. The majority of these data has a large number of variables. Many of the existing models, such as vector autoregressive integrated moving average (VARIMA) models, face the challenge of complexity in their structures when applied to series with large dimensions. This complexity occurs because the number of parameters expands enormously fast as the dimension increases, which leads to complex and over-parametrized models. Therefore, reducing the dimension of time series becomes critical to manage such data. The idea behind dimension reduction is to represent original data with a less-dimensional version that has almost all information as the original one. Once the reduced dimensional data is produced, further analyses can be applied to it instead of the original series.

Many approaches have been proposed in literature for dimension reduction. One popular approach to achieving this is the use of principal component analysis (PCA) by projecting the original data into a space with fewer dimensions. PCA is a commonly used technique to perform dimension reduction for static and independent multivariate data. However, because of the dynamic nature of multivariate time

series data, the classical PCA technique will not be applicable. The reason is that classically PCA is static and therefore will not be able to capture the dynamic dependence between the components (i.e. variables) of a multivariate time series.

Ku et al. (1995) introduced dynamic principal component analysis (DPCA) by including lagged series into the analysis. As its name suggests, DPCA is an extension of the classical PCA to dynamic data (e.g., time series data). Without losing a valuable amount of information, the results of projected components are linear combinations of both current and lagged values of the data.

The principal component analysis for time series (TS-PCA) of Chang et al. (2018) extended PCA by transforming the original series into uncorrelated subseries with lower dimensions. The resulted subseries can be separately analysed as they are uncorrelated.

Both DPCA and TS-PCA assume stationary series, and therefore, they are not suitable to reduce the dimensions of non-stationary series such as stock prices. Thus, the need for a dimension reduction method that can deal directly with non-stationary multivariate time series increases.

Many PCA-based methods were proposed in order to account for non-stationarity such as moving window principal component analysis (MWPCA) by Lennox et al. (2001) and Variable MWPCA by He and Yang (2008) and many others. These methods were mostly developed for process monitoring where PCA is performed separately on each window. By including the next time point and excluding the oldest time point, new results are obtained based on the new window and so on. However, valuable information would be lost by excluding a large amount of observation as the results are based on one window at a time.

Brillinger (1981) proposed another related approach where the reduction is produced based on a reconstruction criterion. The resulted dynamic components are

linear combinations of the original series. Peña and Yohai (2016) proposed the generalized dynamic principal component analysis (GDPCA) where the original data is reconstructed based on a loss function. This method accounts for both stationary and non-stationary series and produces dynamic principal components that could be non-linear combinations of the original data with nearly zero reconstruction error. This precision is a result of using an iteration method that minimizes the reconstruction error. However, using this iteration method reduces the accuracy of performing forecasting using GDPCA's results.

Factor models are widely used tools to reduce the dimension of multivariate time series where the variables of the observed series are considered as linear combinations of some hidden factors that could be interpreted subjectively. See, for example, Peña and Box (1987), Stock and Watson (1988, 2002), Bai and Ng (2002), Forni et al. (2005), Peña and Poncela (2006), Pan and Yao (2008), Lam and Yao (2012) and many others. These models are related to PCA as they are based on eigenanalysis of the covariance matrix of the data.

Dimension reduction for time series data can also be achieved using canonical correlation analysis proposed by Box and Tiao (1977) and scalar component analysis by Tiao and Tsay (1989).

In this research, new dimension reduction methods will be proposed. The new methods are general in the sense that they can be applied to a wide range of multivariate stationary and non-stationary time series data. Under the PCA framework, it will be shown that using an advance matrix that can be updated as we move in time will enable PCA to deal with non-stationary time series more effectively. The assumptions of the introduced methods will be discussed, along with their theoretical properties. Additionally, the numerical properties of the introduced methods will be investigated based on simulated and real data sets.

1.2 Overview of the Study

In this thesis, dimension reduction methods will be developed in order to account for both stationary and non-stationary multivariate time series data. PCA-based methods offer flexible frameworks for reducing the dimension of a wide range of data. We are going to show that by implementing an advance cross-covariance matrix named moving cross-covariance matrix instead of the classical cross-covariance matrix, PCA can be extended to account for static, dynamic, stationary and non-stationary behaviours in a time series data.

Chapter 1 includes a background and a historical review of dimension reduction methods. In Chapter 2, a brief review of preliminary topics related to time series data and dimension reduction will be provided.

In Chapter 3, moving dynamic principal component analysis (MDPCA) will be proposed. This method will generalize DPCA in order to be applicable on non-stationary time series. The main difference between DPCA and the new method is that the former uses the classical covariance matrix and the latter uses a moving cross-covariance matrix. The method we propose is different from MWPCA methods mentioned earlier, where our method will use all observations to calculate one moving cross-covariance matrix that consists of static and dynamic information of the whole series. New proposed diagnostic and optimization tools for MDPCA' results will be provided. The theoretical properties of our estimators will be shown. The ability of MDPCA to dimension reduction will be examined on both simulated and real data.

Chapter 4 introduces a new method named the quadratic moving dynamic principal component analysis (QMDPCA). This dimension reduction method also extends DPCA to reduce the dimension of non-stationary series. This method will be different from DPCA and MDPCA, where QMDPCA uses a quadratic form of the moving cross-covariance matrix in its calculations. Theoretical properties of the used estimators will be studied. By the end of this chapter, numerical evidence will

be provided to support our method where both simulated and real data are considered.

In Chapter 5, another dimension reduction method named the generalized principal component analysis for time series (GTS-PCA) will be revealed. This method will extend the stationary TS-PCA of Chang et al. (2018) to non-stationary time series. It will be shown that the results of GTS-PCA are uncorrelated subseries with smaller dimensions that can be analysed individually. In this chapter, a full description of the GTS-PCA method will be given. Also, a new columns' rearrangement tool called the maximum moving cross-correlation method will be proposed in order to determine the resulted uncorrelated subseries. Theoretical asymptotic properties of the used estimators will also be shown. This chapter will include simulation studies and real data examples in order to justify the generality of the GTS-PCA method to reduce the dimension of both stationary and non-stationary multivariate time series.

The last chapter of this thesis is Chapter 6, where we conclude our work by providing essential remarks, and stating possible ideas for further research.

We developed eight packages (i.e. libraries) using the statistical software R. These packages consist of the necessary functions used to carry out the analyses in Chapter 3, 4 and 5. The functions are publicly available under the following R packages: `GTSPCA` of Alshammri (2020a), `MACF` of Alshammri (2020b), `MCOV` of Alshammri (2020c), `MDPCA` of Alshammri (2020d), `MpermutMax` of Alshammri (2020e), `QMDPCA` of Alshammri (2020f), `RCC_MDPCA` of Alshammri (2020g) and `RCC_QMDPCA` of Alshammri (2020h).

Chapter 2

Preliminaries

2.1 Multivariate Time Series

Multivariate time series analysis can be considered as a vital branch of multivariate statistical analysis. A multivariate time series can be obtained by observing a sequence of measurements of two or more variables of interest over equally spaced time points (e.g. hourly, daily or monthly). Multivariate time series analysis studies the overall and the dynamic relations among variables under study.

Let \mathbf{Z} be a multivariate time series with m variables that are observed at T equally spaced time points (indexed by t where $1 \leq t \leq T$), which can be represented as an $m \times T$ matrix or as an m -dimensional column vector $\mathbf{z}_t = (z_{1,t}, z_{2,t}, \dots, z_{m,t})'$. For example, consider the daily stock prices of the companies Apple and Microsoft in US Dollar from Nov 07, 2013 to Dec 18, 2017. This data is available on *Yahoo!* Finance. Let $z_{1,t}$ represents the daily stock price of Apple and $z_{2,t}$ represents daily stock price of Microsoft. By combining these two series into a multivariate time series, one can gain more knowledge about the behaviours of $z_{1,t}$ taking $z_{2,t}$ into consideration, instead of studying each series separately. In other words, by using a multivariate time series, we are able to explore the overall and the within-relation between daily stock prices of Apple and Microsoft, where the value of one of them could be correlated with the other. A time series plot of the multivariate time series of the daily stock price of the two companies is shown in Figure 2.1.

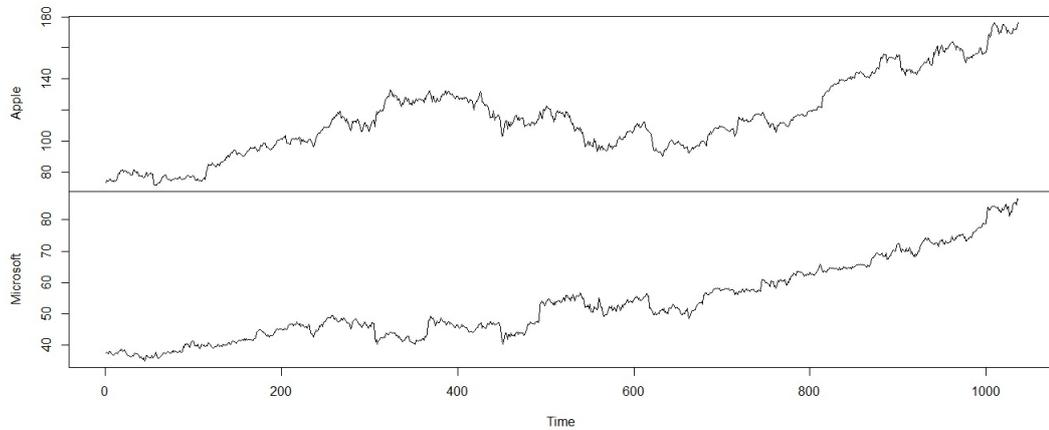


Figure 2.1: The daily stock prices of the companies Apple and Microsoft from November 2013 to December 2017.

There are many ways to explore a multivariate time series data such as the daily stock prices data above. For instance, one could describe the overall pattern of each series. Also, it is possible to study the interaction between the variables and investigate how the past values of one variable affect the current and future values of itself and other variables. Furthermore, a prediction of future values of the series can be carried out.

There are basic characters that enable us to describe a time series such as trends, seasonality, cycles and random variations. Trends can be found when the values of a time series increase or decrease gradually over time. This can cause the mean of the series to be non-constant over time. Seasonality exists where a pattern is repeated at regular intervals called seasons (e.g. weeks, quarters or years). Cycles are patterns that are repeated at long-term periods that are not related to seasons. Random variations are sudden changes that might not be explained in the series, which occur randomly and do not follow an apparent pattern. By looking back at the daily stock prices example to describe the time series of the two variables according to their plots in Figure 2.1, we can notice that the daily stock prices of both Apple and Microsoft increased gradually over time, which is sign for existing trends

for both variables. Also, we do not notice any repeated patterns (i.e. seasonality or long-term cycles) in both variables.

Furthermore, linear models can be used to predict the current values of a multivariate series based on past values (also called lagged values) of the series. The prediction can be made using models such as vector autoregressive integrated moving average (VARIMA) family models. These models have some regularity assumptions. One of the main assumptions is related to the stationarity of the multivariate series under study. The concept of stationarity will be discussed in the next topic.

2.2 Stationarity of a Multivariate Time Series

The following is an overview of the stationarity of a multivariate time series. Generally, when studying time series analysis, stationarity is a critical concept as statistical tools used to analyse stationary time series might be not suitable for a non-stationary time series.

A multivariate time series \mathbf{z}_t is said to be *strictly stationary* if the joint distribution of $(\mathbf{z}_{t_1}, \mathbf{z}_{t_2}, \dots, \mathbf{z}_{t_k})$ is the same as that of $(\mathbf{z}_{t_1+l}, \mathbf{z}_{t_2+l}, \dots, \mathbf{z}_{t_k+l})$ for arbitrary positive integers t_1, \dots, t_k , k and l . This is, shifting the time origin of a strictly stationary data will not affect the joint distribution of the data; See Reinsel (2003). An example of a strictly stationary time series is the independent and identically distributed Gaussian white noise process. Other types of stationarity with fewer restrictions on the data are available and called *weakly stationary* time series.

A time series \mathbf{z}_t is said to be weakly stationary if its first two moments exist and are independent of time. That is, in order for the series \mathbf{z}_t to be weakly stationary, we need to verify the following two conditions:

1. $E(\mathbf{z}_t) = \boldsymbol{\mu}$ is constant and independent of t .
2. $\text{Cov}(\mathbf{z}_t, \mathbf{z}_{t+l}) = E[(\mathbf{z}_t - \boldsymbol{\mu})(\mathbf{z}_{t+l} - \boldsymbol{\mu})']$ is constant and independent of t for each $l \geq 0$.

Here $E(\mathbf{z}_t)$ is the expectation of \mathbf{z}_t , $\boldsymbol{\mu}$ is the m -dimensional mean vector of \mathbf{z}_t and $\text{Cov}(\mathbf{z}_t, \mathbf{z}_{t+l})$ is the $m \times m$ covariance matrix between \mathbf{z}_t and its self l time-points apart; See Reinsel (2003). An example of a weak stationarity is the white noise process. Furthermore, it can be seen that the conditions of weakly stationarity are implied by strict stationarity. Hence a strictly stationary series is also weakly stationary, but the opposite does not have to be true.

As described above that a weakly stationary series has appealing properties that make the handling of this type of data easier such as constant statistical properties (i.e. mean, covariance and correlation) over time. This is one main reason that many approaches are developed for stationary series such as the early mentioned VARIMA models. In the following, we are going to use the term *stationary* to refer to a weakly stationary series.

Local stationarity is an important type of non-stationarity. Informally speaking, a time series is locally stationary if its statistical properties change slowly as we move in time; See Dahlhaus and Rao (2006). A formal definition of a locally stationary process can be found in Vogt (2012). An interesting feature of local stationary series is that its statistical properties can be calculated within certain width (i.e. window) around each time point similar to stationary series.

Checking the stationarity of a multivariate time series can be done in different ways such as investigating time series plots or using statistical tests. Plotting time series data is a subjective way to assess the stationarity of each variable in the series, where one can spot an apparent trend or seasonality pattern that indicate a significant change in the mean of the series or notice a change in data variation over time which can be a sign of a non-constant covariance. If a multivariate time series is stationary, then all its variables are stationary. If one variable of a multivariate time series is non-stationary, then the multivariate time series is non-stationary. If we look back again at the time series plots of variables from the daily stock prices example in Figure 2.1, we find this series to be an example of a non-stationary mul-

tivariate time series as each variable exhibits an apparent trend, which violates the constant mean assumption of a stationary series.

Statistical tests can also be used to assess the stationarity of a multivariate time series, which are more objective than using time series plots. Jentsch and Subba Rao (2015) developed a test for checking the stationarity of a multivariate time series based on the discrete Fourier transforms (DFT) of a multivariate time series. This test investigates the stationarity of a multivariate time series at once instead of checking each variable individually. It is also possible to use univariate tests such as Dickey–Fuller test to spot non-stationary variables in a multivariate time series; See Dickey and Fuller (1979).

It is possible to obtain a stationary series that is a linear combination of non-stationary variables. The non-stationary variables in this case are called *co-integrated* variables, and this procedure is called *co-integration*. In other words, co-integration could produce stationary linear combinations of variables in the series, where these variables are non-stationary. A popular co-integration test is Johansen’s test; See Johansen (1995) for more details about co-integration tests. Zhang et al. (2019) proposed to do co-integration analysis for dimensional reduction of a high dimensional time series based a nonnegative definite matrix when there is co-integration relation among components. The co-integration approach is entirely different from PCA. However, one wonder if these two approaches can be combined in a new approach, which can be a subject of a new research.

It is a common practice to transform a non-stationary series into a stationary series so that the data can be analyzed using stationary models and tools. One can take the logarithm of the data so that the transformed data (i.e. data obtained after applying the logarithm) has a reduced variance. Moreover, one can use methods such as differencing to remove a deterministic trend or a seasonality. Differencing can be done by taking the difference between consecutive observations in the series. For example, for a variable z_t , which is non-stationary series due to a trend, then

$\nabla z_t = z_t - z_{t-1}$ can be stationary, where this is called first-order differencing. There are cases where first differencing will not stabilize the mean of the series so that second differencing is needed (i.e. first differencing is performed again on a first differenced series) and so on. Differencing for seasonality can be done by differencing between observations that are one season apart. See Tsay (2013) for further reading.

2.3 Principal Component Analysis

Principal component analysis is a widely used tool in multivariate analysis in general. PCA can reach dimensionality reduction of the data by searching for linear combinations of the original variables. These combinations, in turn, are supposed to represent the original data with a smaller number of components. The idea behind this is to return important combinations that explain most variations of the original data and exclude those with non-important contributions.

In specific, let \mathbf{z} be an m -dimensional random vector $\mathbf{z} = (z_1, z_2, \dots, z_m)'$. For simplicity, assume \mathbf{z} has a zero $m \times 1$ mean vector $\bar{\mathbf{z}}$. This is

$$\bar{\mathbf{z}} = \frac{1}{n} \sum_{i=1}^n \mathbf{z}_i = \mathbf{0}.$$

Then the $m \times m$ cross-covariance matrix of \mathbf{z} will be

$$\text{Cov}(\mathbf{z}) = \Sigma = \frac{1}{n} \mathbf{z}\mathbf{z}'.$$

Now, suppose $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$ are the m eigenvalues, and u_1, u_2, \dots, u_m are the corresponding eigenvectors of Σ . Methods such as spectral decomposition can be used to obtain the eigenvalues and eigenvectors of Σ . In particular, spectral decomposition of Σ gives

$$\Sigma = \mathbf{U}\Lambda\mathbf{U}'$$

where \mathbf{U} is an $m \times m$ matrix whose columns are the eigenvectors u_1, u_2, \dots, u_m of Σ and Λ is an $m \times m$ diagonal matrix whose diagonal entries are the corresponding eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_m$.

PCA transforms \mathbf{z} into a new data vector \mathbf{x} that consists of m components called principal components (PC), which are linear combinations of the m variables of \mathbf{z} . The new vector \mathbf{x} is obtained by left multiplying \mathbf{z} by the transpose of the eigenvectors' matrix \mathbf{U} as follows

$$\mathbf{x} = \mathbf{U}'\mathbf{z}.$$

Notice that x_i (i.e. the i^{th} component of \mathbf{x}) where $1 \leq i \leq m$, can be obtained as follows

$$x_i = u_i'\mathbf{z}.$$

Furthermore, the variance of x_i is i^{th} eigenvalue of Σ . This is

$$\text{Var}(x_i) = \lambda_i.$$

Thus, the amount of variation in original data that is explained by x_i is

$$\frac{\lambda_i}{\lambda_1 + \lambda_2 + \cdots + \lambda_m}.$$

Up to this point, \mathbf{x} still has a dimension of m which is equivalent to the dimension of \mathbf{z} . Dimension reduction in PCA can be done by retaining only the first k PCs, where $k \leq m$, such that

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_m} \simeq 1.$$

In other words, PCA is aiming to retain only the first k components of \mathbf{x} (i.e. $\mathbf{x}^* = (x_1, x_2, \cdots, x_k)'$) that explain almost all variation of the original data \mathbf{z} . The retained k PCs will represent \mathbf{z} where further analyses can be carried out on them instead of \mathbf{z} . See Jolliffe (2002) and Tsay (2013) for further reading about PCA.

Example 2.3.1 In this example, PCA will be applied on a multivariate data set *mtcars* that is available in R database. This data set compares 32 different cars in terms of 11 features taken from a car magazine; See Henderson and Velleman (1981). In this example, nine variables (i.e. features) from the *mtcars* data set will be used with all 32 samples. Notice that the number of samples used in this example is relatively small for data set with nine variables. In order to gain better

results, a larger sample could be used. However, the purpose of this example is just to show how PCA can be used to reduce the dimension of multivariate data. The used variables are listed below along with their labels:

- 1- mpg: Miles per gallon (US).
- 2- cyl: Number of cylinders.
- 3- disp: Displacement (cu.in.).
- 4- hp: Gross horsepower.
- 5- drat: Rear axle ratio.
- 6- wt: Weight (1000 lbs).
- 7- qsec: 1/4 mile time.
- 8- gear: Number of forward gears.
- 9- carb: Number of carburetors.

PCA is a scale-sensitive method, where it assumes all variables to be measured using the same scale, where important variables are those with larger variations. When variables are measured on different scales, this can enable less-important variables to have larger variation, which can lead to non-valid results. In order to avoid this, one can scale (i.e. standardize) the data before applying PCA, which is the case for *mtcars* data set, where each variable is measured using a different scale. The cross-covariance matrix of the nine variables after scaling is provided in Table 2.1, where generally moderate to strong relationships exist among all the variables. For example, the variables fuel consumption (mpg) and the number of cylinders (cyl) have a strong negative relationship about -0.85 .

	mpg	cyl	disp	hp	drat	wt	qsec	gear	carb
mpg	1.0000000	-0.8521620	-0.8475514	-0.7761684	0.68117191	-0.8676594	0.41868403	0.4802848	-0.5509251
cyl	-0.8521620	1.0000000	0.9020329	0.8324475	-0.69993811	0.7824958	-0.59124207	-0.4926866	0.5269883
disp	-0.8475514	0.9020329	1.0000000	0.7909486	-0.71021393	0.8879799	-0.43369788	-0.5555692	0.3949769
hp	-0.7761684	0.8324475	0.7909486	1.0000000	-0.44875912	0.6587479	-0.70822339	-0.1257043	0.7498125
drat	0.68117191	-0.6999381	-0.7102139	-0.4487591	1.00000000	-0.7124406	0.09120476	0.6996101	-0.0907898
wt	-0.8676594	0.7824958	0.8879799	0.6587479	-0.71244065	1.0000000	-0.17471588	-0.5832870	0.4276059
qsec	0.4186840	-0.5912421	-0.4336979	-0.7082234	0.09120476	-0.1747159	1.00000000	-0.2126822	-0.6562492
gear	0.4802848	-0.4926866	-0.5555692	-0.1257043	0.69961013	-0.5832870	-0.21268223	1.0000000	0.2740728
carb	-0.5509251	0.5269883	0.3949769	0.7498125	-0.09078980	0.4276059	-0.65624923	0.2740728	1.0000000

Table 2.1: The cross-correlation matrix of the nine variables in Example 2.3.1.

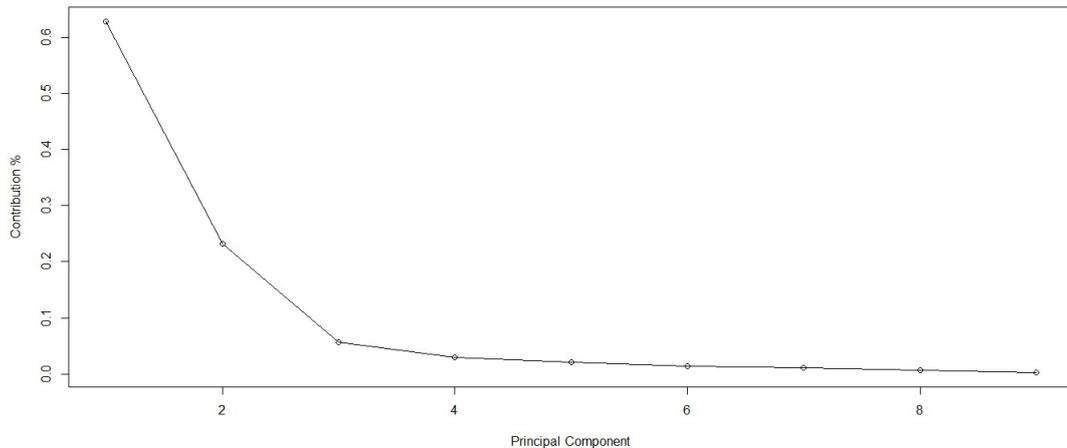


Figure 2.2: Eigenvalues' plot after applying PCA on the nine variables in Example 2.3.1.

After PCA is applied to the scaled data, the nine eigenvalues of the cross-covariance matrix of the *mtcars* data (rounded to two decimals) are 5.66, 2.08, 0.50, 0.27, 0.18, 0.12, 0.11, 0.06 and 0.02. Also, a plot of the eigenvalues is provided in Figure 2.2. It can be seen that the first PC explains 62.84% of the total variation in the original data, where the first two PCs explain 85.97% and the first three PCs explain 91.58% of the total variation. Thus, two or three PCs can be used to represent the original data set. If three PCs are retained, then we conclude that PCA reduces the dimension of the data from nine to three.

Recall that the transformed data set with reduced dimension can be obtained by left multiplying the original data matrix by the transpose of the first three eigenvectors that correspond to the first three eigenvalues. Further analyses can be performed on the transformed data set with three dimensions.

It is important to mention that PCA is static and does not account for the dynamic dependence in a multivariate time series data, where the values observed at time t depend on the values from another time (e.g. $t - 1$). Furthermore, PCA has assumptions such as a constant mean vector, which is not the case for non-stationary data. In the following chapters, we will attempt to extend PCA to multivariate time

series data that can be non-stationary.

Chapter 3

Moving Dynamic Principal Component Analysis

In this chapter, a new dimension reduction method will be proposed. This method will generalize DPCA of Ku et al. (1995) in order to be applicable to a wide range of data as it accounts for the dynamic and non-stationarity of time series data. The calculations of our new method will be based on a moving cross-covariance function that is updated at each time point to extract accurate information from different types of multivariate time series data. Therefore, the proposed method will be called the moving dynamic principal component analysis (MDPCA). In the following sections, we are going to reveal the building-structure of MDPCA along with new proposed diagnostic and optimization tools for the results of MDPCA. Also, the theoretical properties of the used estimators will be shown. The ability of MDPCA to achieve dimension reduction will be examined on both simulated and real data examples.

3.1 Methodology

Consider an m -dimensional time series $\mathbf{z}_t = (z_{1,t}, z_{2,t}, \dots, z_{m,t})'$, which is allowed to be non-stationary. The initial step in the MDPCA method is to build an $m(l+1)$ -dimensional extended data vector, denoted by \mathbf{y}_t , which consists of the series \mathbf{z}_t and its lagged series up to a pre-specified lag l . Then the extended data vector \mathbf{y}_t is

going to have the following structure

$$\mathbf{y}_t = (\mathbf{z}'_t, \mathbf{z}'_{t-1}, \dots, \mathbf{z}'_{t-l})' = (z_{1,t}, \dots, z_{m,t}, z_{1,t-1}, \dots, z_{m,t-1}, \dots, z_{1,t-l}, \dots, z_{m,t-l})'.$$

The rest of the analyses will be performed on \mathbf{y}_t instead of \mathbf{z}_t . Assume the series \mathbf{z}_t is observed at T time points. Let $m(l+1) = M$ and $T-l = N$. Let \mathbf{Y} be an $M \times N$ extended data matrix whose columns are $\mathbf{y}_1, \dots, \mathbf{y}_N$. A critical feature of the extended data vector \mathbf{y}_t is that its cross-covariance matrix will account for the dynamic relations that exist among the components (i.e., variables) of \mathbf{z}_t . This idea was first introduced to PCA by Ku et al. (1995) to reduce the dimension of dynamic data, while PCA is limited to static data.

For a stationary series, the DPCA of Ku et al. (1995) applies its analysis to the cross-covariance matrix of \mathbf{y}_t to reduce the dimension of \mathbf{z}_t . However, for a non-stationary time series, the results of the DPCA would not be valid as it assumes the first two moments to be constant for all time points. Furthermore, if DPCA is applied to a non-stationary series, it could produce correlated DPCs. This is mainly because the cross-covariance matrix will not be able to measure the dynamic dependence between the variables of non-stationary series. To address this issue, consider the following example.

Example 3.1.1 In this example, we show the results of applying DPCA of Ku et al. (1995) on a non-stationary series. A simulated non-stationary time series with three variables and a sample size of 1000 is generated. Three different non-stationary univariate models are used to generate the components of \mathbf{z}_t using the `arma.sim` command in R software. Let ∇ be the differencing notation, where $\nabla \mathbf{x}_t = \mathbf{x}_t - \mathbf{x}_{t-1}$. Let a_t , b_t , and c_t be three independent standard normal white noises, which are the innovation terms of the following three models, respectively, then:

$$\begin{cases} z_{1,t} = u_t \\ z_{2,t} = v_t \\ z_{3,t} = w_t \end{cases} \quad (3.1)$$

where $(u_t, v_t, w_t)'$ satisfies

$$\begin{cases} \nabla u_t = 0.25 \nabla u_{t-1} + a_t \\ \nabla v_t = -0.5 \nabla v_{t-1} + b_t \\ \nabla w_t = 0.7 \nabla w_{t-1} + c_t \end{cases} \quad (3.2)$$

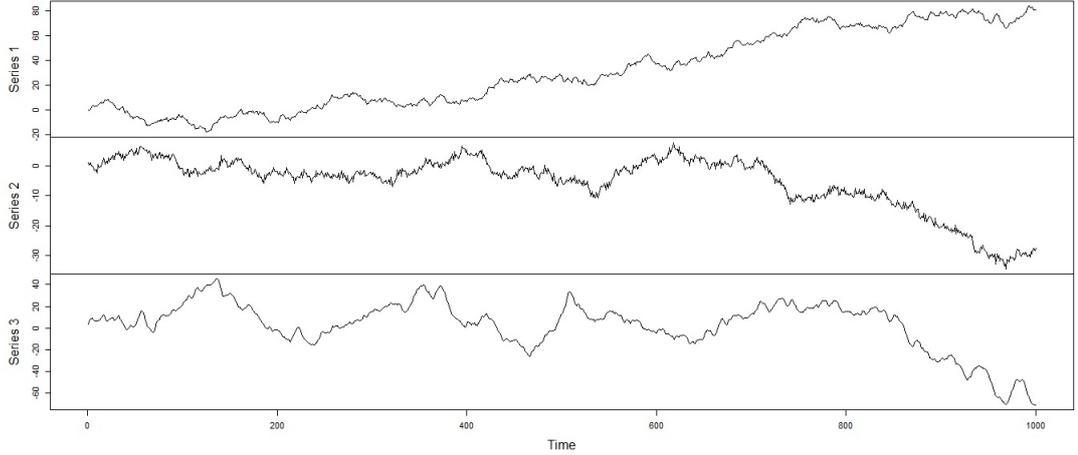


Figure 3.1: A time series plot of the simulated series with three variables in Example 3.1.1.

Figure 3.1 is a time series plot of the simulated non-stationary time series. If DPCA with one lag is applied to \mathbf{z}_t , then the extended data vector is going to be

$$\mathbf{y}_t = (\mathbf{z}'_t, \mathbf{z}'_{t-1})' = (z_{1,t}, z_{2,t}, z_{3,t}, z_{1,t-1}, z_{2,t-1}, z_{3,t-1})'$$

A plot of the six eigenvalues is provided in Figure 3.2. Two DPCs (i.e. dynamic principal components) explained 98.10 % of the total variation of the original series. This might look appealing. However, by using the moving cross-correlation plot with a window size of size 101 (will be introduced latter) to investigate the dynamic relationship between the two DPCs further, then we can see that they are strongly correlated, as shown in Figure 3.3. This is mainly because the DPCA assumes the data to be stationary, so it should not be applied to non-stationary series as its results might be misleading.

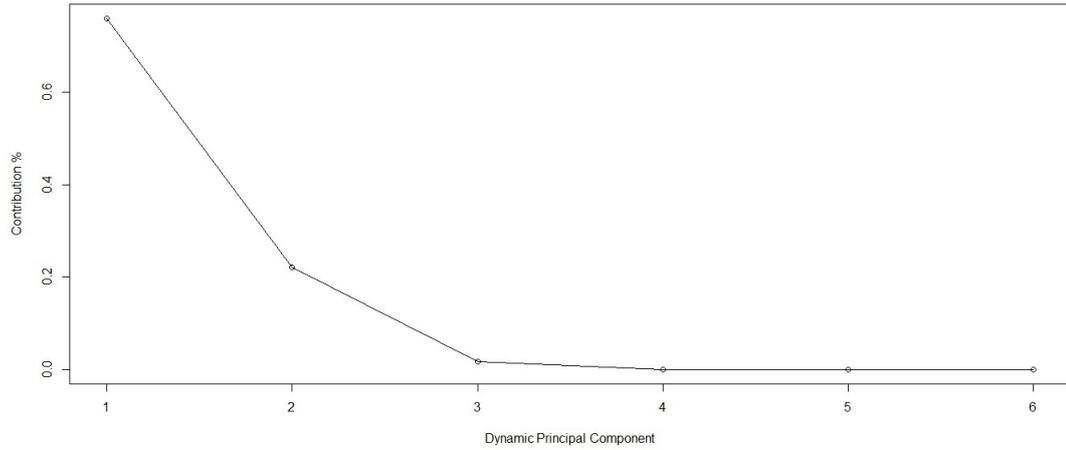


Figure 3.2: An eigenvalues' plot of the simulated series with three variables in Example 3.1.1 after applying DPCA $l = 1$.

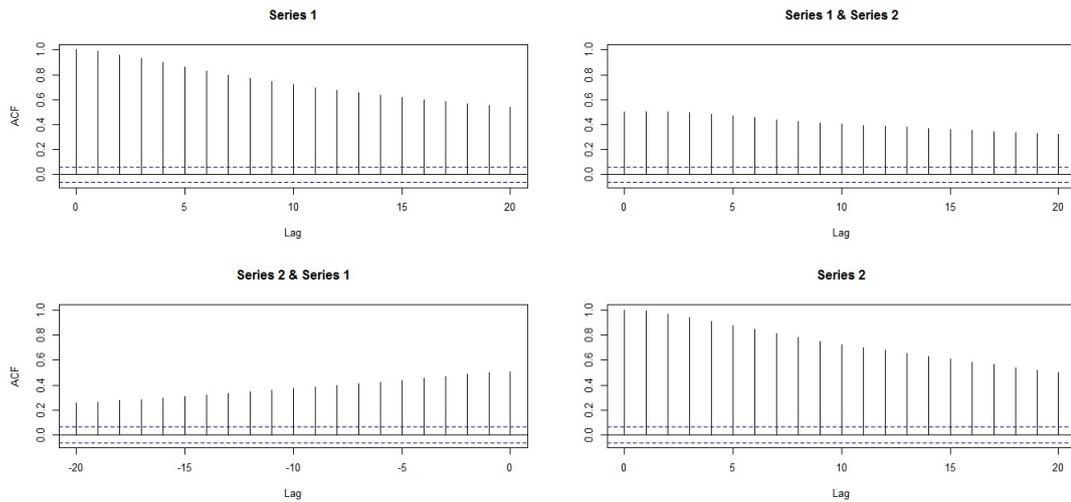


Figure 3.3: Moving cross-correlation plots with $W = 101$ of the DPCs of the simulated series with three variables in Example 3.1.1 after applying DPCA with $l = 1$.

One solution we propose is to use moving cross-covariance matrices in the calculations of DPCA. We show that these matrices will allow to capture the dynamic relations among the components of non-stationary time series, as they are able to be updated as we move in time. Before defining the moving cross-covariance matrix, the following definitions are needed. We assume that \mathbf{y}_i to be locally stationary.

Define the cross-covariance matrix of \mathbf{y}_i to be

$$\Gamma_i = \text{Var}(\mathbf{y}_i). \quad (3.3)$$

Once \mathbf{z}_t is observed, the sample cross-covariance matrix defined over window \mathbf{y}_i with a pre-specified size $W = 2w + 1$ can be calculated as follows

$$\hat{\Gamma}_i = \frac{1}{2w + 1} \sum_{t=i-w}^{i+w} (\mathbf{y}_t - \bar{\mathbf{y}}_i)(\mathbf{y}_t - \bar{\mathbf{y}}_i)' \quad (3.4)$$

where

$$\bar{\mathbf{y}}_i = \frac{1}{2w + 1} \sum_{t=i-w}^{i+w} \mathbf{y}_t$$

where w is a positive integer. The window size W is determined by the stationarity of the data. Determining the window size will be discussed further in section 3.1.1 of this chapter. Then, the moving cross-covariance matrix of \mathbf{y}_t can be defined as

$$\mathbf{M}\Gamma = \frac{1}{N - 2w} \sum_{i=w+1}^{N-w} \Gamma_i. \quad (3.5)$$

Notice that the moving cross-covariance matrix $\mathbf{M}\Gamma$ is an $M \times M$ symmetric matrix. Based on sample data, $\mathbf{M}\Gamma$ can be estimated by substituting $\hat{\Gamma}_i$ into equation (3.5) as

$$\hat{\mathbf{M}}\Gamma = \frac{1}{N - 2w} \sum_{i=w+1}^{N-w} \hat{\Gamma}_i. \quad (3.6)$$

The building structure of $\hat{\mathbf{M}}\Gamma$ allows it to measure the dynamic dependence of non-stationary series' components as it collects its information from the sample cross-covariance matrices defined over the updated local windows of \mathbf{y}_t . To be specific, the first cross-covariance matrix is calculated over the first window, then the second cross-covariance matrix is calculated over the second window (i.e. by including the next time point and excluding the oldest time point), and so on. Then $\hat{\mathbf{M}}\Gamma$ uses all these cross-covariance matrices to extract the non-stationary dynamic dependence from \mathbf{y}_t as a whole.

Furthermore, spectral decomposition can be used to decompose the moving cross-covariance matrix $\mathbf{M}\Gamma$ of \mathbf{y}_t as follows

$$\mathbf{M}\Gamma = \mathbf{U}\Lambda\mathbf{U}' \quad (3.7)$$

where \mathbf{U} is an $M \times M$ orthogonal matrix whose columns are the eigenvectors of $\mathbf{M}\Gamma$ and Λ is an $M \times M$ diagonal matrix consists of the eigenvalues of $\mathbf{M}\Gamma$ along its diagonal. Let λ_j , $1 \leq j \leq M$, be the j^{th} eigenvalue of $\mathbf{M}\Gamma$ (i.e. λ_j is the $(j, j)^{\text{th}}$ element of Λ), where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_M$. Let \mathbf{u}_j be the eigenvector corresponding the j^{th} eigenvalue of $\mathbf{M}\Gamma$ (i.e. \mathbf{u}_j is the j^{th} column of \mathbf{U}). MDPCA reduces the dimension of \mathbf{z}_t by producing M uncorrelated moving dynamic principal components (MDPCs) and transform \mathbf{y}_t into a space with dimension $k < m$ such that

$$\frac{\lambda_1 + \dots + \lambda_k}{\lambda_1 + \dots + \lambda_M} \simeq 1 . \quad (3.8)$$

Here, the value of k also indicates the number of MDPCs being used to reconstruct the data. The optimal value for k is the minimum number of MDPCs that consist of the maximum variation of the data and produce the minimum error when used to reconstruct the original data. More details about determining the optimal choice of k will be provided in the next section.

3.1.1 Optimizing MDPCA's Results

In order to improve the results of MDPCA, one would choose the optimal values for the window size W , the number of lags l to include in the extended data vector, and the number of retained MDPCs.

Choosing a size for W is vital to enhance the results of MDPCA and extract accurate information from the data. The size of W depends on the stationarity of the data. For stationary series, a wide window of size $W = N$ should be used. Shorter window sizes are suitable for data that exhibit stronger non-stationarity. For stationary series, we should choose the window size appropriately such that it is not too small and not too large. If it is too small, this could lead to a non-valid estimation of the local cross-covariance matrices. If it is too large, then the local cross-covariance matrices will not be able to capture the dynamic dependence of non-stationary series. Determining a size for W can be done by looking at the time series plot and assessing the stationarity of the data. More analyses on determining the size of W will be conducted in the simulations section of this chapter. Notice

that MDPCA can be applied on both stationary and non-stationary series by adjusting the window size, as mention earlier. Therefore, DPCA is a special case of MDPCA where $W = N$.

In the following, we are going to provide a procedure to determine the optimal size for l . Additionally, a new criterion will be proposed in order to objectively determine the number of optimal components (i.e. MDPCs) to retain.

3.1.1.1 Choosing Optimal Number of Lags

Including more lagged series can provide more dynamic information to the analysis; however, it would also increase the dimension of \mathbf{y}_t , which makes the analysis more complicated. Therefore, one would include only lagged series that provide more dynamic information related to the original series in order to gain accurate results with the lowest dimension possible for \mathbf{y}_t . In order to choose an optimal value for l , we are going to adapt the procedure suggested by Ku et al. (1995), which can be summarized as follows:

1. Start with $l = 0$.
2. Build the extended data vector \mathbf{y}_t by including l lagged series.
3. Apply MDPCA to \mathbf{y}_t and obtain all MDPCs.
4. Set $j = m(l + 1)$ and $r(l) = 0$ where r is the number of relations.
5. Determine if the j^{th} MDPC provides a linear relation. If yes, go to next step, otherwise go to step 7.
6. Set $j = j - 1$ and $r = r(l) + 1$, then repeat step 5.
7. Calculate the number of new relations added by including one more lag, $r_{\text{new}}(l)$, by:

$$r_{\text{new}}(l) = r(l) - \sum_{k=0}^{l-1} (l - k + 1)r_{\text{new}}(k)$$

if $r_{\text{new}}(l) \leq 0$, go to step 9, otherwise go to next step.

8. Set $l = l + 1$, go to step 2.

9. Stop.

The above steps assumed the size of W to be given or already determined. The number of significant MDPCs can be determined by examining the plot of the eigenvalues of $\hat{\mathbf{M}}\Gamma$. This can be done by excluding MDPCs that have small contribution percentages and retaining those with higher contribution percentages (i.e. consist of most variation of the original data). Then, the number of relations r can be obtained by subtracting the number of significant MDPCs from the total number of variables (i.e. $r = M - \text{number of significant MDPCs}$).

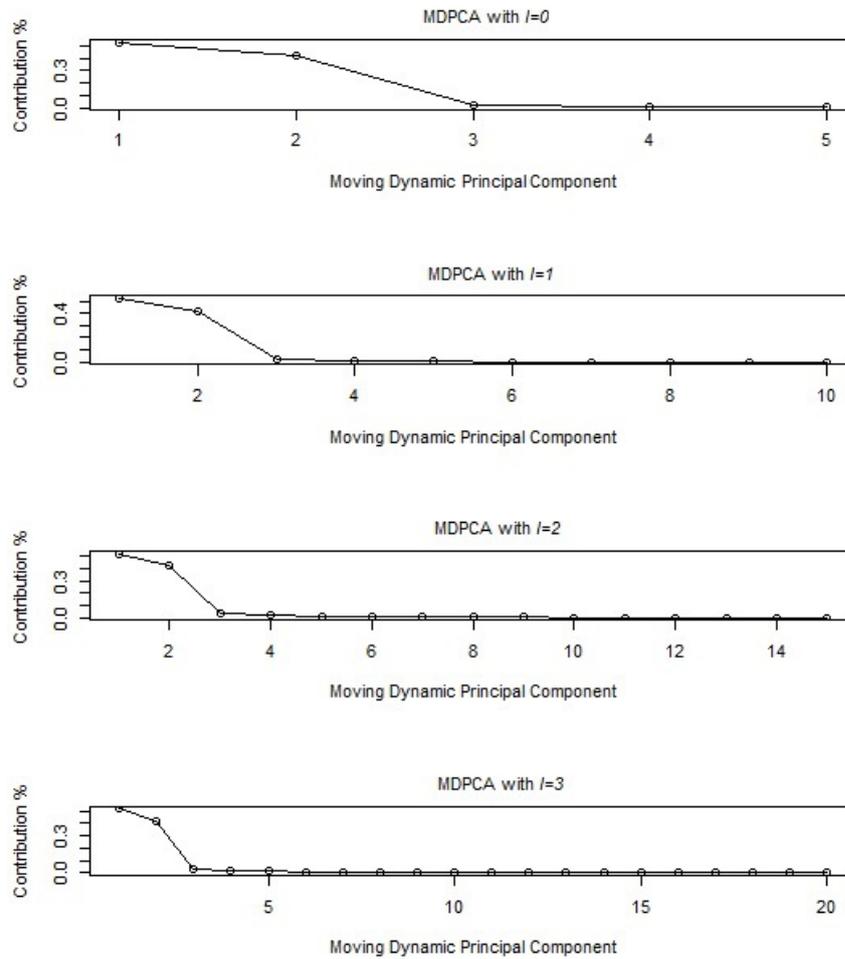


Figure 3.4: Eigenvalues' plots after applying MDPCA to a series of five variables with three static and two dynamic relations.

The following example can clarify the idea behind the above procedure. Let the series \mathbf{z}_t consists of five variables that have some relationships among them. The eigenvalue plot after applying MDPCA with 0, 1, 2, and 3 lags are shown in Figure 3.4. Three static relations are found when MDPCA with $l = 0$ is applied because only two MDPCs are significant. Notice that by including each lag, new relations might be detected, and the previous relations will be repeated $(l+1)$ times. By applying MDPCA with $l = 1$, eight relations are found, which are the three static relations repeated twice and two new dynamic relations that are exposed by including the first legged series. By using MDPCA with $l = 2$, 13 relations are found, which are the three static relations repeated three times, and two dynamic relations

repeated twice. Hence, no new relations are found, and the procedure would suggest not including more lags as $l = 1$ is the optimal choice.

Notice that in the above example, if these five variables are independent and do not have any relationship between them, then all MDPCs resulted from applying MDPCA will be significant. Therefore, no relation will be detected in this case (i.e. $r = 0$).

3.1.1.2 Retained Component Criterion (RCC)

Once W and l are already determined, and MDPCA is applied to \mathbf{y}_t , then the next task is to choose the optimal number of MDPCs to retain, k . This can be done by balancing between the following desires: maximizing the percentage of explained variance, minimizing the MSE (i.e., mean of squared error) of reconstructing the original data, and reducing the dimension of the series. The percentage of explained variance can be measured as given in equation (3.8). The following equation calculates the MSE of reconstructed data by the first k MDPCs:

$$\text{MSE}_k(\mathbf{y}_t, \mathbf{y}_t^{\text{recon}}) = \frac{1}{MN} \sum_{j=1}^M \sum_{t=1}^N (y_{j,t} - \sum_{v=1}^k u_{j,v} C_{v,t})^2 \quad (3.9)$$

where $\mathbf{y}_t^{\text{recon}}$ is the reconstructed data by the first k MDPCs and $C_{v,t}$ is the t^{th} observation of the v^{th} MDPC (C_v) which can be obtained by left multiplying \mathbf{y}_t by the transpose of the first v eigenvectors of \mathbf{U} . Notice that choosing more MDPCs might increase the percentage of explained variance and reduce the MSE of reconstructed data; however, it also would increase the final dimension. Therefore, our goal here is to retain the minimum number of MDPCs that explain most of the variation and have minimum reconstruction error. In literature, this is usually done subjectively by balancing between the above desires. To this end, we are going to propose a criterion that can balance between the above desires and objectively suggests the optimal number of MDPCs to retain. This criterion will be called the Retained Component Criterion (RCC).

In order to determine the optimal number of MDPCs, we need to measure the

effect of adding each MDPC on the accuracy of the final results of MDPCA, where maximum accuracy can be achieved by explaining all variations in the original data and reducing the MSE of reconstructed data to zero. Notice, we are going to assume that both the percentage of explained variance and the MSE of reconstructed data to be equally important in measuring the accuracy of MDPCA's results.

Consider the case where a time series \mathbf{y}_t with dimension M is observed. Assume an ideal case where all M variables are independent and equally important to explain the variability in the data. In particular, all variables consist of equally important information and contribute equally to the variation in \mathbf{y}_t . Then after applying MDPCA to \mathbf{y}_t , we expect that each MDPC will equally explain $1/M$ % of the total variation of \mathbf{y}_t and reduce the MSE of reconstructing the data by an equal amount of $1/M$ %. Therefore, each MDPC will improve the accuracy of the final results of MDPCA by $2/M$ %. The reason behind assuming an ideal case and giving the components of \mathbf{y}_t equal weights is to include an objective penalty term in our criterion for retaining an extra MDPC in the final results. Before we move further, consider the following definition. Let MaxMSE be the maximum MSE of reconstructing data defined by

$$\text{MaxMSE} = \frac{1}{MN} \sum_{j=1}^M \sum_{t=1}^N (y_{j,t})^2. \quad (3.10)$$

Notice that MaxMSE is equivalent to the MSE of reconstructing data with no MDPCs available and replacing elements of $\mathbf{y}_t^{\text{recon}}$ in (3.9) by zeros.

The RCC criterion of the first k MDPCs is defined as

$$\text{RCC}_k = 2 - \left(\sum_{j=1}^k \lambda_j / \sum_{j=1}^M \lambda_j \right) - ((\text{MaxMSE} - \text{MSE}_k) / \text{MaxMSE}) + (2k/M) \quad (3.11)$$

where λ_j is the j^{th} eigenvalue of the matrix $\mathbf{M}\mathbf{\Gamma}$ defined in (3.5), MaxMSE is defined in (3.10), and MSE_k is given in (3.9). The RCC criterion consists of three main terms: the term $(\sum_{j=1}^k \lambda_j / \sum_{j=1}^M \lambda_j)$, which represents the percentage of explained variance by first k MDPCs, the term $((\text{MaxMSE} - \text{MSE}_k) / \text{MaxMSE})$, which represents the percentage of reduced MSE by the first k MDPCs, and the term $(2k/M)$,

which is a penalty for retaining k MDPCs. The constant “2” is included in the calculation in (3.11) to retain positive values for the RCC criterion. This is a technical reason as the constant value will not change the final decision of the RCC criterion. The optimal number of MDPCs to retain is the number corresponding to the minimum RCC value in (3.11). Furthermore, the RCC criterion can be adapted to determine the optimal number of components in most of the PCA-based reduction methods (e.g., classical PCA and DPCA).

For example, consider a series \mathbf{y}_t with dimension $M = 8$. After applying MDPCA, if the first MDPC explains 50% of the total variation in the series \mathbf{y}_t (i.e., $(\sum_{j=1}^k \lambda_j / \sum_{j=1}^M \lambda_j) = 0.5$) and reduces MaxMSE by 85% (i.e., $((\text{MaxMSE} - \text{MSE}_k) / \text{MaxMSE}) = 0.85$), then the RCC criterion will have a value of $2 - 0.5 - 0.85 + 0.25 = 0.9$. Now, let the second MDPC explains 40% of the total variation of \mathbf{y}_t and reduces MaxMSE by 10%, then the RCC criterion of the first two MDPCs will have a value of $2 - (0.5 + 0.4) - (0.85 + 0.1) + (0.25 + 0.25) = 0.65$. Hence, adding the second component will contribute significantly to increase the accuracy of MDPCA’s results. Additionally, if the third MDPC explains 5% of the total variation of \mathbf{y}_t and reduces MaxMSE by 3%, then the RCC criterion of the first three MDPCs will have a value of $2 - (0.5 + 0.4 + 0.05) - (0.85 + 0.1 + 0.03) + (0.25 + 0.25 + 0.25) = 0.87$, which means that adding the third MDPC will increase the accuracy by a non-significant amount. This can be explained as the penalty of using the third MDPC is larger than the amount of accuracy added to MDPCA’s results. Hence, for this example, the optimal number of retained MDPCs will be 2, as it has the lowest RCC value of 0.65.

3.1.2 MDPCA Calculation Procedure

The following is a summary of the steps of MDPCA:

1. Create the extended data vector \mathbf{y}_t by including lagged series of the observed series \mathbf{z}_t up to lag l .
2. Calculate the sample moving cross-covariance matrix $\hat{\mathbf{M}}\Gamma$ based on \mathbf{y}_t , as

defined in (3.6).

3. Calculate the eigenvalues and the corresponding eigenvectors of $\hat{\mathbf{M}}\Gamma$.
4. Use the RCC criterion to determine k , the optimal number of MDPCs to retain.
5. Left multiplying \mathbf{y}_t by transpose of the first k columns of the matrix $\hat{\mathbf{U}}$ defined in (3.7) produces the transformed data with reduced dimension.

3.1.3 Evaluating Dynamic Relationships among MDPCs

For stationary series, examining for significant correlations between the variables of a multivariate time series can be done by visualizing tools such as the cross-correlation plots, which is a generalization of the autocorrelation function plot (ACF) of Box and Jenkins (1976) to the multivariate time series. Methods involving testing the significance of the multiple null hypotheses exist in literature such as the multivariate portmanteau statistic; See Hosking (1980). However, the above mentioned methods were developed to capture dynamic dependence of stationary series and would not be meaningful for non-stationary series because they use the classical cross-correlation function with a fixed mean throughout the calculations. To this end, we need to extend some of the methods mentioned above to find correlated components or variables of non-stationary series by using an advanced measurement of correlation that can be updated as we move forward or backward in time. Hence, we propose the use of a moving cross-correlation function. This function will be used to check whether two non-stationary variables are correlated. It will also be used to evaluate the relationship between MDPCs. Before we proceed further, the following definitions are needed. Define the lag l cross-covariance matrix of \mathbf{y}_i as

$$\Gamma_i(l) = \text{Cov}(\mathbf{y}_i, \mathbf{y}_{i-l}) . \quad (3.12)$$

Also, define the lag l cross-correlation matrix of \mathbf{y}_i to be

$$\rho_i(l) = \text{Corr}(\mathbf{y}_i, \mathbf{y}_{i-l}) = \mathbf{S}_i^{-1} \Gamma_i(l) \mathbf{S}_{i-l}^{-1} , \quad (3.13)$$

where l is a non-negative integer, $\Gamma_i(l)$ is defined in (3.12) and \mathbf{S}_i is the diagonal matrix of the standard deviations of \mathbf{y}_i . The $(j, j)^{th}$ element of \mathbf{S}_i is the square root of the $(j, j)^{th}$ element of $\Gamma_i(0)$ defined over \mathbf{y}_i . The above functions can be estimated using the following formulas as follows. The sample lag l cross-covariance matrix over window \mathbf{y}_i with a pre-specified size of $W = 2w + 1$ will be calculated using

$$\hat{\Gamma}_i(l) = \frac{1}{2w + 1} \sum_{t=i-w}^{i+w} (\mathbf{y}_t - \bar{\mathbf{y}}_i)(\mathbf{y}_{t-l} - \bar{\mathbf{y}}_i)' \quad (3.14)$$

where

$$\bar{\mathbf{y}}_i = \frac{1}{2w + 1} \sum_{t=i-w}^{i+w} \mathbf{y}_t .$$

Then, $\hat{\Gamma}_i(l)$ defined in (3.14) can be used to calculate the sample lag l moving cross-correlation matrix over the window \mathbf{y}_i , $\hat{\rho}_i(l)$, to estimate $\rho_i(l)$ as

$$\hat{\rho}_i(l) = \hat{\mathbf{S}}_i^{-1} \hat{\Gamma}_i(l) \hat{\mathbf{S}}_{i-l}^{-1} \quad (3.15)$$

where the $(j, j)^{th}$ element of $\hat{\mathbf{S}}_i$ is the square root of the $(j, j)^{th}$ element of $\hat{\Gamma}_i(0)$ defined over the same window \mathbf{y}_i . Further, define the lag l moving cross-correlation matrix of the series \mathbf{y}_t to be

$$\mathbf{M}\rho(l) = \frac{1}{N - 2 \max(l, w)} \sum_{i=\max(l, w)+1}^{N-\max(l, w)} \rho_i(l). \quad (3.16)$$

Based on sample data, we can estimate $\mathbf{M}\rho(l)$ using the sample lag l moving cross-correlation matrix as follows

$$\hat{\mathbf{M}}\rho(l) = \frac{1}{N - 2 \max(l, w)} \sum_{i=\max(l, w)+1}^{N-\max(l, w)} \hat{\rho}_i(l). \quad (3.17)$$

Notice that $\hat{\mathbf{M}}\rho(l)$ will be updated at each time point as we move in time to account for non-stationary series.

Based on the above-stated definitions, both visualization and multiple hypotheses testing methods can be developed to check for the significance of correlations between the components of either stationary or non-stationary series. For visualization, one can plot the sample moving cross-correlation matrices with different time lags

$l = 0, \pm 1, \pm 2, \dots, \pm p$; where p is a positive integer taken to be $p = 10 \log_{10}(N/M)$, similarly to those in ACF plot. The significance of the correlation can be evaluated by looking at the 95% confidence interval computed using $\pm 1.96/\sqrt{N}$. We demonstrate the use of the moving cross-correlation function in the following examples.

Example 3.1.2 This example is a short simulation study to test the ability of the moving cross-correlation plots to capture the dynamic relationship among different variables of a multivariate time series. A window with size 101 will be used in the calculation of the moving cross-correlation function. The results then will be compared with those based on the cross-correlation function.

The simulated data in this example consists of eight variables and a sample of size 1200, where three different non-stationary models were used to generate three subseries of 4, 3, and 1 variable as described below. Let a_t , b_t , and c_t be three independent standard normal white noises, which are the innovation terms of the following three models, respectively, then:

$$\begin{cases} y_{j,t} = u_{t+j-1}, & j = 1, 2, 3 \text{ and } 4 \\ y_{j,t} = v_{t+j-4}, & j = 5, 6 \text{ and } 7 \\ y_{j,t} = w_t, & j = 8 \end{cases} \quad (3.18)$$

where $(u_t, v_t, w_t)'$ satisfies

$$\begin{cases} \nabla u_t = 0.6 \nabla u_{t-1} + a_t + 0.5a_{t-1} \\ \nabla v_t = b_t + 0.8b_{t-1} - 2.2b_{t-2} \\ \nabla w_t = -0.55 \nabla w_{t-1} - 0.5 \nabla w_{t-2} + 0.4 \nabla w_{t-3} + c_t \end{cases} \quad (3.19)$$

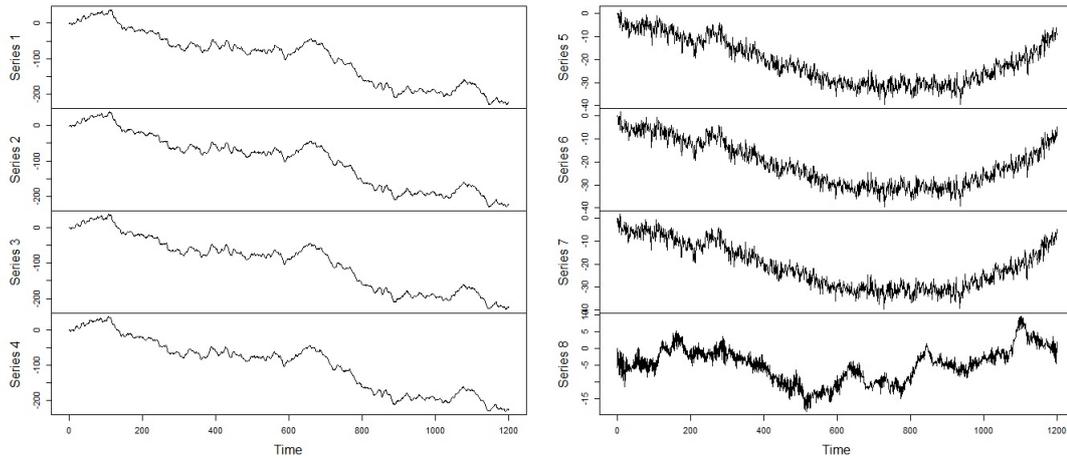
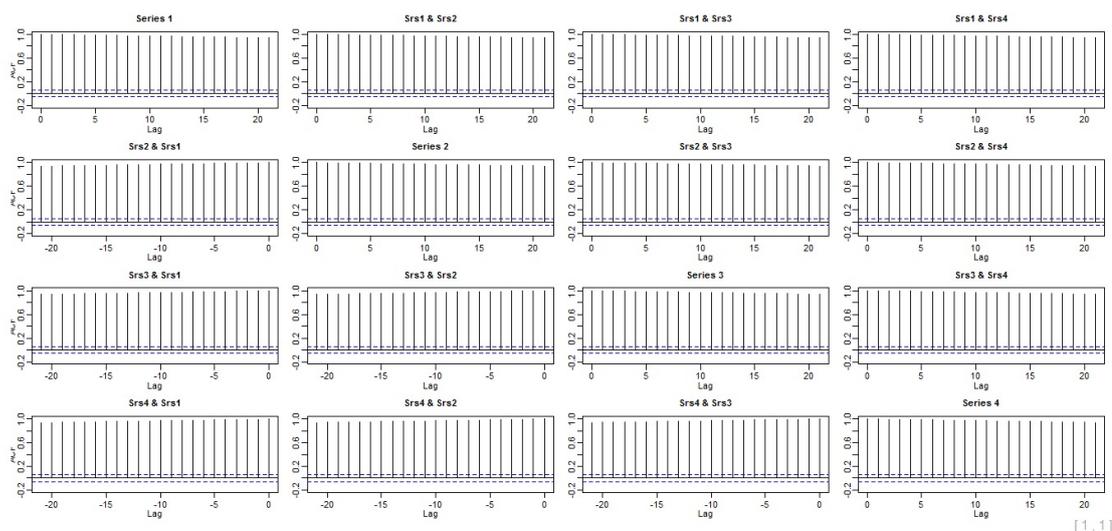


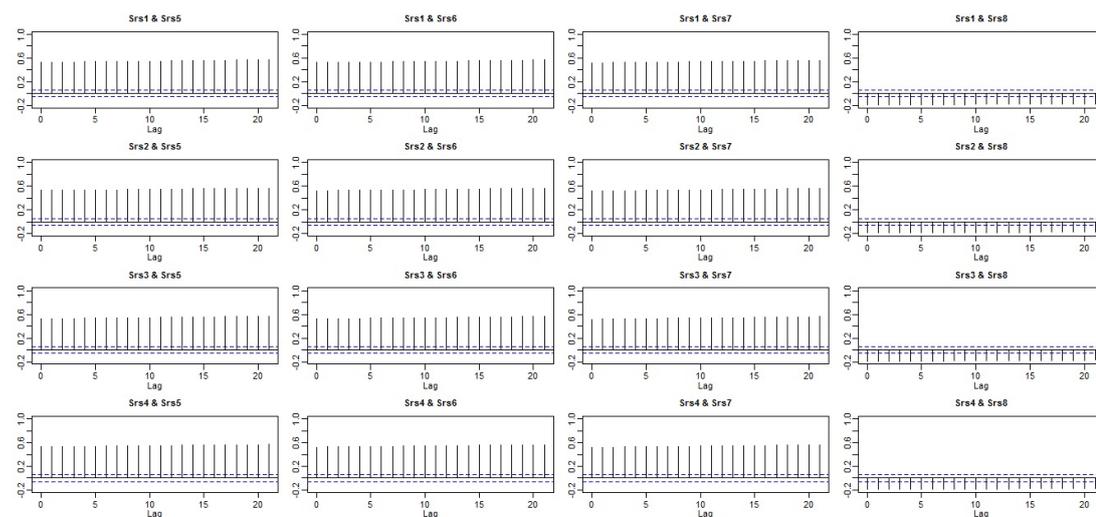
Figure 3.5: A time series plot of the simulated series with eight variables in Example 3.1.2.

A time series plot of the simulated multivariate time series is available in Figure 3.5. First, we examine the sample cross-correlation plots (i.e., using the classical cross-correlation function) of the data; See figures 3.6, 3.7, 3.8 and 3.9. Based on these plots, a strong dynamic relationship exists among the eight variables, which implies that all three simulated subgroups are strongly correlated with each other. The last result contradicts with the way that we simulated the data. Therefore, the cross-correlation plots could lead to non-correct results when dealing with non-stationary series.



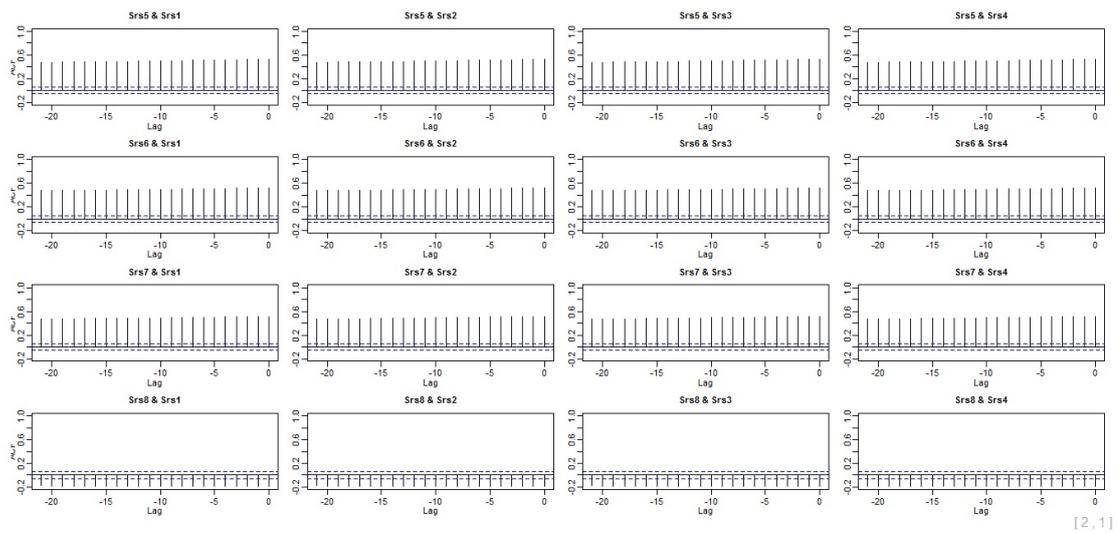
[1, 1]

Figure 3.6: Sample cross-correlation plots of the simulated series with eight variables in Example 3.1.2 - Part 1.



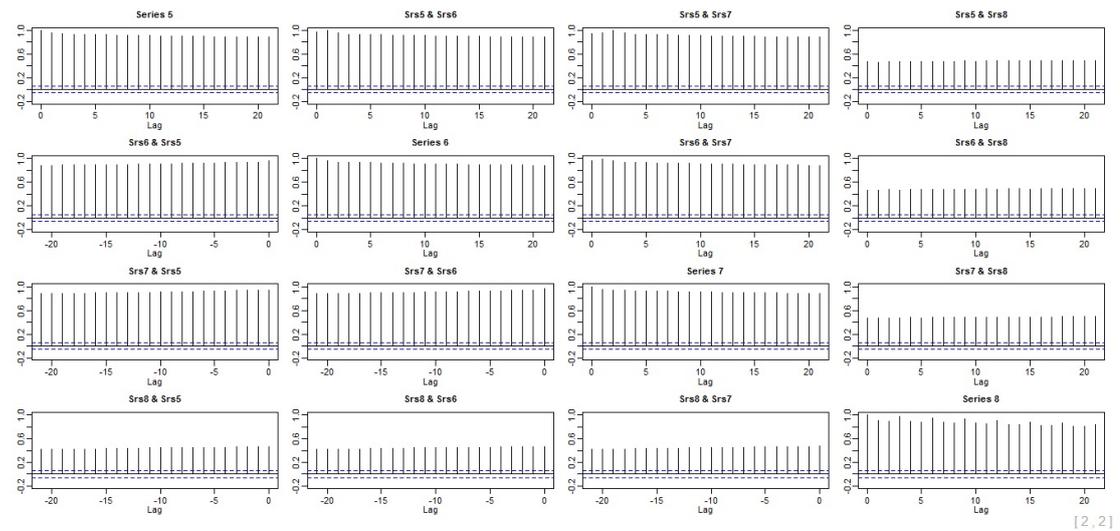
[1, 2]

Figure 3.7: Sample cross-correlation plots of the simulated series with eight variables in Example 3.1.2 - Part 2.



[2, 1]

Figure 3.8: Sample cross-correlation plots of the simulated series with eight variables in Example 3.1.2 - Part 3.



[2, 2]

Figure 3.9: Sample cross-correlation plots of the simulated series with eight variables in Example 3.1.2 - Part 4.

The sample moving cross-correlation plots of the simulated series are provided in figures 3.10, 3.11, 3.12, and 3.13. The moving cross-correlation function was able to capture the correct dynamic relationship among the simulated variables in each of the three different subgroups. The results of the two visualization methods are summarized in Table 3.1.

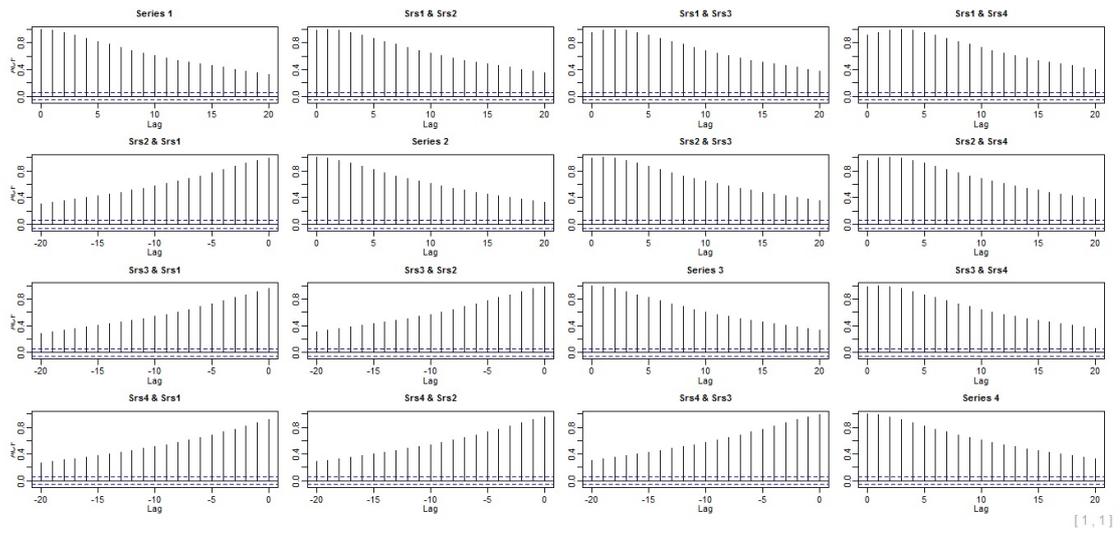


Figure 3.10: Sample moving cross-correlation plots with a window size 101 of the simulated series with eight variables in Example 3.1.2 - part 1.

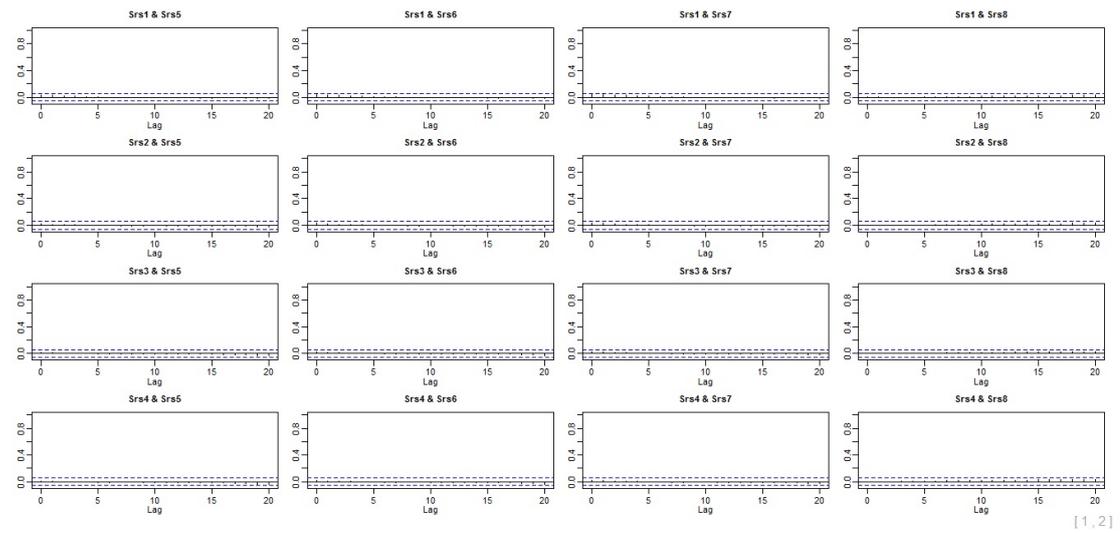


Figure 3.11: Sample moving cross-correlation plots with a window size 101 of the simulated series with eight variables in Example 3.1.2 - part 2.

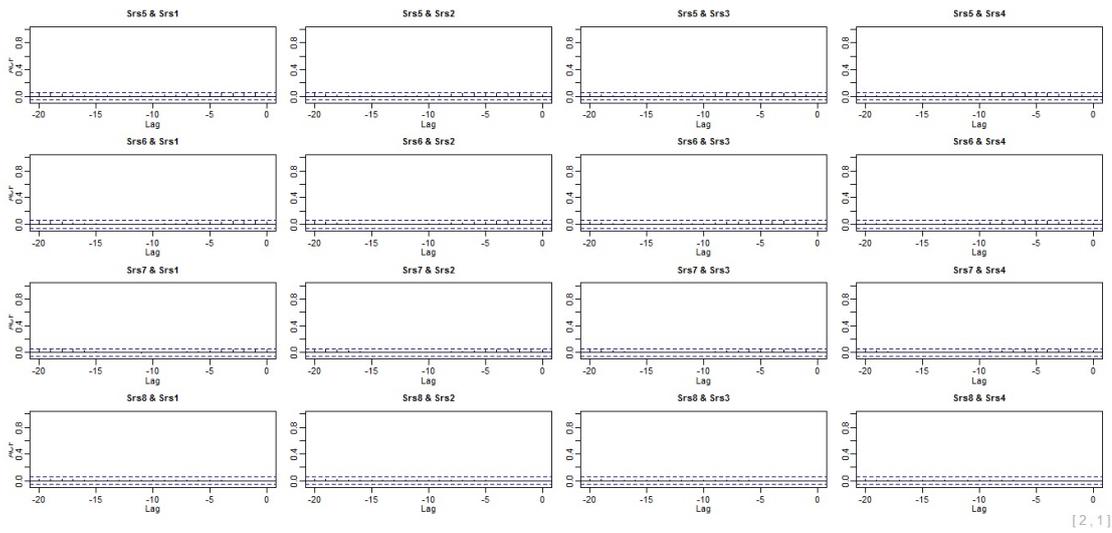


Figure 3.12: Sample moving cross-correlation plots with a window size 101 of the simulated series with eight variables in Example 3.1.2 - part 3.

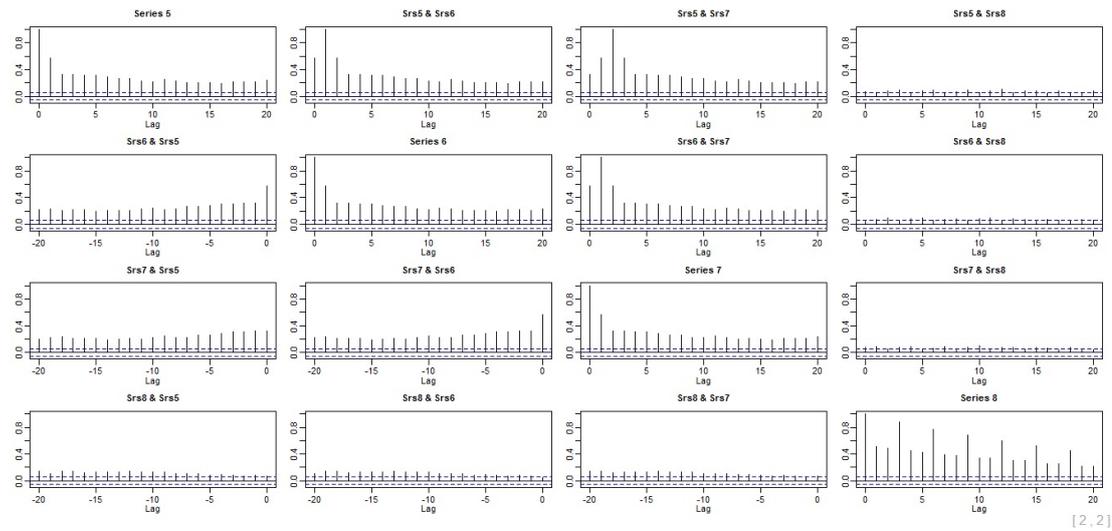


Figure 3.13: Sample moving cross-correlation plots with a window size 101 of the simulated series with eight variables in Example 3.1.2 - part 4.

Results	Correlated Variables
Visualization Method	
Moving Cross-Correlation Plots (window=101)	{1, 2, 3, 4}, {5, 6, 7} and {8}
Cross-Correlation Plots	{1, 2, 3, 4, 5, 6, 7, 8}

Table 3.1: Correlated variables of the simulated data with eight variables in Example 3.1.2 based on different visualization methods

The main reason that the results are different using the two visualization methods above is the non-stationary nature of the data. Therefore, we can conclude that the moving cross-correlation plots are more suitable to capture the dynamic relationship between non-stationary series. Also, it can be shown that the above two methods will produce similar conclusions when applied to stationary series.

3.2 Theoretical Properties

To show the reliability of the results obtained by the proposed MDPCA, we shall prove the consistency of its estimated moving dynamic principal components (i.e. MDPCs) based on sample data. Recall that the estimated MDPCs are generated by left multiplying the extended data matrix \mathbf{Y} by $\hat{\mathbf{U}}'$, where $\hat{\mathbf{U}}$ is the matrix that consists of the eigenvectors of $\hat{\mathbf{M}}\Gamma$ in (3.6) and used to estimate \mathbf{U} . Therefore, we shall show that these eigenvectors are consistent. In other words, we are going to show that $\hat{\mathbf{U}}$ is a consistent estimator of \mathbf{U} . One approach to show the consistency is by showing that $D(\mathcal{M}(\hat{\mathbf{U}}), \mathcal{M}(\mathbf{U})) \rightarrow 0$ as $W \rightarrow \infty$. Here, $\mathcal{M}(\mathbf{U})$ is the linear space spanned by \mathbf{U} 's columns, and $D(\mathcal{M}(\hat{\mathbf{U}}), \mathcal{M}(\mathbf{U}))$ is the distance between the spaces $\mathcal{M}(\hat{\mathbf{U}})$ and $\mathcal{M}(\mathbf{U})$. For ease of notation, we are going to use c, c_1, c_2, \dots to denote constants that their values might be different from place to place.

For $c_1 < c_2$, let \mathbf{B}_1 and \mathbf{B}_2 be any $c_2 \times (c_2 - c_1)$ matrices satisfying the condition $\mathbf{B}'_i \mathbf{B}_i = \mathbf{I}_{(c_2 - c_1) \times (c_2 - c_1)}$, where $i \in \{1, 2\}$. Define the distance between the \mathbf{B}_1 and \mathbf{B}_2 to be

$$D(\mathbf{B}_1, \mathbf{B}_2) = \sqrt{c_2 - c_1 - \text{tr}(\mathbf{B}_1 \mathbf{B}'_1 \mathbf{B}_2 \mathbf{B}'_2)}. \quad (3.20)$$

Notice that $D(\mathbf{B}_1, \mathbf{B}_2) = 0$ if and only if $\mathcal{M}(\mathbf{B}_1) = \mathcal{M}(\mathbf{B}_2)$. This measurement was used in Pan and Yao (2008) and Chang et al. (2018).

It is important to know that the convergence of the estimator $\hat{\mathbf{U}}$ is implied by the convergence of the moving cross-covariance matrix $\hat{\mathbf{M}}\Gamma$. This can be seen since $\hat{\mathbf{U}}$ consists of the eigenvectors of $\hat{\mathbf{M}}\Gamma$. The fact that $\mathbf{M}\Gamma$ is calculated based on moving windows whose width depends on the stationarity of the data makes the moving

cross-covariance function more complicated than the ordinary cross-covariance function. However, the consistency of $\hat{\mathbf{M}}\Gamma$ still can be reached as each window in $\hat{\mathbf{M}}\Gamma$ is calculated as in the stationary case. Therefore, the convergence of the estimated moving principal components MDPCs will depend on the size of W . Recall that $W \leq N$, where $W = N$ when data is stationary, and W gets smaller as the data becomes more non-stationary.

In what follows, the asymptotic properties of our estimator will be shown where the dimension of the extended data vector M is fixed. The needed conditions will be stated. We are going to follow a similar framework to those of Chang et al. (2018). Since time series data is known to be dependent data, we are going to consider the following measurement of dependence:

$$\theta_l = \sup_c \sup_{A \in \mathcal{F}_{-\infty}^c, B \in \mathcal{F}_{c+l}^\infty} |P(A \cap B) - P(A)P(B)| \quad (3.21)$$

where $\mathcal{F}_{c_3}^{c_4}$ is the σ -field generated by \mathbf{y}_t for $c_3 \leq t \leq c_4$. The measurement of dependence θ_l in (3.21) is called the mixing coefficients in the literature and $\theta_l = 0$ if A and B are independent. It indicates the asymptotic independence, where two data points which are l times apart will be independent as $l \rightarrow \infty$. More info on the use of the mixing coefficient can be found in Bradley (1986).

Assumption 3.1. Assume that

$$\sup_i \max_{1 \leq j \leq M} E|y_{j,i} - E(y_{j,i})|^{2q}$$

to be upper bounded by a positive constant c for some constant $q > 2$, where $y_{j,i}$ is the j^{th} component of \mathbf{y}_i .

Assumption 3.1 requires the tail probabilities of the variables of \mathbf{y}_i to decay fast enough such that $\hat{\Gamma}_i$ converges to Γ_i with a fast rate. For example, let $y_{j,i}$ follows $unif(0, 1 + \frac{1}{i} + \frac{1}{j})$, then Assumption 3.1 above is justified for $y_{j,i}$.

Assumption 3.2. Assume that

$$\sum_{l=1}^{\infty} \theta_l^{1-2/q} < \infty$$

for q defined in Assumption 3.1, where θ_l is the mixing coefficient defined in (3.21).

Assumption 3.2 above guarantees that θ_l goes to zero fast enough.

Theorem 3.1. Assume the dimension M is fixed. Let assumptions 3.1 and 3.2 hold. Then, there exists $\hat{\mathbf{U}} = (\hat{\mathbf{U}}_1, \dots, \hat{\mathbf{U}}_M)$ such that

$$\max_{1 \leq j \leq M} D(\mathcal{M}(\hat{\mathbf{U}}_j), \mathcal{M}(\mathbf{U}_j)) = O_p(1/\sqrt{W})$$

as $W \rightarrow \infty$.

In order to prove Theorem 3.1, three lemmas are needed. The following lemma of Chang et al. (2018) will be used to establish our results.

Lemma 3.1. For a stationary series $y_{j,t}$, assume for $\gamma > 2$, $E(|y_{j,t} - \mu_j|^{2\gamma})$ is uniformly bounded away from infinity for $j = 1, \dots, p$ and $t \geq 1$. Assume the mixing coefficients

$$\theta_l = \sup_j \sup_{A \in \mathcal{F}_{-\infty}^j, B \in \mathcal{F}_{j+l}^\infty} (|P(A \cap B) - P(A)P(B)|)$$

satisfy the condition $\sum_{l=1}^{\infty} \theta_l^{1-2/\gamma} < \infty$. Assume the dimension p is fixed, then

$$\|\hat{\Sigma}_y(l) - \Sigma_y(l)\|_2 = O_p(1/\sqrt{T})$$

for all $l \leq l_1$, where $\Sigma_y(l)$ is the cross-covariance matrix of the series \mathbf{y}_t and T is its sample size.

The following lemma is based on Lemma 3.1.

Lemma 3.2. Let the assumptions 3.1 and 3.2 hold. Also, if we assume the dimension M to be fixed, then $\|\hat{\Gamma}_i - \Gamma_i\|_2 = O_p(1/\sqrt{W})$ as $W \rightarrow \infty$, where Γ_i and $\hat{\Gamma}_i$ are defined in (3.3) and (3.4), respectively.

Proof of Lemma 3.2. By assuming assumptions 3.1 and 3.2, then Lemma 3.1 can be applied to each window, and we have

$$\max_{w+1 \leq i \leq N-w} \|\hat{\Gamma}_i(l) - \Gamma_i(l)\|_2 = O_p(1/\sqrt{W}).$$

This also implies that

$$\|\hat{\Gamma}_i(l) - \Gamma_i(l)\|_2 = O_p(1/\sqrt{W}).$$

This result holds for the lagged covariance matrices up to a specified lag l_1 . Thus, this result holds when $l = 0$. Notice, in MDPCA we are only considering cross-correlation with no lags (i.e. $\Gamma_i = \Gamma_i(0)$), then the following is true

$$\|\hat{\Gamma}_i - \Gamma_i\|_2 = O_p(1/\sqrt{W}).$$

Therefore, we have the following lemma based on the results from Lemma 3.2.

Lemma 3.3. *Assume the dimension M is fixed and under the assumptions 3.1 and 3.2, then the following holds as $W \rightarrow \infty$*

$$\|\hat{\mathbf{M}}\Gamma - \mathbf{M}\Gamma\|_2 = O_p(1/\sqrt{W}),$$

where $\mathbf{M}\Gamma$ and $\hat{\mathbf{M}}\Gamma$ are defined in (3.5) and (3.6), respectively.

Proof of Lemma 3.3 By assuming the dimension M to be fixed, and under assumptions 3.1 and 3.2, we have the following result:

$$\begin{aligned} \|\hat{\mathbf{M}}\Gamma - \mathbf{M}\Gamma\|_2 &= \left\| \frac{1}{N-2w} \sum_{i=w+1}^{N-w} \hat{\Gamma}_i - \frac{1}{N-2w} \sum_{i=w+1}^{N-w} \Gamma_i \right\|_2 \\ &\leq \max_{w+1 \leq i \leq N-w} \left\| \frac{N-2w}{N-2w} \hat{\Gamma}_i - \frac{N-2w}{N-2w} \Gamma_i \right\|_2 \\ &= O_p(1/\sqrt{W}). \end{aligned}$$

The last inequality holds because from Lemma 3.2 as

$$\max_{w+1 \leq i \leq N-w} \|\hat{\Gamma}_i - \Gamma_i\|_2 = O_p(1/\sqrt{W}).$$

We are now ready to prove Theorem 3.1.

Proof of Theorem 3.1. Assume the dimension M is fixed. Let the assumptions 3.1 and 3.2 hold. By applying the results of Theorem 8.1.10 of Golub and Van Loan (1996) and Remark 1 of Lemma 1 in Chang et al. (2018), then

$$\max_{1 \leq j \leq M} D(\mathcal{M}(\hat{\mathbf{U}}_j), \mathcal{M}(\mathbf{U}_j)) = O_p(\|\hat{\mathbf{M}}\Gamma - \mathbf{M}\Gamma\|_2)$$

Therefore, if we apply the results from Lemma 3.3, then it holds that

$$\max_{1 \leq j \leq M} D(\mathcal{M}(\hat{\mathbf{U}}_j), \mathcal{M}(\mathbf{U}_j)) = O_p(1/\sqrt{W})$$

as $W \rightarrow \infty$.

3.3 Simulations and Real Data Examples

In this section, we are going to test the ability of the proposed method on both real and simulated data sets. The method will be applied to different data sets with various dimensional sizes. The following examples will focus on non-stationary data. It can be seen that MDPCA and DPCA will produce identical results when applied to stationary data since MDPCA uses a window of size $W = N$ in the stationary case. The following studies will justify the generality and suitability of the MDPCA to reduce the dimensions of multivariate time series data. The performance of MDPCA will be assessed by considering the percentage of explained variance (i.e. contribution percentage), the MSE of reconstructed data, and the moving cross-correlation plots of the retained MDPCs. Notice that, the moving cross-correlation function with window $W = 101$ will be used to evaluate the correlation between the retained MDPCs from MDPCA with different window sizes. Using a small window size (e.g. $W = 101$) in the moving cross-correlation function enables to capture non-stationary behaviours in the retained MDPCs that would not be captured when using larger window sizes. All analyses are done using R software. We developed the needed functions to produce and assess MDPCA's results. These functions can be found under the following R packages (i.e. libraries): `MACF` of Alshammri (2020b), `MCOV` of Alshammri (2020c), `MDPCA` of Alshammri (2020d) and `RCC_MDPCA` of Alshammri (2020g).

3.3.1 Simulations

In the following simulation studies, we are going to apply MDPCA with different combinations of window and lag sizes on simulated data sets of different dimensions and sample sizes. Each simulation will be replicated 500 times and the retained MDPCs will be assessed. Data sets will be generated using `arma.sim` command in R.

Example 3.3.1 In this example, we apply MDPCA to a non-stationary series \mathbf{z}_t with three variables and 1000 samples generated using the models in Example 3.1.1. This example consists of three parts. The first part compares the results of MDPCA

when using different combinations of W and l . The second part studies the effect of the size of T on MDPCA's results. The third part focuses on optimizing the results of MDPCA.

Table 3.2 is a comparison between the results of applying MDPCA with different choices of W and l based on 500 replicas, where two MDPCs are retained. The mean percentages of explained variance by two MDPCs were slightly higher for the cases where one lag was used. For example, on average, two MDPCs explained 96.06% of the variance of the data when $W = 101$ and $l = 1$ were used, compared with 93.98% when $W = 101$ and $l = 5$ were used. It can be seen that those percentages increased by increasing the window size. For example, by using $W = 301$ and $l = 5$, we obtained a mean percentage of explained variance of 96.10%, which is higher than what we obtained when using $W = 101$ and $l = 5$. The mean of MSE of reconstructed data varied from 53.16 to 67.31 for the considered choices of W and l , where it was at its lowest when $W = 301$ and $l = 5$ were used.

Additionally, the means of the absolute value of the moving cross-correlation with $W = 101$ are plotted for each case in figures 3.14 - 3.16, and the corresponding standard deviations are reported in Table 3.3. It can be seen that significant correlations existed between the two MDPCs when MDPCA with $W = 301$ was used. The correlations became smaller and less significant when reducing the window size to $W = 201$. The two MDPCs became uncorrelated when reducing the window size further to $W = 101$. Notice that the correlations for larger lags (i.e. $|l| \geq 15$) started to cross the significant line, as shown Figure 3.14. This is caused by the continuous one-step movement of the moving cross-correlation function and does not imply correlated components. Also, similar moving cross-correlation plots were observed across all 500 replicas as the standard deviations ranged between 0 to 0.12, which is very close to zero; See Table 3.3.

Window Size (W)	$W=101$ ($l=1$)	$W=201$ ($l=1$)	$W=301$ ($l=1$)
Results			
Contribution % Mean (Standard Deviation)	96.06 % (0.01)	96.50 % (0.02)	96.75 % (0.02)
MSE Mean (Standard Deviation)	58.95 (65.23)	54.27 (63.01)	56.76 (65.26)
Window Size (W)	$W=101$ ($l=5$)	$W=201$ ($l=5$)	$W=301$ ($l=5$)
Results			
Contribution % Mean (Standard Deviation)	93.98 % (0.02)	95.35 % (0.02)	96.10 % (0.02)
MSE Mean (Standard Deviation)	67.31 (78.97)	59.68 (77.71)	53.16 (64.30)

Table 3.2: A comparison between the results of MDPCA with different combinations of W and l in Example 3.3.1.

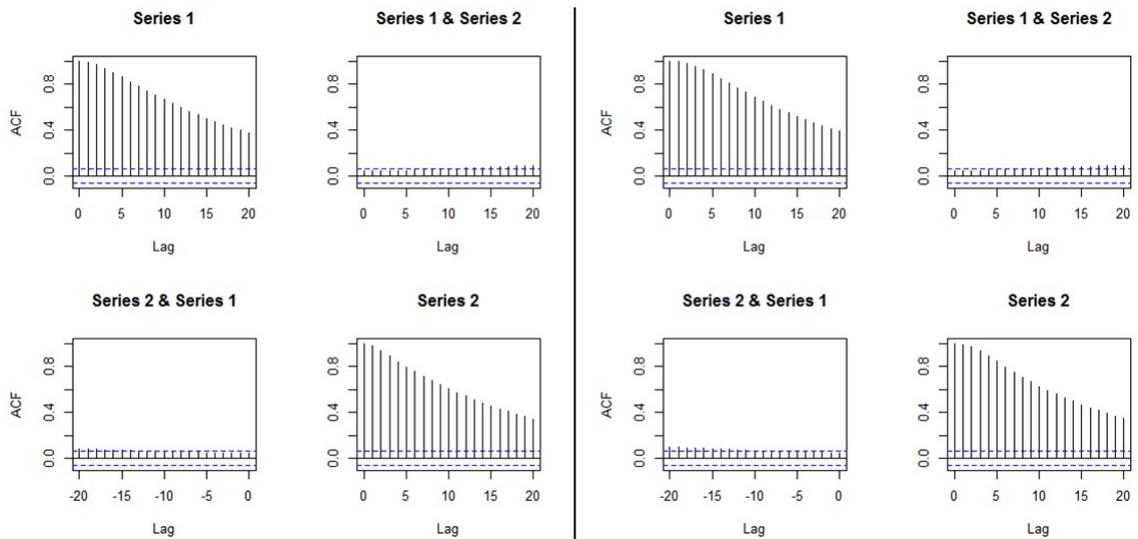


Figure 3.14: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.1 [Left: MDPCA ($W = 101, l = 1$), Right: MDPCA($W = 101, l = 5$)].

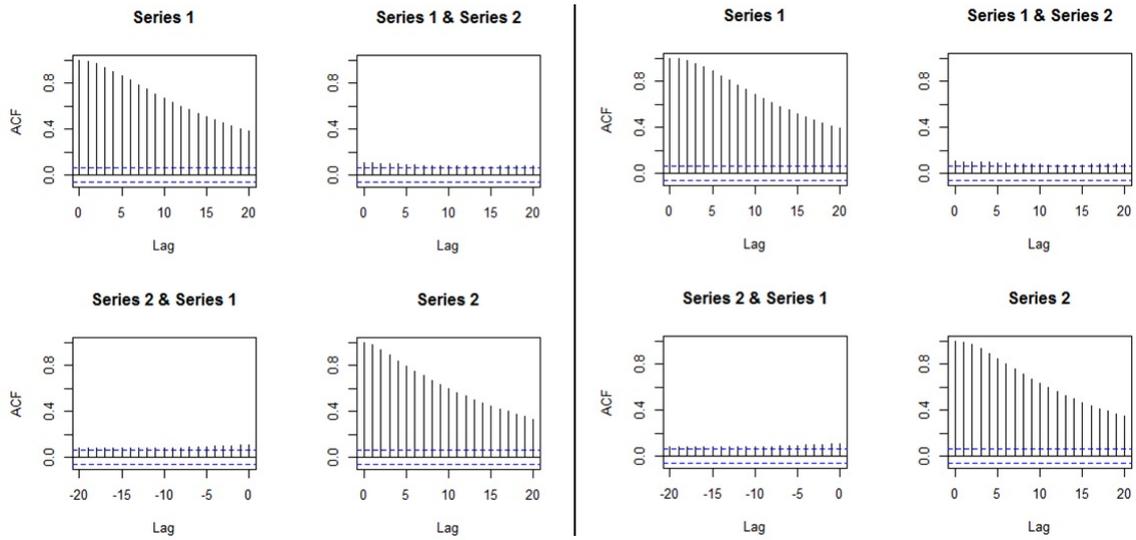


Figure 3.15: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.1 [Left: MDPCA ($W = 201$, $l = 1$), Right: MDPCA($W = 201$, $l = 5$)].

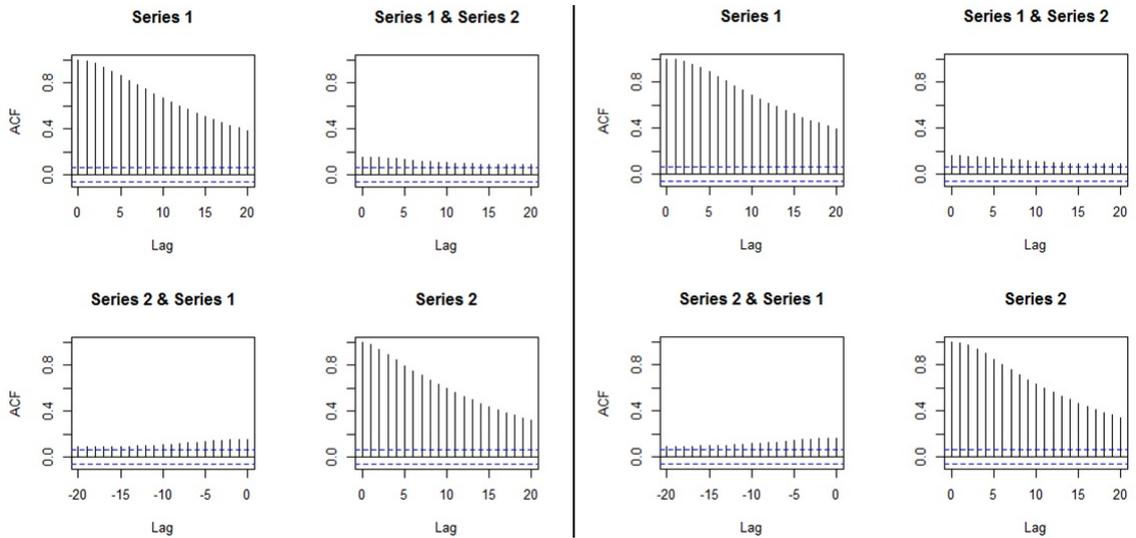


Figure 3.16: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.1 [Left: MDPCA ($W = 301$, $l = 1$), Right: MDPCA($W = 301$, $l = 5$)].

		MDPCA(W=101, l=1)																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09	0.10	0.10
MDPC1 vs MDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07
MDPC2 vs MDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
MDPC2 vs MDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09	0.10	0.10	0.10
		MDPCA(W=201, l=1)																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.05	0.05	0.06	0.07	0.07	0.08	0.08	0.09	0.09	0.10	0.10	0.10	0.10	0.10
MDPC1 vs MDPC2	0.08	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC1	0.08	0.08	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09	0.10
		MDPCA(W=301, l=1)																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09
MDPC1 vs MDPC2	0.12	0.12	0.11	0.11	0.11	0.10	0.10	0.10	0.10	0.09	0.09	0.09	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
MDPC2 vs MDPC1	0.12	0.11	0.11	0.11	0.11	0.10	0.10	0.10	0.09	0.09	0.09	0.08	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.07
MDPC2 vs MDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.05	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09	0.09
		MDPCA(W=101, l=5)																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.02	0.02	0.02	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.08	0.08	0.08	0.09	0.09	0.10	0.10	0.10
MDPC1 vs MDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07
MDPC2 vs MDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07
MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.05	0.05	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.10
		MDPCA(W=201, l=5)																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.02	0.02	0.03	0.03	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.09	0.09	0.09	0.09	0.09	0.10	0.10
MDPC1 vs MDPC2	0.08	0.08	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
MDPC2 vs MDPC1	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.05	0.05	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09	0.10	0.10
		MDPCA(W=301, l=5)																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.02	0.02	0.03	0.03	0.04	0.05	0.06	0.06	0.07	0.08	0.08	0.09	0.09	0.09	0.09	0.09	0.10	0.10
MDPC1 vs MDPC2	0.12	0.12	0.11	0.11	0.11	0.11	0.10	0.10	0.10	0.10	0.09	0.09	0.09	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
MDPC2 vs MDPC1	0.12	0.12	0.11	0.11	0.11	0.11	0.10	0.10	0.10	0.09	0.09	0.09	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.05	0.05	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09	0.10	0.10

Table 3.3: Standard deviations of the absolute value of the moving cross-correlations between two MDPCs with different combinations of W and l in Example 3.3.1.

Therefore, based on the above results and discussion, one can reduce the dimension of the simulated series in this example by using MDPCA with $W = 101$ and $l = 1$. Alternatively, MDPCA with $W = 101$ and $l = 5$ can be used, which might add more dynamic information to the analysis. However, the results of using $l = 5$ produced a slightly lower percentage of explained variance and a higher reconstructing error compared with those using $l = 1$.

A further investigation can be done to see the effect of the sample size T on the performance of MDPCA. Recall that the above results were obtained by using $T = 1000$. Based on 500 replicas, Table 3.4 summarizes the results of applying MDPCA with $W = 101$ and $l = 1$ to the data with different sample sizes, where two MDPCs are retained in all cases. For the mean of explained variance percentage, similar performance was obtained by applying MDPCA to the data with different sizes of T . For example, the highest percentage of 96.47% was obtained for the case where $T = 200$, compared with the lowest percentage of 96.07% for the case where $T = 800$. These percentages did not differ across the 500 replicas as they had standard deviations that ranged between 0.01 and 0.03. For the mean of MSE of reconstructed data, we noticed an increase in its value when increasing T . This can be explained as a larger error is expected when reconstructing more data. For example, the highest MSE of 49.46 was obtained when $T = 800$, compared with the lowest MSE of 10.37 when $T = 200$.

Also, the means of the absolute value of the moving cross-correlation with $W = 101$ are plotted for each sample size in figures 3.17 and 3.18, where the corresponding standard deviations are reported in Table 3.5. We noticed that some significant dynamic relationships between the two MDPCs existed when using $T = 200$ for $|l| \geq 10$. These correlations became non-significant when increasing the sample size to $T = 400$. The standard deviations also had a negative relationship with the sample size. For example, the standard deviations ranged between 0 to 0.24 when $T = 200$, 0 to 0.16 when $T = 400$, 0 to 0.13 when $T = 600$ and 0 to 0.12 when $T = 800$. Thus, MDPCA performed well when increasing the sample size.

Notice that the performance of MDPCA was reasonable for small sample sizes (i.e. $T = 400$).

Method	MDPCA ($W=101, l=1$)			
Sample size	T=200	T=400	T=600	T=800
Contribution %	96.47 %	96.32 %	96.25 %	96.07 %
Mean (Standard Deviation)	(0.03)	(0.02)	(0.01)	(0.02)
MSE	10.37	21.80	34.22	49.46
Mean (Standard Deviation)	(13.46)	(25.18)	(38.20)	(58.01)

Table 3.4: A comparison between the results of MDPCA with different sizes of T in Example 3.3.1.

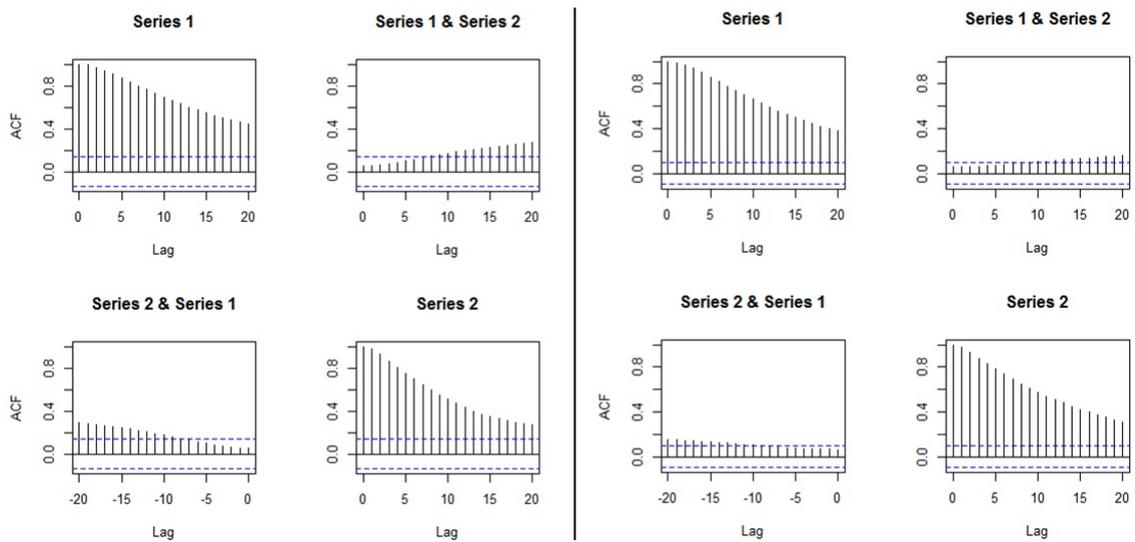


Figure 3.17: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.1 [Left: $T = 200$, Right: $T = 400$].

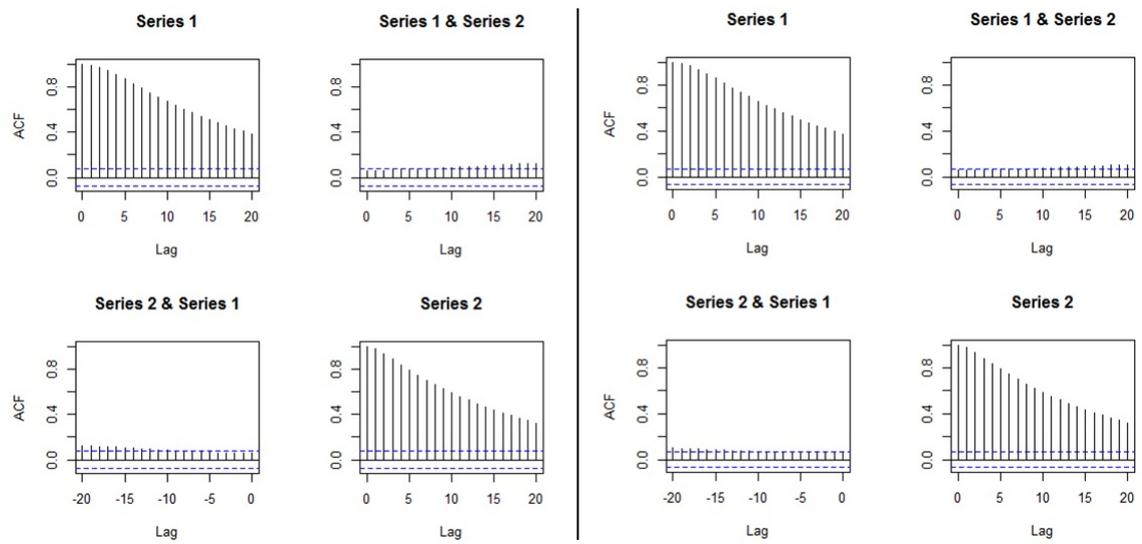


Figure 3.18: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.1 [Left: $T = 600$, Right: $T = 800$].

T=200																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.02	0.03	0.05	0.07	0.09	0.11	0.13	0.14	0.16	0.18	0.19	0.20	0.21	0.22	0.22	0.23	0.23	0.23	0.24
MDPC1 vs MDPC2	0.05	0.05	0.06	0.06	0.07	0.08	0.09	0.10	0.11	0.12	0.13	0.14	0.14	0.15	0.16	0.17	0.17	0.18	0.18	0.19	0.19
MDPC2 vs MDPC1	0.05	0.05	0.05	0.06	0.07	0.07	0.08	0.09	0.10	0.11	0.12	0.13	0.13	0.14	0.15	0.16	0.16	0.17	0.17	0.18	0.18
MDPC2 vs MDPC2	0.00	0.01	0.04	0.06	0.08	0.11	0.13	0.15	0.17	0.18	0.19	0.20	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.20	0.20
T=400																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.02	0.03	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.12	0.13	0.13	0.14	0.15	0.15	0.15	0.16	0.16
MDPC1 vs MDPC2	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.08	0.09	0.09	0.10	0.10	0.10	0.11	0.11	0.11	0.11
MDPC2 vs MDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.09	0.09	0.10	0.10	0.10	0.11	0.11	0.11
MDPC2 vs MDPC2	0.00	0.01	0.02	0.04	0.05	0.07	0.08	0.09	0.10	0.11	0.12	0.13	0.13	0.14	0.14	0.15	0.15	0.15	0.15	0.15	0.15
T=600																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.02	0.03	0.04	0.05	0.05	0.06	0.07	0.08	0.09	0.09	0.10	0.10	0.11	0.11	0.12	0.12	0.13	0.13
MDPC1 vs MDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.10	0.10	0.10
MDPC2 vs MDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09
MDPC2 vs MDPC2	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.07	0.08	0.09	0.09	0.10	0.10	0.11	0.11	0.11	0.12	0.12	0.12	0.12
T=800																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.06	0.07	0.08	0.08	0.09	0.09	0.10	0.10	0.11	0.11	0.11	0.11	0.12
MDPC1 vs MDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08
MDPC2 vs MDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08
MDPC2 vs MDPC2	0.00	0.00	0.02	0.03	0.04	0.05	0.05	0.06	0.07	0.08	0.08	0.09	0.09	0.10	0.10	0.10	0.11	0.11	0.11	0.11	0.12

Table 3.5: Standard deviations of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.1.

In the following, considering one replica of the above simulations with $T = 1000$,

where MDPCA with $W = 101$ is used. The optimal number of lags is suggested to be $l = 1$, as shown in Figure 3.19, as one static and two dynamic relations are found. A plot of the eigenvalues is provided in Figure 3.20, where the RCC criterion suggests to retain the first two MDPCs. These two MDPCs explain 97.39% of the variation in the data and produce a small reconstruction error of 5.17. Furthermore, the sample moving cross-correlation plots of the two MDPCs in Figure 3.21 indicate uncorrelated MDPCs.

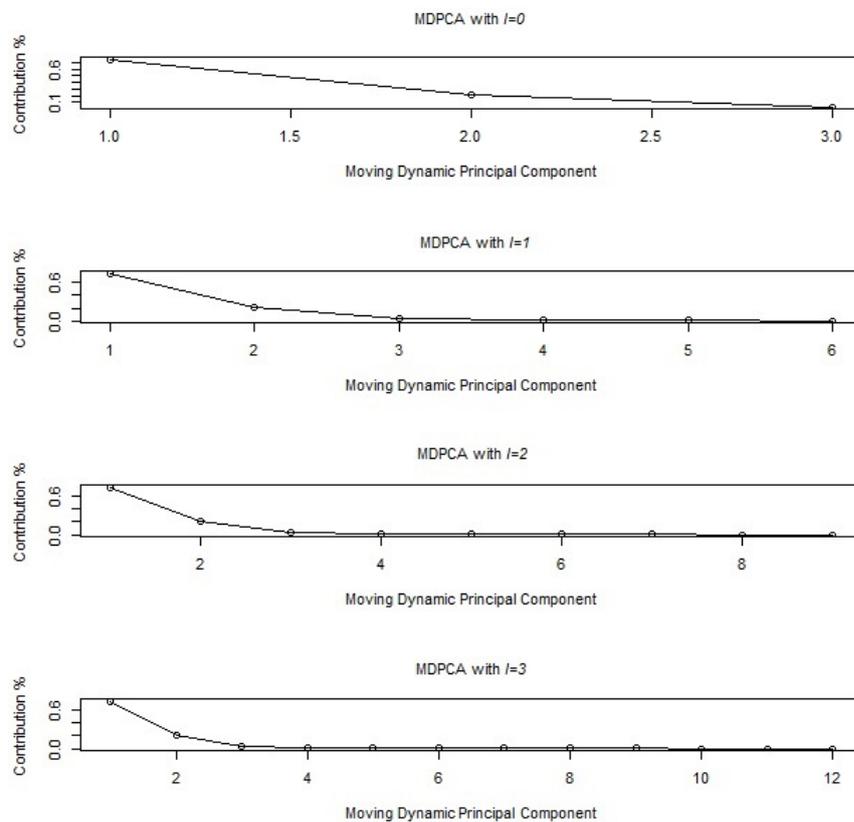


Figure 3.19: Eigenvalues' plots after applying MDPCA with $W = 101$ and different sizes of l to the simulated series with three variables in Example 3.3.1.

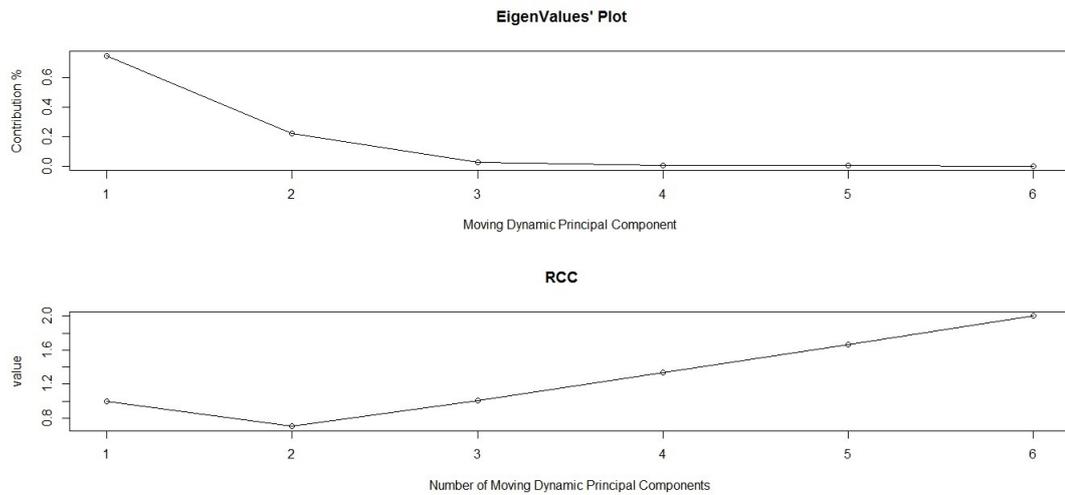


Figure 3.20: Top: An eigenvalues' plot of the simulated series with three variables in Example 3.3.1 after applying MDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot.

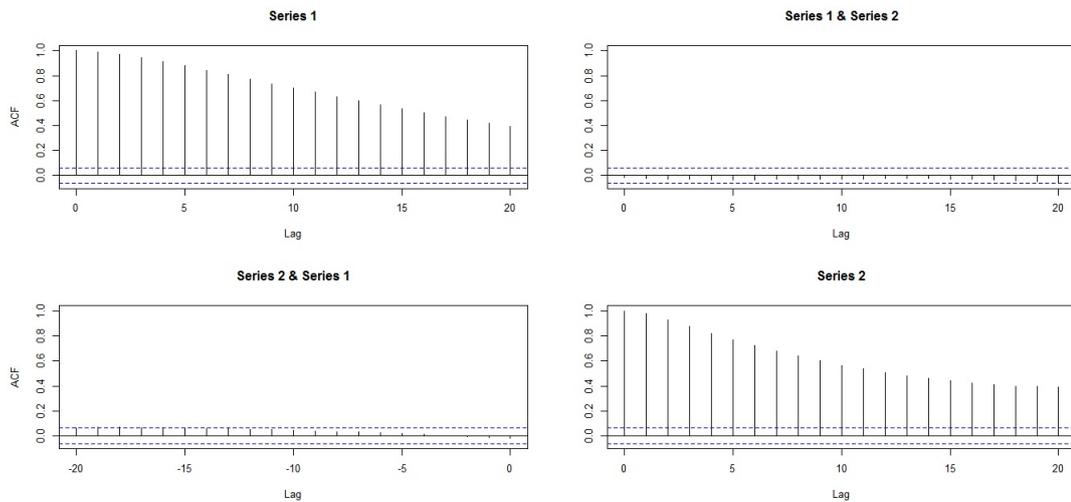


Figure 3.21: Sample moving cross-correlation plots of the MDPCs of the simulated series with three variables in Example 3.3.1 after applying MDPCA with $W = 101$ and $l = 1$.

Example 3.3.2 The following simulation study includes a non-stationary time series \mathbf{z}_t with a dimension $m = 6$. This study consists of three parts. The first part compares the results of MDPCA when using different combinations of W and l . The second part studies the effect of the size of T on MDPCA's results. The third part

focuses on optimizing the results of MDPCA. Six different non-stationary models are used to generate the variables of the series \mathbf{z}_t as described below.

Let a_t, b_t, c_t, d_t, e_t and f_t be six independent standard normal white noises, which are the innovation terms of the following six models, respectively, then:

$$\begin{cases} z_{1,t} = u_t \\ z_{2,t} = v_t \\ z_{3,t} = w_t \\ z_{4,t} = x_t \\ z_{5,t} = q_t \\ z_{6,t} = g_t \end{cases} \quad (3.22)$$

where $(u_t, v_t, w_t, x_t, q_t, g_t)'$ satisfies

$$\begin{cases} \nabla u_t = 0.75 \nabla u_{t-1} + a_t + 0.9a_{t-1} \\ \nabla v_t = 0.6 \nabla v_{t-1} + b_t - 1.4b_{t-1} \\ \nabla w_t = -0.7 \nabla w_{t-1} + c_t + 2.3c_{t-1} \\ \nabla x_t = -0.5 \nabla x_{t-1} + d_t + 0.55d_{t-1} \\ \nabla q_t = 0.6 \nabla q_{t-1} + e_t + 1.65e_{t-1} \\ \nabla g_t = -0.75 \nabla g_{t-1} + f_t - 0.8f_{t-1} \end{cases} \quad (3.23)$$

Figure 3.22 is a time series plot of one randomly selected simulation. The data showed an apparent non-stationary behaviour.

The first part of this simulation study focuses on the behaviour of MDPCA with different combinations of W and l . Based on 500 replicas, Table 3.6 shows the results of applying MDPCA with different sizes of W and l to the simulated series \mathbf{z}_t with $T = 1500$, where three MDPCs are retained. Notice that the mean of explained variance by the three MDPCs when using $l = 1$ was slightly higher compared with those when using $l = 5$. For instance, on average, the three MDPCs explained 97.50% of the variation of the data when using MDPCA with $W = 101$ and $l = 1$,

compared with 95.49% when using $W = 101$ and $l = 5$. Both cases produced a relative standard deviation of 0.01, which indicated steady contribution percentages in all replicas. The mean of MSE of reconstructed data ranged between 175.90 to 228.48 for the considered combinations of W and l , where it reached its lowest when using $W = 301$ and $l = 1$, and it reached its highest when using $W = 101$ and $l = 5$.

On the other hand, the means of the absolute value of the moving cross-correlation with $W = 101$ are plotted for each case in figures 3.23 - 3.25, and the corresponding standard deviations are shown in Tables 3.7 and 3.8. The three MDPCs were uncorrelated for the cases where $W = 101$ was used. For the case where $W = 201$ was used, the correlations got slightly larger. We noticed significant correlations when increasing the window size to $W = 301$.

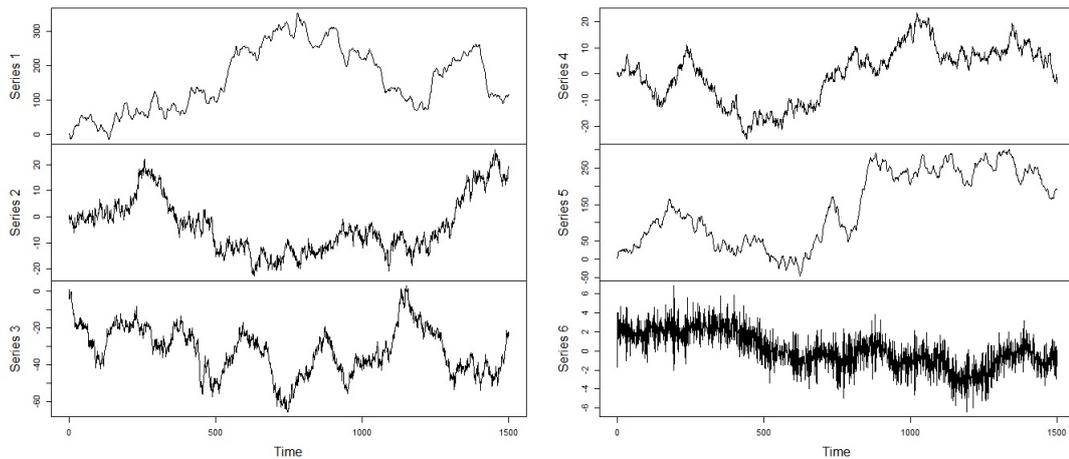


Figure 3.22: A time series plot of the simulated series with six variables in Example 3.3.2.

Window Size (W) Results	W=101 ($l=1$)	W=201 ($l=1$)	W=301 ($l=1$)
Contribution % Mean (Standard Deviation)	97.50 % (0.00)	97.94 % (0.01)	98.12 % (0.01)
MSE Mean (Standard Deviation)	217.53 (181.78)	194.01 (154.48)	175.90 (147.55)
Window Size (W) Results	W=101 ($l=5$)	W=201 ($l=5$)	W=301 ($l=5$)
Contribution % Mean (Standard Deviation)	95.49 % (0.01)	96.91 % (0.01)	97.43 % (0.01)
MSE Mean (Standard Deviation)	228.48 (179.07)	202.76 (165.39)	178.97 (150.13)

Table 3.6: A comparison between the results of MDPCA with different combinations of W and l in Example 3.3.2.

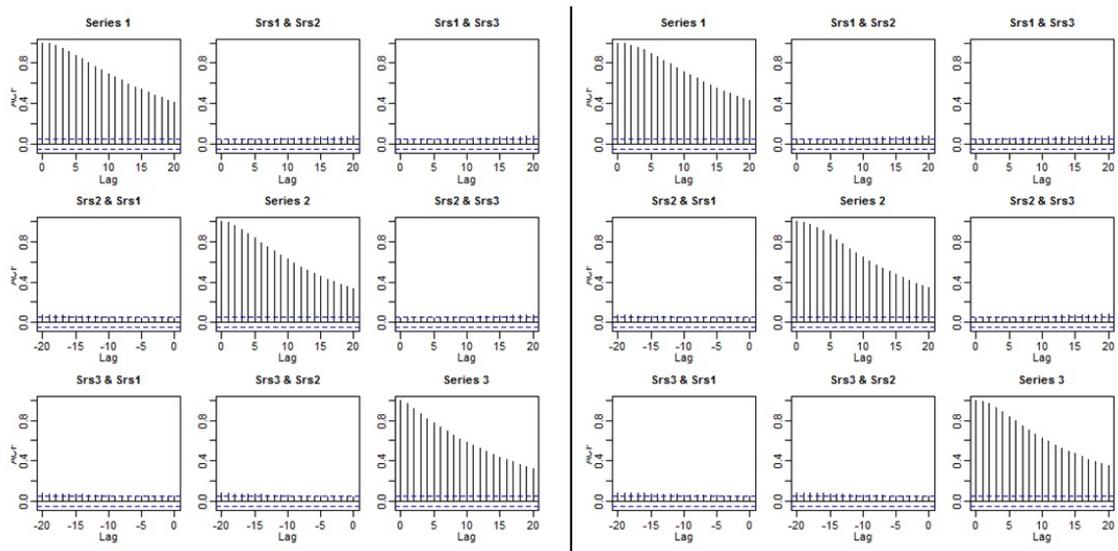


Figure 3.23: Plots of the mean of the absolute value of the moving cross-correlations between three MDPCs in Example 3.3.2 [Left: MDPCA ($W = 101, l = 1$), Right: MDPCA($W = 101, l = 5$)].

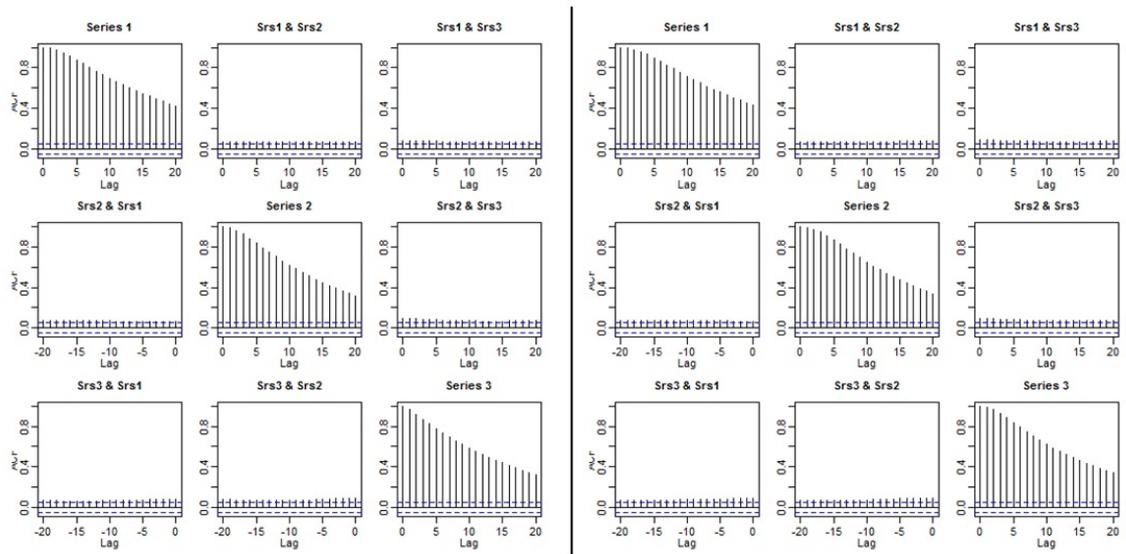


Figure 3.24: Plots of the mean of the absolute value of the moving cross-correlations between three MDPCs in Example 3.3.2 [Left: MDPCA ($W = 201, l = 1$), Right: MDPCA($W = 201, l = 5$)].

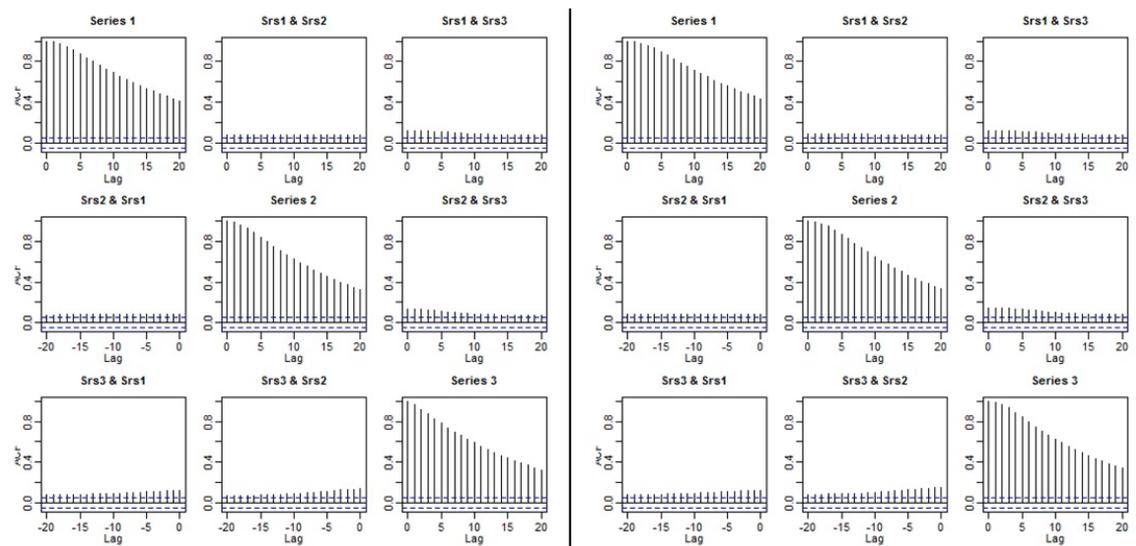


Figure 3.25: Plots of the mean of the absolute value of the moving cross-correlations between three MDPCs in Example 3.3.2 [Left: MDPCA ($W = 301, l = 1$), Right: MDPCA($W = 301, l = 5$)].

		MDPCA(W=101, l=1)																			
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08
MDPC1 vs MDPC2	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06
MDPC1 vs MDPC3	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.06
MDPC2 vs MDPC1	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.06
MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08
MDPC2 vs MDPC3	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05
MDPC3 vs MDPC1	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05
MDPC3 vs MDPC2	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.06
MDPC3 vs MDPC3	0.00	0.00	0.01	0.02	0.03	0.03	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08
		MDPCA(W=201, l=1)																			
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08
MDPC1 vs MDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
MDPC1 vs MDPC3	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08
MDPC2 vs MDPC3	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC3 vs MDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC3 vs MDPC2	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC3 vs MDPC3	0.00	0.00	0.01	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08
		MDPCA(W=301, l=1)																			
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08
MDPC1 vs MDPC2	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC1 vs MDPC3	0.09	0.09	0.09	0.08	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.08
MDPC2 vs MDPC3	0.10	0.10	0.09	0.09	0.09	0.08	0.08	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC3 vs MDPC1	0.09	0.09	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC3 vs MDPC2	0.10	0.10	0.09	0.09	0.09	0.08	0.08	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC3 vs MDPC3	0.00	0.00	0.01	0.02	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07

Table 3.7: Standard deviations of the absolute value of the moving cross-correlations between three MDPCs with different combinations of W and l in Example 3.3.2 - Part 1.

		MDPCA(W=101, l=5)																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07
MDPC1 vs MDPC2	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05
MDPC1 vs MDPC3	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC1	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC2	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08
MDPC2 vs MDPC3	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06
MDPC3 vs MDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05
MDPC3 vs MDPC2	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05
MDPC3 vs MDPC3	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08
		MDPCA(W=301, l=5)																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07
MDPC1 vs MDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06
MDPC1 vs MDPC3	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC2	0.00	0.00	0.00	0.01	0.02	0.02	0.03	0.03	0.03	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08
MDPC2 vs MDPC3	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
MDPC3 vs MDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC3 vs MDPC2	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
MDPC3 vs MDPC3	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08
		MDPCA(W=301, l=5)																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07
MDPC1 vs MDPC2	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC1 vs MDPC3	0.09	0.09	0.09	0.09	0.09	0.08	0.08	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC2	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08
MDPC2 vs MDPC3	0.11	0.11	0.10	0.10	0.10	0.10	0.09	0.09	0.09	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06
MDPC3 vs MDPC1	0.09	0.09	0.09	0.08	0.08	0.08	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC3 vs MDPC2	0.11	0.11	0.11	0.11	0.11	0.10	0.10	0.09	0.09	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06
MDPC3 vs MDPC3	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08

Table 3.8: Standard deviations of the absolute value of the moving cross-correlations between three MDPCs with different combinations of W and l in Example 3.3.2 - Part 2.

Therefore, the dimension of \mathbf{z}_t can be reduced by using MDPCA with $W = 101$ and $l = 1$. Alternatively, with slightly less accuracy, MDPCA with $W = 101$ and

$l = 5$ can be used to reduce the dimension of \mathbf{z}_t , where more dynamic information is included in the analysis.

Method	MDPCA (W=101, l=5)			
Sample size	T=600	T=800	T=1000	T=1200
Contribution %	95.69 %	95.60 %	95.57 %	95.53 %
Mean (Standard Deviation)	(0.01)	(0.01)	(0.01)	(0.01)
MSE	81.15	103.66	139.33	186.99
Mean (Standard Deviation)	(59.93)	(79.53)	(108.24)	(138.06)

Table 3.9: A comparison between the results of MDPCA with different sizes of T in Example 3.3.2.

The second part of this study focuses on the effect of the sample size T on the results of MDPCA. Based on 500 replica, Table 3.9 compares the results of applying MDPCA with $W = 101$ and $l = 5$ to \mathbf{z}_t with different sample sizes, where three MDPCs are retained in all cases. Notice that the means of explained variance had similar percentages when applying MDPCA to the data with different sizes of T . For example, the highest percentage of 95.69% was obtained for the case where $T = 600$, compared with the lowest percentage of 95.53% for the case where $T = 1200$. Similar percentages were obtained in all 500 replicas as they have standard deviations of 0.01 in all cases. For the mean of MSE of reconstructed data, we noticed an increase of the MSE when increasing T . The highest MSE of 186.99 was obtained when $T = 1200$, compared with the lowest MSE of 81.15 when $T = 600$.

Furthermore, the means of the absolute value of the moving cross-correlation with $W = 101$ are plotted for each sample size in figures 3.26 and 3.27. There were no significant dynamic relationships between the three MDPCs on the first ± 15 lags, in all cases. For lags that are larger than 15 and smaller than -15, some hardly significant correlations existed between the MDPCs for the cases where $T = 600$ and 800. These correlations became less significant as we increased T . The corresponding standard deviations are reported in Table 3.10, where similar values were obtained in all cases. However, a slight improvement on the standard deviations of the absolute value of the moving cross-correlation can be obtained when increasing

T . For instance, the standard deviations ranged from 0 to 0.13 when $T = 600$, and from 0 to 0.09 when $T = 1200$.

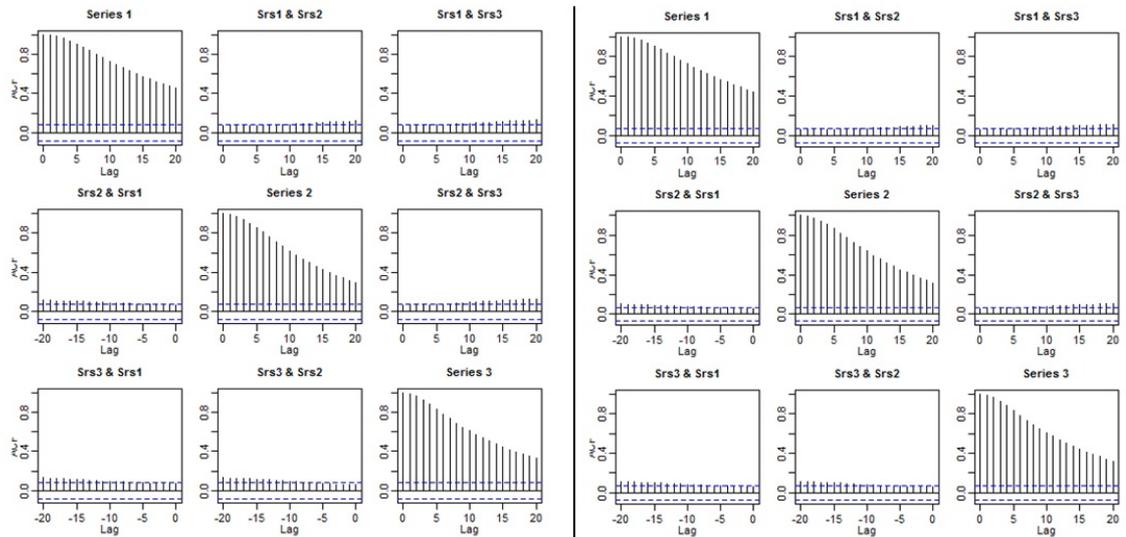


Figure 3.26: Plots of the mean of the absolute value of the moving cross-correlations between three MDPCs with different sizes of T in Example 3.3.2 [Left: $T = 600$, Right: $T = 800$].

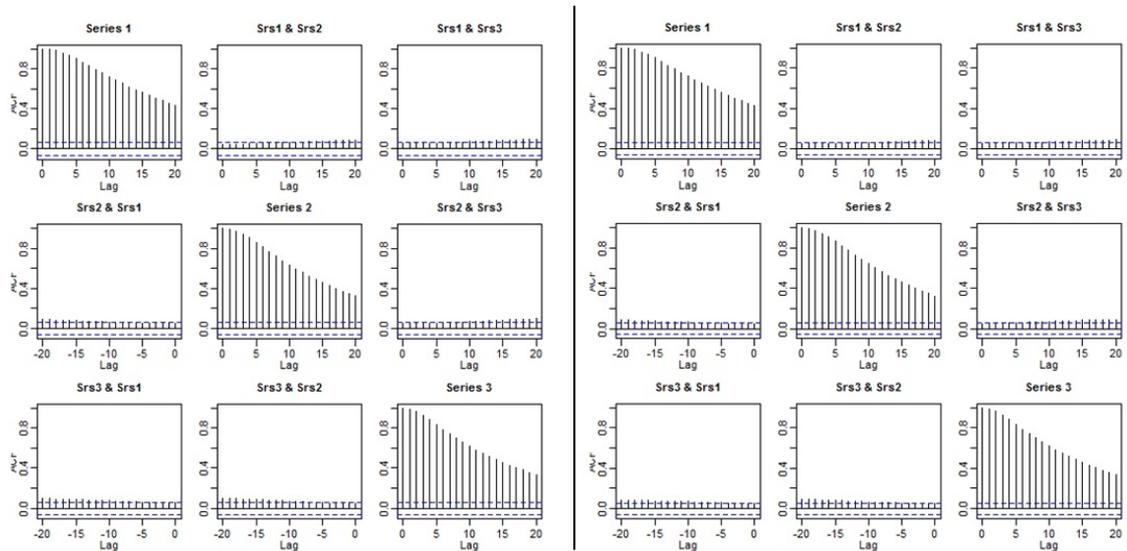


Figure 3.27: Plots of the mean of the absolute value of the moving cross-correlations between three MDPCs with different sizes of T in Example 3.3.2 [Left: $T = 1000$, Right: $T = 1200$].

		T=600																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	MDFCI vs MDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.04	0.04	0.05	0.06	0.07	0.08	0.08	0.09	0.10	0.10	0.11	0.11	0.12	0.13
	MDFCI vs MDPC2	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.08	0.08	0.08
	MDFCI vs MDPC3	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
	MDPC2 vs MDPC1	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.08	0.09	0.09
	MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.12	0.12	0.13	0.13	0.13	0.13	0.13
	MDPC2 vs MDPC3	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.08	0.08	0.09	0.09
	MDPC3 vs MDPC1	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.09	0.09
	MDPC3 vs MDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.08	0.08	0.09	0.09
	MDPC3 vs MDPC3	0.00	0.00	0.01	0.02	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.10	0.11	0.12	0.12	0.13	0.13	0.13	0.13	0.13	0.13
		T=800																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	MDFCI vs MDPC1	0.00	0.00	0.00	0.01	0.02	0.02	0.03	0.04	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.09	0.09	0.09	0.10	0.10
	MDFCI vs MDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.08	0.08
	MDFCI vs MDPC3	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08	0.08
	MDPC2 vs MDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
	MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.08	0.08
	MDPC2 vs MDPC3	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.08	0.08
	MDPC3 vs MDPC1	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.08	0.08
	MDPC3 vs MDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.08	0.08
	MDPC3 vs MDPC3	0.00	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.06	0.07	0.08	0.08	0.09	0.09	0.10	0.10	0.10	0.11	0.11	0.11	0.11
		T=1000																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	MDFCI vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.09
	MDFCI vs MDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
	MDFCI vs MDPC3	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
	MDPC2 vs MDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
	MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.09	0.09
	MDPC2 vs MDPC3	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
	MDPC3 vs MDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
	MDPC3 vs MDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
	MDPC3 vs MDPC3	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.10	0.10
		T=1200																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	MDFCI vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.08
	MDFCI vs MDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
	MDFCI vs MDPC3	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
	MDPC2 vs MDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
	MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.09	0.09
	MDPC2 vs MDPC3	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
	MDPC3 vs MDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
	MDPC3 vs MDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
	MDPC3 vs MDPC3	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.10	0.10

Table 3.10: Standard deviations of the absolute value of the moving cross-correlations between three MDPCs with different sizes of T in Example 3.3.2.

Consider one randomly selected replica with $T = 1500$, where MDPCA with

$W = 101$ is used. The optimal number of lags is suggested to be $l = 1$, as shown in Figure 3.28, where three static and three dynamic relations were found. A plot of the eigenvalues and the RCC's results are shown in Figure 3.29. The RCC criterion suggested three MDPCs to be optimal. The retained MDPCs were able to explain 95.42% of the variation in the data and produced an MSE of 140.35 when used to reconstruct the data. The sample moving cross-correlation plots of the retained components suggest uncorrelated components; See Figure 3.30.

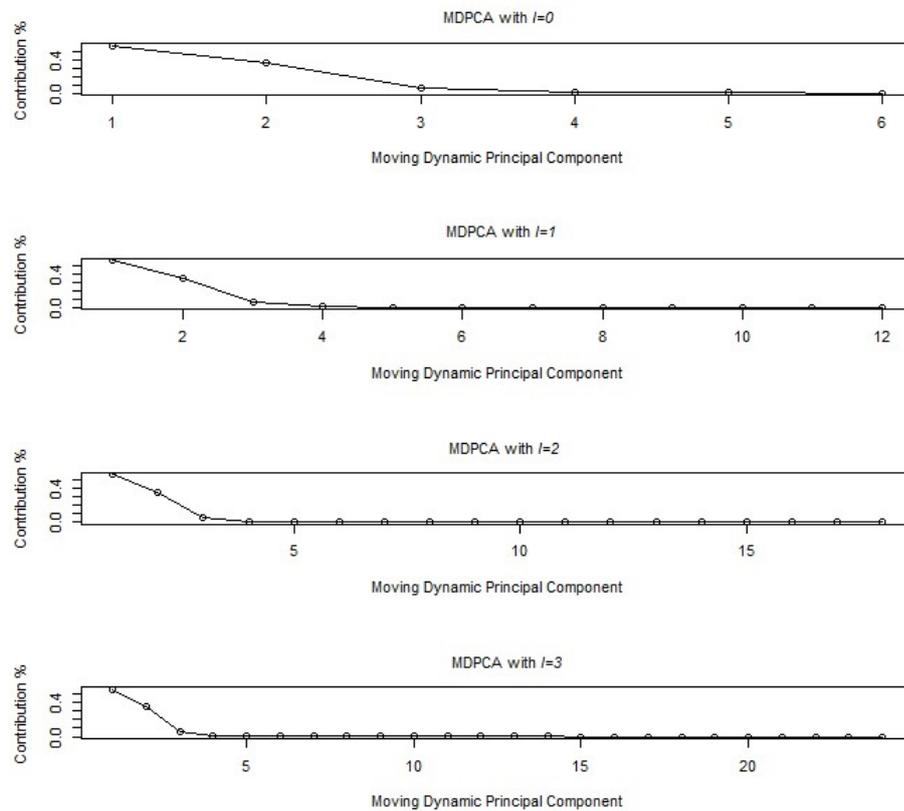


Figure 3.28: Eigenvalues' plots after applying MDPCA with $W = 101$ and different sizes of l to the simulated series with six variables in Example 3.3.2.

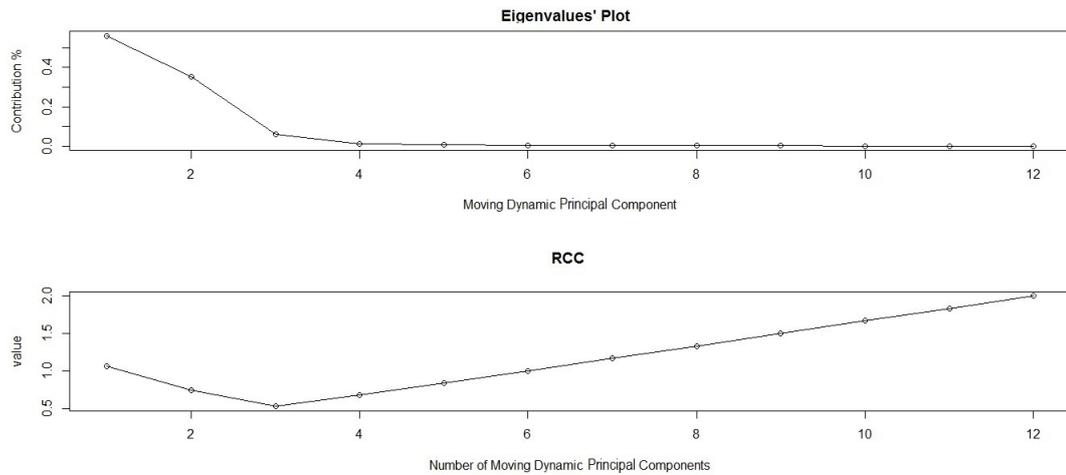


Figure 3.29: Top: An eigenvalues' plot of the simulated series with six variables in Example 3.3.2 after applying MDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot.

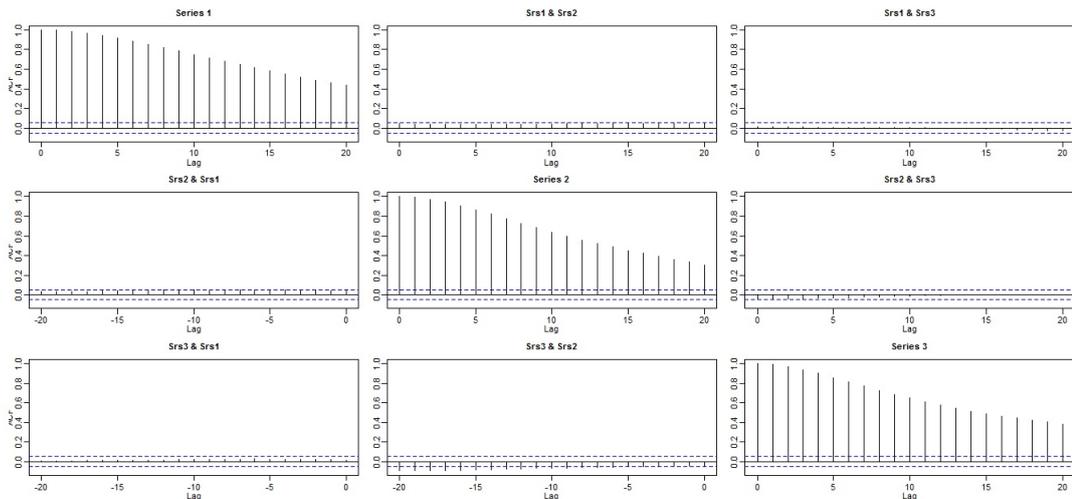


Figure 3.30: Sample moving cross-correlation plots of the MDPCs of the simulated series in Example 3.3.2 after MDPCA with $W = 101$ and $l = 1$ is applied.

Example 3.3.3 In this example, we apply the MDPCA on a non-stationary series \mathbf{z}_t with ten variables. This example consists of two parts. The first part studies the results of MDPCA when using different combinations of W and l . The second part compares the effect of the size of T on MDPCA's results. The series \mathbf{z}_t is generated

using five different models, such that each model produces two correlated variables as follows.

Let a_t, b_t, c_t, d_t and e_t be independent standard normal white noises, which are the innovation terms of the following five models, respectively, then:

$$\left\{ \begin{array}{ll} z_{j,t} = u_{t+j-1}, & j = 1 \text{ and } 2 \\ z_{j,t} = v_{t+j-3}, & j = 3 \text{ and } 4 \\ z_{j,t} = w_{t+j-5}, & j = 5 \text{ and } 6 \\ z_{j,t} = x_{t+j-7}, & j = 7 \text{ and } 8 \\ z_{j,t} = q_{t+j-9}, & j = 9 \text{ and } 10 \end{array} \right. \quad (3.24)$$

where $(u_t, v_t, w_t, x_t, q_t)'$ satisfies

$$\left\{ \begin{array}{l} \nabla u_t = 0.75 \nabla u_{t-1} + a_t + 0.9a_{t-1} \\ \nabla v_t = 0.6 \nabla v_{t-1} + b_t - 1.4b_{t-1} \\ \nabla w_t = -0.7 \nabla w_{t-1} + c_t - 2.3c_{t-1} \\ \nabla x_t = -0.5 \nabla x_{t-1} + d_t + 0.55d_{t-1} \\ \nabla q_t = 0.6 \nabla q_{t-1} + e_t + 1.65e_{t-1} \end{array} \right. \quad (3.25)$$

A time series plot of the simulated data is shown in Figure 3.31 where all variables exhibit non-stationary behaviours.

First, we would like to see the results of MDPCA with different options of W and l . Based on 500 replicas, Table 3.11 is a comparison between the results of MDPCA with different sizes of W and l when applied to the simulated series \mathbf{z}_t with 1500 samples, where two MDPCs are considered. With respect to the mean percentages of explained variance by the two MDPCs, the percentages obtained by using $l = 1$ were higher than those obtained by using $l = 5$. However, the percentages differed by a small amount. For example, on average, two MDPCs explained 96.48% of the variation of the data when using MDPCA with $W = 101$ and $l = 1$, compared with 94.59% when using $W = 101$ and $l = 5$. The standard deviations of explained

variance in all cases were 0.01, which indicate steady percentages in all replicas. For the mean of MSE of reconstructed data, it ranged between 309.09 to 403.74. The latter had its lowest when $W = 301$ and $l = 1$ are used, and had its highest when $W = 101$ and $l = 5$ are used. This behaviour is similar to what we observed in the previous examples when \mathbf{z}_t had fewer dimensions.

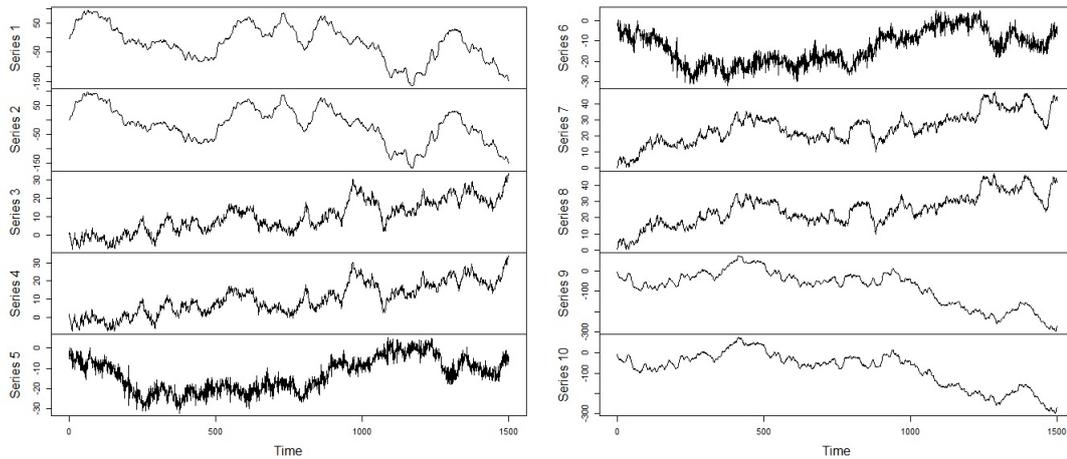


Figure 3.31: A time series plot of the simulated series with ten variables in Example 3.3.3.

Furthermore, the dynamic dependence between the two MDPCs was revealed by plotting the means of the absolute value of the moving cross-correlation with $W = 101$; See figures 3.32 - 3.34 and the corresponding standard deviations in Table 3.12. There were no significant dynamic relationships between the two MDPCs for both cases when using $W = 101$. However, the correlations got slightly larger when using $W = 201$, and crossed the significant line when using $W = 301$.

Window Size (W) Results	W=101 ($l=1$)	W=201 ($l=1$)	W=301 ($l=1$)
Contribution % Mean (Standard Deviation)	96.48 % (0.01)	97.01 % (0.01)	97.34 % (0.01)
MSE Mean (Standard Deviation)	359.27 (234.25)	350.18 (246.62)	309.09 (226.98)
Window Size (W) Results	W=101 ($l=5$)	W=201 ($l=5$)	W=301 ($l=5$)
Contribution % Mean (Standard Deviation)	94.59 % (0.01)	96.11 % (0.01)	96.71 % (0.01)
MSE Mean (Standard Deviation)	403.74 (288.54)	335.89 (221.02)	333.04 (213.36)

Table 3.11: A comparison between the results of MDPCA with different combinations of W and l in Example 3.3.3.

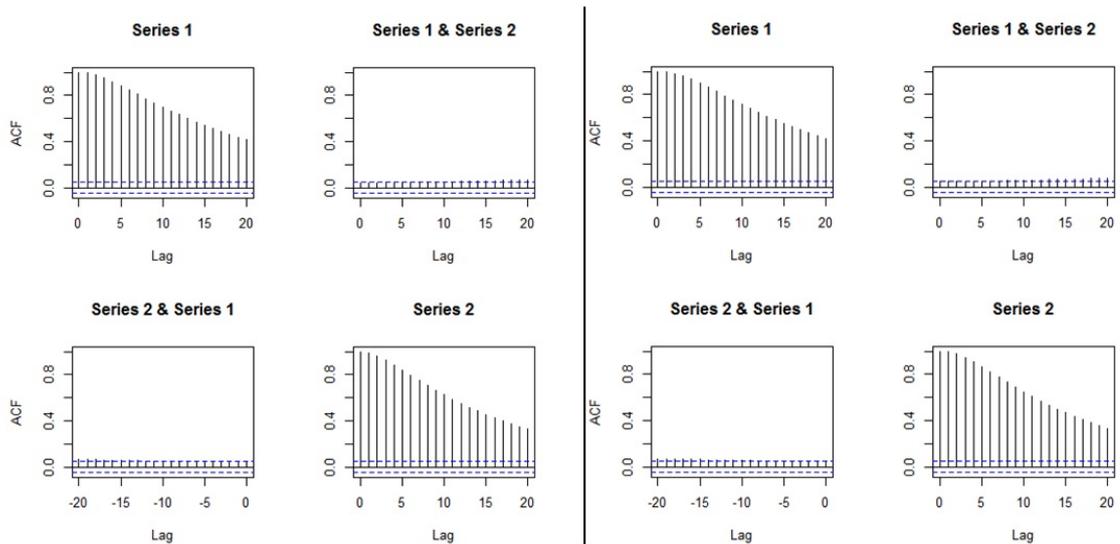


Figure 3.32: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.3 [Left: MDPCA ($W = 101, l = 1$), Right: MDPCA($W = 101, l = 5$)].

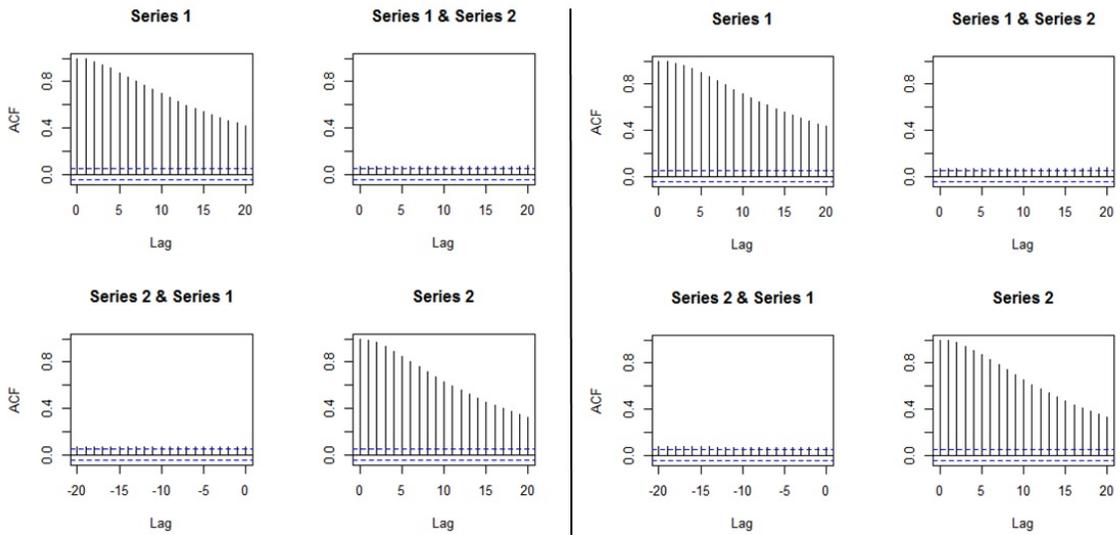


Figure 3.33: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.3 [Left: MDPCA ($W = 201, l = 1$), Right: MDPCA($W = 201, l = 5$)].

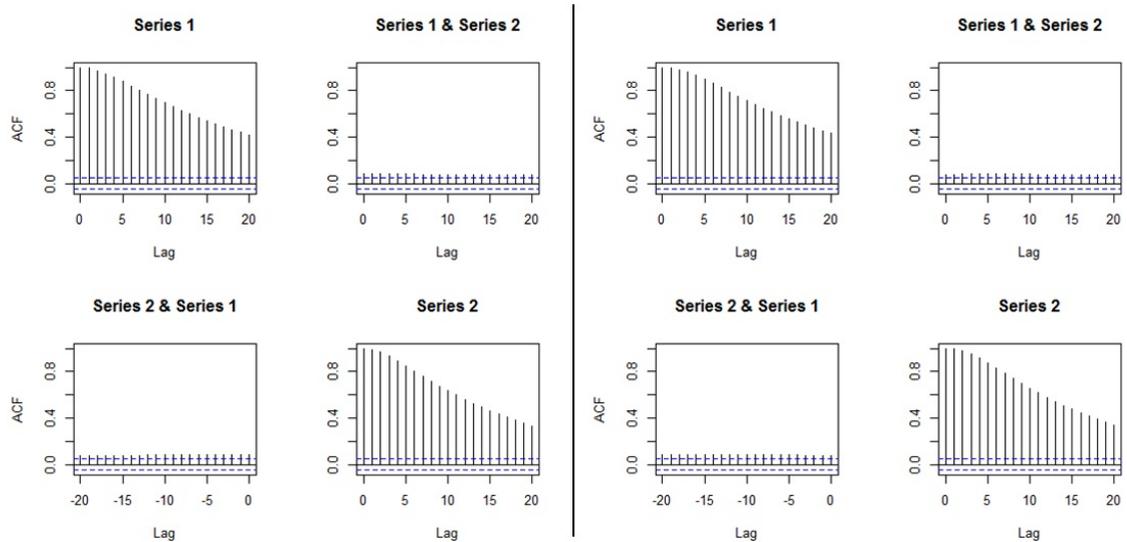


Figure 3.34: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.3 [Left: MDPCA ($W = 301, l = 1$), Right: MDPCA($W = 301, l = 5$)].

MDPCA(W=101, l=1)																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.08
MDPC1 vs MDPC2	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC1	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06
MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.08
MDPCA(W=201, l=1)																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07
MDPC1 vs MDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC2	0.00	0.00	0.01	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08
MDPCA(W=301, l=1)																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.03	0.04	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07
MDPC1 vs MDPC2	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC2	0.00	0.00	0.01	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08
MDPCA(W=101, l=5)																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08
MDPC1 vs MDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06
MDPC2 vs MDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06
MDPC2 vs MDPC2	0.00	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08
MDPCA(W=201, l=5)																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08
MDPC1 vs MDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06
MDPC2 vs MDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06
MDPC2 vs MDPC2	0.00	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08
MDPCA(W=301, l=5)																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08
MDPC1 vs MDPC2	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC2	0.00	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08

Table 3.12: Standard deviations of the absolute value of the moving cross-correlations between two MDPCs with different options of W and l in Example 3.3.3.

Therefore, based on the above discussion, the dimension of the simulated non-stationary series with ten variables in this example can be reduced by MDPCA with $W = 101$ and $l = 1$ or using MDPCA with $W = 101$ and $l = 5$.

Second, we would like to see the results of MDPCA when the series \mathbf{z}_t has different sample sizes (i.e. $T = \{200, 400, 600, 800\}$). In this part, we consider MDPCA with $W = 101$ and $l = 5$. Based on 500 replicas, the results of applying MDPCA on \mathbf{z}_t with different sample sizes are shown in Table 3.13, where two MDPCs are retained. For the mean percentage of explained variance, the highest percentage of 96.68% was obtained when $T = 200$, then decreased to 94.54% when $T = 400$, then followed by a continues increasing to 94.61% when $T = 600$ and 94.67% when $T = 800$. The standard deviations of the percentage of explained variance decreased from 0.03 when $T = 200$ to 0.01 when $T = 600$, then stayed at this value for larger sample sizes. For the mean of MSE of reconstructed data, its values had a positive relationship with T . For example, the mean MSE was 52.68 when $T = 200$, then increased to 191.59 when $T = 800$. Notice that the MSE values are not steady across the 500 replicas, as their standard deviations were large (e.g. 66.27 standard deviation when $T = 400$).

On the other hand, the means of the absolute value of the moving cross-correlation with $W = 101$ indicated a significant dynamic relationship between the two MDPCs when $T = 200$; See Figure 3.35. The correlations then decreased between the two MDPCs when $T = 400$ with some minor significant cross-correlations. For the cases where $T = 600$ and 800, the plots indicated uncorrelated MDPCs; See Figure 3.36. Also, Table 3.14 shows the standard deviations of the absolute value of the moving cross-correlation. We noticed an improvement on the standard deviations when increasing the sample size of the data. For example, the standard deviations ranged from 0 to 0.24 when $T = 200$, compared with 0 to 0.12 when $T = 800$. Notice that by applying MDPCA to the simulated data with $T = 600$, we were able to obtain similar results to those with $T = 1500$. Therefore, even though we increased the dimension of \mathbf{z}_t to $m = 10$, MDPCA still performs well on data with moderate

sample sizes.

Method	MDPCA ($W=101, l=5$)			
Sample size	T=200	T=400	T=600	T=800
Contribution %	96.68 %	94.54 %	94.61 %	94.67 %
Mean (Standard Deviation)	(0.03)	(0.02)	(0.01)	(0.01)
MSE	52.68	95.24	148.57	191.59
Mean (Standard Deviation)	(37.04)	(66.27)	(87.82)	(125.03)

Table 3.13: A comparison between the results of MDPCA with different sizes of T in Example 3.3.3.

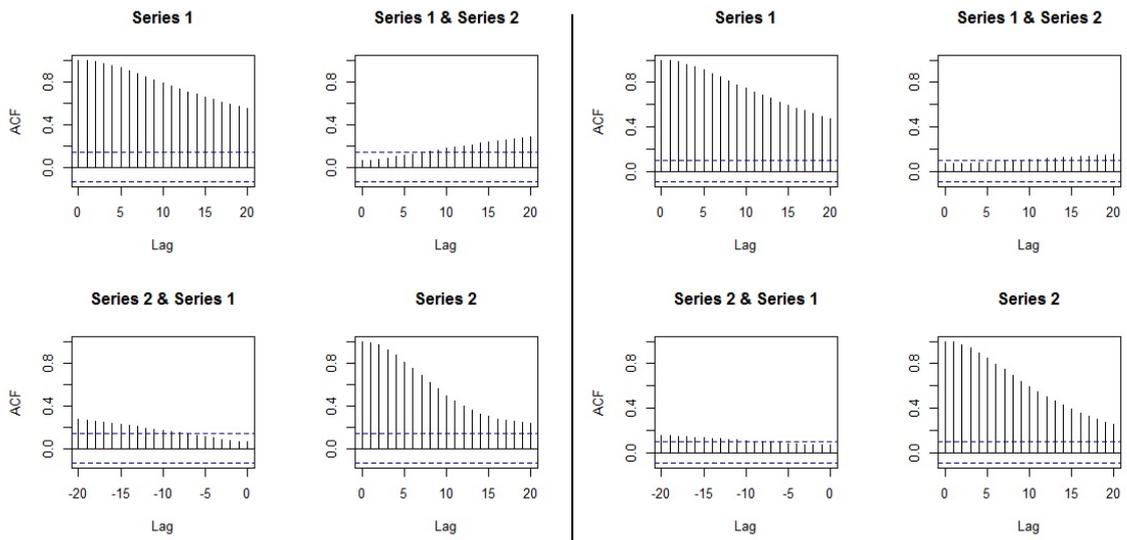


Figure 3.35: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.3 [Left: $T = 200$, Right: $T = 400$].

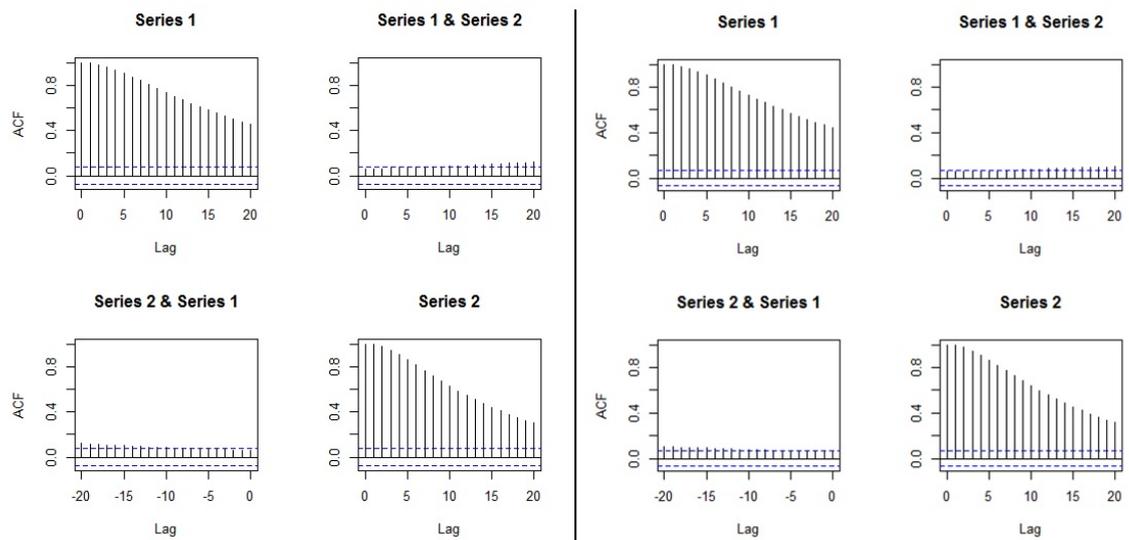


Figure 3.36: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.3 [Left: $T = 600$, Right: $T = 800$].

T=200																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.02	0.03	0.04	0.06	0.07	0.09	0.11	0.12	0.14	0.15	0.16	0.18	0.19	0.20	0.21	0.22	0.23	0.24
MDPC1 vs MDPC2	0.06	0.06	0.06	0.07	0.07	0.08	0.09	0.10	0.11	0.11	0.12	0.13	0.13	0.14	0.15	0.15	0.16	0.16	0.17	0.17	0.18
MDPC2 vs MDPC1	0.06	0.06	0.06	0.07	0.07	0.08	0.09	0.10	0.10	0.11	0.12	0.13	0.14	0.15	0.15	0.16	0.17	0.17	0.18	0.18	0.19
MDPC2 vs MDPC2	0.00	0.00	0.02	0.03	0.05	0.08	0.10	0.13	0.15	0.16	0.18	0.19	0.20	0.19	0.19	0.18	0.17	0.17	0.17	0.17	0.16
T=400																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.12	0.12	0.13	0.14	0.14	0.15	0.15
MDPC1 vs MDPC2	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.09	0.10	0.10	0.11	0.11	0.11	0.12
MDPC2 vs MDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.08	0.09	0.10	0.10	0.10	0.11	0.11	0.11
MDPC2 vs MDPC2	0.00	0.00	0.01	0.02	0.03	0.05	0.06	0.08	0.09	0.10	0.11	0.12	0.13	0.14	0.15	0.15	0.16	0.16	0.16	0.16	0.16
T=600																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.02	0.03	0.03	0.04	0.05	0.06	0.07	0.07	0.07	0.08	0.08	0.09	0.10	0.10	0.11	0.11	0.12
MDPC1 vs MDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09
MDPC2 vs MDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09
MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.09	0.10	0.11	0.11	0.11	0.12	0.12	0.12	0.13
T=800																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.04	0.05	0.06	0.06	0.06	0.07	0.08	0.09	0.09	0.10	0.10	0.10	0.11
MDPC1 vs MDPC2	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08
MDPC2 vs MDPC1	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08
MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.08	0.09	0.10	0.10	0.10	0.11	0.11	0.11	0.11	0.12

Table 3.14: Standard deviations of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.3.

Example 3.3.4 In the following simulation study, MDPCA is applied to a non-stationary series \mathbf{z}_t that consists of 15 variables. This study consists of two parts. The first part compares the MDPCA's results when using different combinations of W and l . The second part compares the effect of the size of T on MDPCA's results. The data is generated using five different models, such that each model produces three correlated variables.

Let a_t, b_t, c_t, d_t and e_t be independent standard normal white noises, which are the innovation terms of the following five models, respectively, then:

$$\left\{ \begin{array}{ll} z_{j,t} = u_{t+j-1}, & j = 1, 2 \text{ and } 3 \\ z_{j,t} = v_{t+j-4}, & j = 4, 5 \text{ and } 6 \\ z_{j,t} = w_{t+j-7}, & j = 7, 8 \text{ and } 9 \\ z_{j,t} = x_{t+j-10}, & j = 10, 11 \text{ and } 12 \\ z_{j,t} = q_{t+j-13}, & j = 13, 14 \text{ and } 15 \end{array} \right. \quad (3.26)$$

where $(u_t, v_t, w_t, x_t, q_t)'$ satisfies

$$\left\{ \begin{array}{l} \nabla u_t = 0.75 \nabla u_{t-1} + a_t + 0.9a_{t-1} \\ \nabla v_t = 0.6 \nabla v_{t-1} + b_t - 1.4b_{t-1} \\ \nabla w_t = -0.7 \nabla w_{t-1} + c_t - 2.3c_{t-1} \\ \nabla x_t = -0.5 \nabla x_{t-1} + d_t + 0.55d_{t-1} \\ \nabla q_t = 0.6 \nabla q_{t-1} + e_t + 1.65e_{t-1} \end{array} \right. \quad (3.27)$$

Based on 500 replicas, Table 3.15 shows the results of MDPCA with different combinations of W and l applied to \mathbf{z}_t with 2000 samples, where two MDPCs are considered. In general, the results in this example followed a similar pattern as in the previous examples where MDPCA was applied to series with lower dimensions. It can be seen that the mean percentages of the explained variance by the two MDPCs increased slightly as we increased W . For example, the mean percentage was 96.13% when using MDPCA with $W = 101$ and $l = 1$, then increased to 97.22% when using MDPCA with $W = 301$ and $l = 1$. Also, the percentages were slightly

lower when using more lagged data (i.e. when increasing l). For example, the mean percentage was 96.85% when using MDPCA with $W = 201$ and $l = 1$, then decreased to 95.90% when using MDPCA with $W = 201$ and $l = 5$. The standard deviation of the percentage of explained variance had a small value of 0.01 in all cases. This indicates steady results for the explained variance in all cases. Additionally, the MSE of reconstructed data had a large standard deviations, which means that its values can be small or large depending on the data. For example, the mean MSE was 348.56 when using MDPCA with $W = 101$ and $l = 1$, and had a standard deviation of 230.96.

Window Size (W)	W=101	W=201	W=301
Results	($l=1$)	($l=1$)	($l=1$)
Contribution % Mean (Standard Deviation)	96.13 % (0.01)	96.85 % (0.01)	97.22 % (0.01)
MSE Mean (Standard Deviation)	348.56 (230.96)	331.78 (223.55)	314.06 (207.18)
Window Size (W)	W=101	W=201	W=301
Results	($l=5$)	($l=5$)	($l=5$)
Contribution % Mean (Standard Deviation)	94.43 % (0.01)	95.90 % (0.01)	96.54 % (0.01)
MSE Mean (Standard Deviation)	364.54 (254.41)	345.85 (227.12)	327.43 (221.58)

Table 3.15: A comparison between the results of MDPCA with different combinations of W and l in Example 3.3.4.

Additionally, the plot of the mean of the absolute value of the moving cross-correlation with $W = 101$ indicated uncorrelated MDPCs when using MDPCA with $W = 101$; See Figure 3.37. However, hardly significant correlations existed between the two components for the cases where MDPCA was used with $W = 201$; See Figure 3.38. The correlations slightly increased when using MDPCA with $W = 301$; See Figure 3.39. Small standard deviations of the absolute value of the moving cross-correlation were reported in Table 3.16, where their values ranged between 0 to 0.08 in all cases.

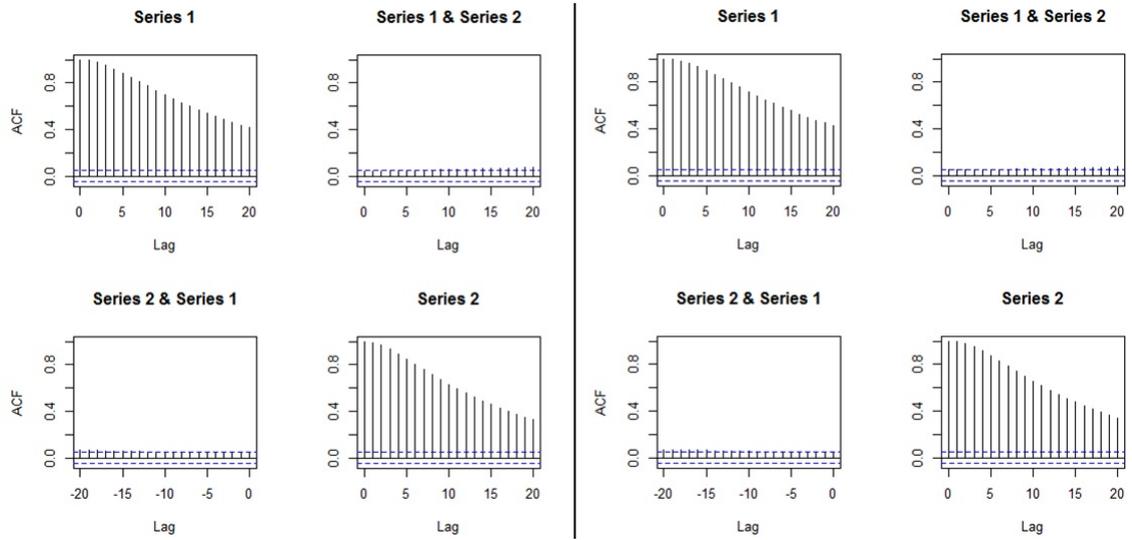


Figure 3.37: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.4 [Left: MDPCA ($W = 101, l = 1$), Right: MDPCA($W = 101, l = 5$)].

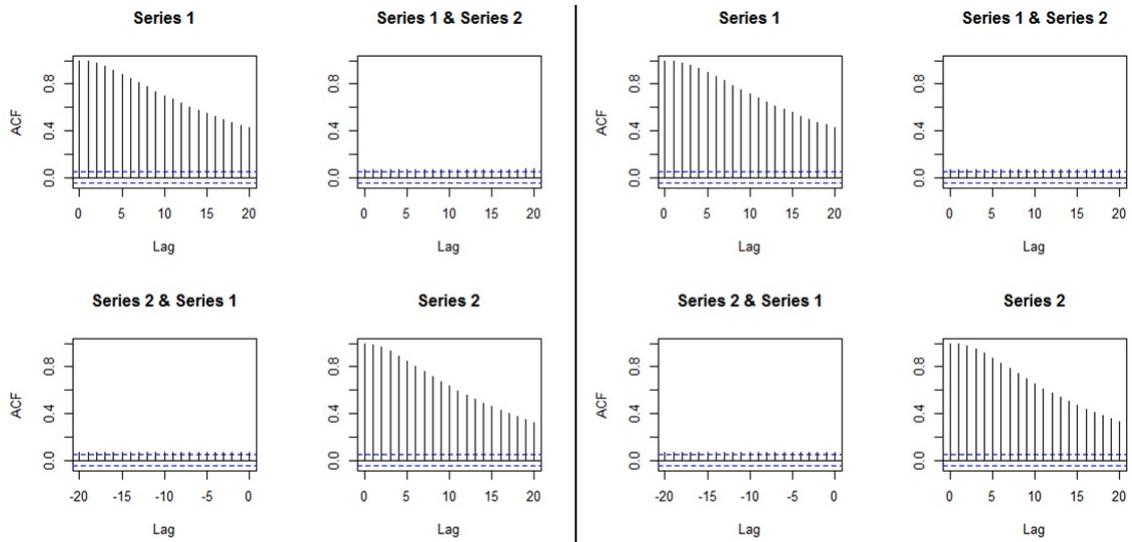


Figure 3.38: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.4 [Left: MDPCA ($W = 201, l = 1$), Right: MDPCA($W = 201, l = 5$)].

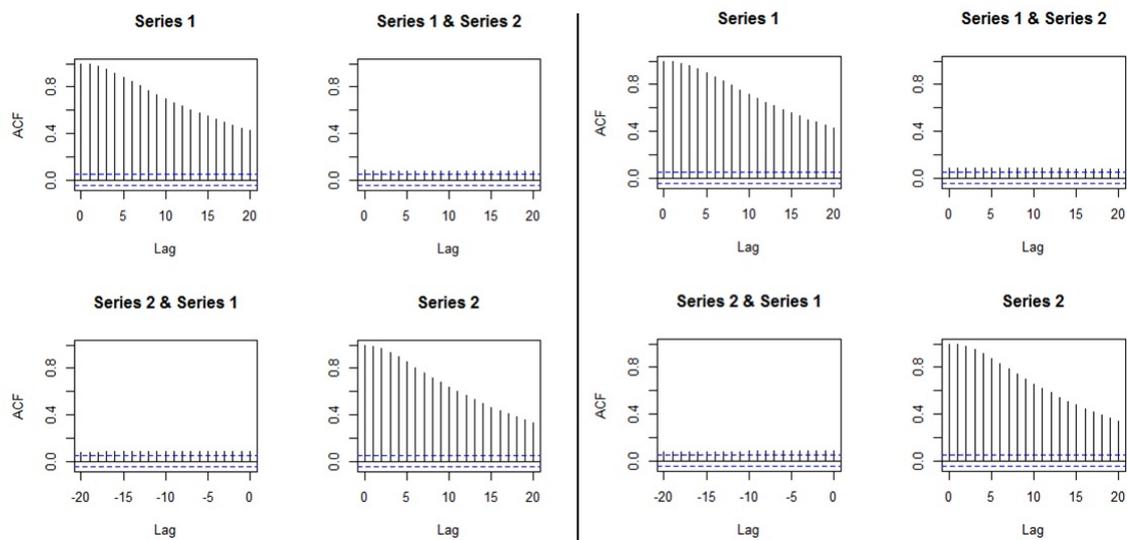


Figure 3.39: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs in Example 3.3.4 [Left: MDPCA ($W = 301$, $l = 1$), Right: MDPCA($W = 301$, $l = 5$)].

MDPCA(W=101, l=1)																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.04	0.04	0.04	0.04	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08
MDPC1 vs MDPC2	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC1	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.06
MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08
MDPCA(W=201, l=1)																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.04	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.08
MDPC1 vs MDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08	0.08
MDPCA(W=301, l=1)																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07
MDPC1 vs MDPC2	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08
MDPCA(W=101, l=5)																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07
MDPC1 vs MDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC2	0.00	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08
MDPCA(W=201, l=5)																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07
MDPC1 vs MDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
MDPC2 vs MDPC2	0.00	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08
MDPCA(W=301, l=5)																					
lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07
MDPC1 vs MDPC2	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
MDPC2 vs MDPC2	0.00	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08

Table 3.16: Standard deviations of the absolute value of the moving cross-correlations between two MDPCs with different combinations of W and l in Example 3.3.4.

Based on the above results, we can best reduce the dimension of the simulated non-stationary series with 15 variables in this example by using MDPCA with $W = 101$ and $l = 1$ or using MDPCA with $W = 101$ and $l = 5$.

To investigate the effect of changing the sample size of \mathbf{z}_t on the MDPCA, we applied MDPCA with $W = 101$ and $l = 1$ on \mathbf{z}_t with sample sizes $T = \{200, 400, 600, 800\}$. The results based on 500 replicas were summarized in Table 3.17, where two MDPCs were retained. Steady results, about 96.25%, for the mean percentage of explained variance were obtained in all cases with small standard deviations. For example, a mean percentage of 96.35% and a standard deviation of 0.02 were obtained when $T = 200$ compared with 96.22% and 0.01 when $T = 800$. The mean MSE of reconstructed data and the corresponding standard deviation increased as we increased T . For example, a mean MSE of 42.66 with a standard deviation of 28.68 was obtained when $T = 200$, compared with 183.39 and 129.85 when $T = 800$.

Additionally, the means of the absolute value of the moving cross-correlation with $W = 101$ are plotted in figures 3.40 and 3.41. It can be seen that the two MDPCs had significant cross-correlations when $T = 200$. The correlations became smaller with minor significant correlations for $|l| \geq 10$ when increasing the dimension to $T = 400$. Uncorrelated MDPCs were obtained when $T = 600$ and $T = 800$. The standard deviations of the absolute value of the moving cross-correlation are reported in Table 3.18, where the values could be improved by increasing the sample size. For instance, the standard deviations ranged from 0 to 0.23 when $T = 200$, compared with 0 to 0.11 when $T = 800$. Therefore, in this example, MDPCA performs well when applied to \mathbf{z}_t with $T \geq 600$. The MDPCA was able to reduce the dimension of \mathbf{z}_t from 15 to 2.

Method	MDPCA (W=101, l=1)			
Sample size	T=200	T=400	T=600	T=800
Contribution % Mean (Standard Deviation)	96.35 % (0.02)	96.25 % (0.01)	96.23 % (0.01)	96.22 % (0.01)
MSE Mean (Standard Deviation)	42.66 (28.68)	91.74 (63.65)	130.54 (92.59)	183.39 (129.85)

Table 3.17: A comparison between the results of MDPCA with different sizes of T in Example 3.3.4.

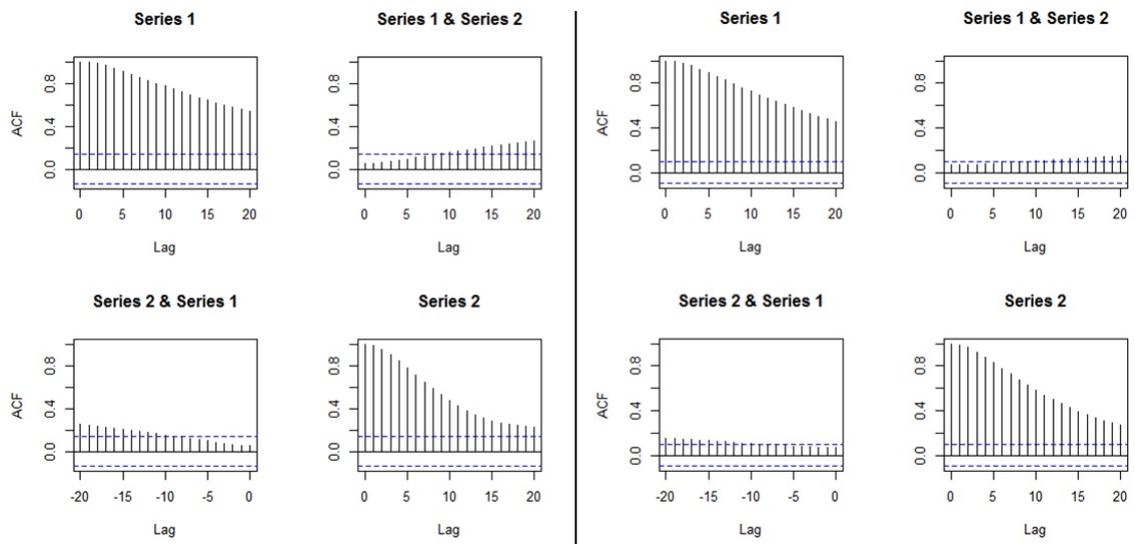


Figure 3.40: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.4 [Left: $T = 200$, Right: $T = 400$].

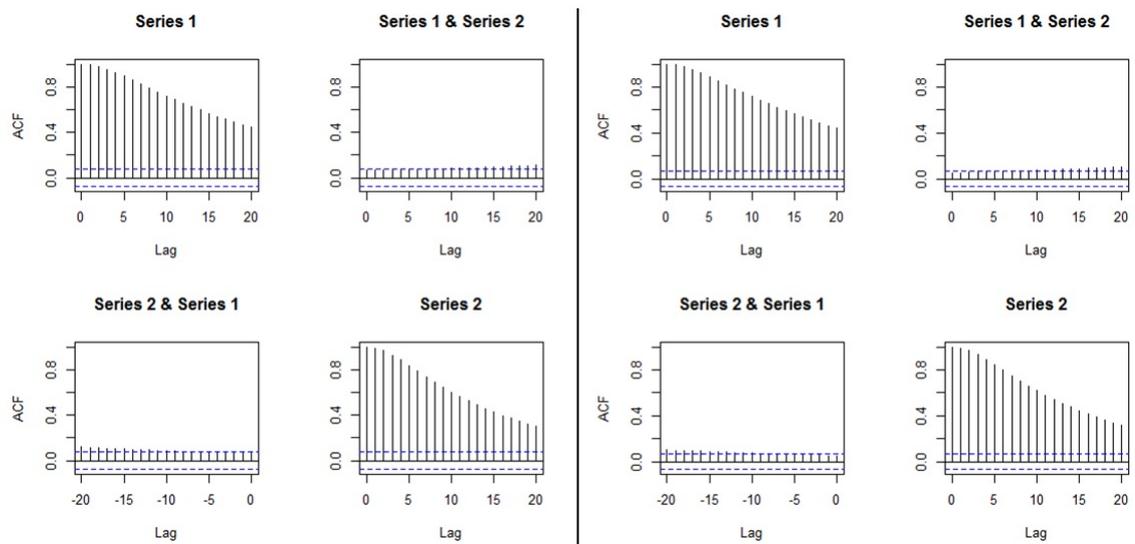


Figure 3.41: Plots of the mean of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.4 [Left: $T = 600$, Right: $T = 800$].

		T=200																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.02	0.03	0.04	0.06	0.07	0.09	0.10	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19	0.20	0.21	0.22	0.23
MDPC1 vs MDPC2	0.05	0.05	0.06	0.06	0.07	0.08	0.09	0.09	0.10	0.11	0.12	0.12	0.13	0.14	0.15	0.16	0.17	0.17	0.17	0.17	0.18	0.18
MDPC2 vs MDPC1	0.05	0.05	0.05	0.06	0.06	0.07	0.08	0.09	0.10	0.11	0.11	0.12	0.13	0.14	0.15	0.15	0.16	0.17	0.17	0.17	0.18	0.18
MDPC2 vs MDPC2	0.00	0.01	0.02	0.04	0.07	0.09	0.12	0.14	0.16	0.17	0.18	0.19	0.19	0.20	0.20	0.20	0.19	0.19	0.18	0.18	0.17	0.17
		T=400																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.06	0.07	0.08	0.09	0.10	0.11	0.11	0.12	0.13	0.13	0.13	0.14	0.14	0.15
MDPC1 vs MDPC2	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.10	0.11	0.11	0.11	0.11
MDPC2 vs MDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.10	0.10	0.11
MDPC2 vs MDPC2	0.00	0.00	0.01	0.02	0.03	0.05	0.06	0.07	0.09	0.10	0.11	0.12	0.13	0.13	0.13	0.14	0.14	0.15	0.15	0.15	0.15	0.15
		T=600																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.04	0.04	0.05	0.06	0.07	0.07	0.08	0.09	0.09	0.10	0.10	0.11	0.11	0.12	0.12
MDPC1 vs MDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.09	0.09
MDPC2 vs MDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.08	0.08	0.08
MDPC2 vs MDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.10	0.10	0.11	0.11	0.12	0.12	0.12	0.12	0.12	0.13
		T=800																				
lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.04	0.04	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.09	0.09	0.10	0.10	0.10
MDPC1 vs MDPC2	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07
MDPC2 vs MDPC1	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07
MDPC2 vs MDPC2	0.00	0.00	0.01	0.02	0.02	0.03	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.11	0.12	0.12	0.12	0.12	0.12	0.13

Table 3.18: Standard deviations of the absolute value of the moving cross-correlations between two MDPCs with different sizes of T in Example 3.3.4.

By the end of the above simulation studies, we conclude that the proposed MDPCA is able to reduce the dimension of a multivariate time series by taking into account the dynamic and non-stationary behaviours of the data. Determining the window size has a significant effect on extracting accurate results from non-stationary series. Additionally, using a window size that is larger than required could cause MDPCA to produce correlated MDPCs. However, it can be shown that the opposite is not true, where MDPCA can produce accurate results using a window size that is smaller than required.

Including more lagged series in the analysis expands the dimension of the data inputted into MDPCA. However, this could add more dynamic information to the retained MDPCs. It was noticed that when more lagged series were included in the analysis, MDPCA produced MDPCs with slightly lower contribution percentages and slightly higher reconstruction error compared to those when fewer lagged series were used.

It was also noticed that the performance of MDPCA was steady even though we increased the dimension of tested time series. Notice that it is not sufficient to evaluate MDPCA's results based on the contribution percentage and the reconstruction error only. MDPCA could produce a small number of components that explain almost all variation in the data and produce small reconstruction error when applied to non-stationary series with small sample sizes. However, due to the one-step movement in the moving cross-covariance matrix, the retained components could have minor, but significant, correlations for small sample sizes. These correlations become non-significant by increasing the sample size. In the simulations above, MDPCA produced reasonable results on series with sample sizes $T \geq 400$. Therefore, it is essential to investigate the correlations between the retained components in order to obtain valid results.

3.3.2 Real Data Examples

Example 3.3.5 In this example, we apply the MDPCA to daily stock prices of ten US companies in US Dollar from Nov 07, 2013 to Dec 18, 2017. The sample size of the data is 1036 days. The variables in this example represent stock prices of the following companies: Facebook (FB), Ford Motor (Ford), General Electric (GE), General Motor (GM), The Home Depot (HD), IBM (IBM), Intel (INTC), Johnson & Johnson (JNJ), Kroger (KR) and Lowe's (LOW). The data is available on *Yahoo! Finance*.

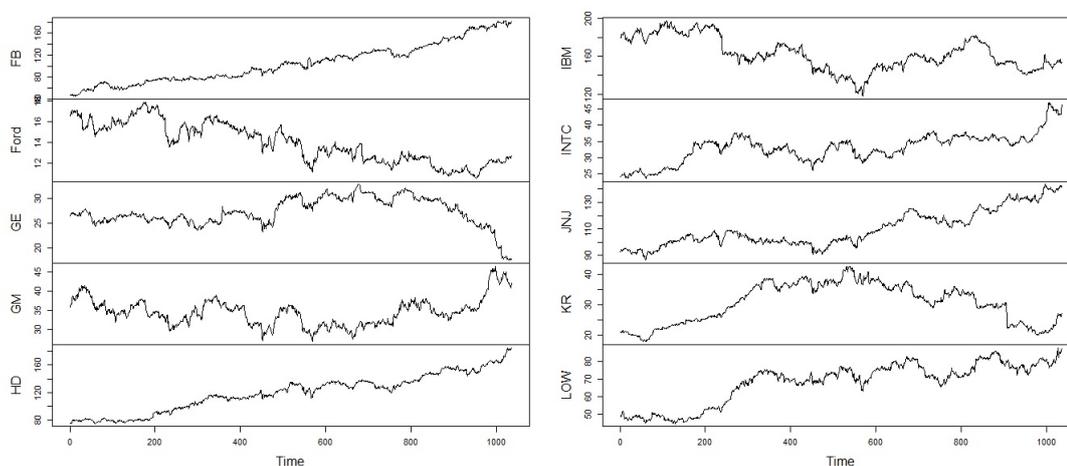


Figure 3.42: The daily stock prices of the ten US companies from November 2013 to December 2017 in Example 3.3.5.

A time series plot of the daily stock prices of the US companies is shown in Figure 3.42, where all ten companies exhibited non-stationary behaviours over time. Sample moving cross-correlation plots with $W = 101$ were used to show the dynamic relationships among the stock prices before the MDPCA was applied; See figures 3.43, 3.44, 3.45 and 3.46. Generally, a moderated dynamic relationship existed among the ten companies. For example, the companies General Motor and Ford Motor were strongly correlated for all time lags. Also, The Home Depot and Kroger had a moderate dynamic relationship. Moreover, a weak relationship existed between companies Intel and Facebook.

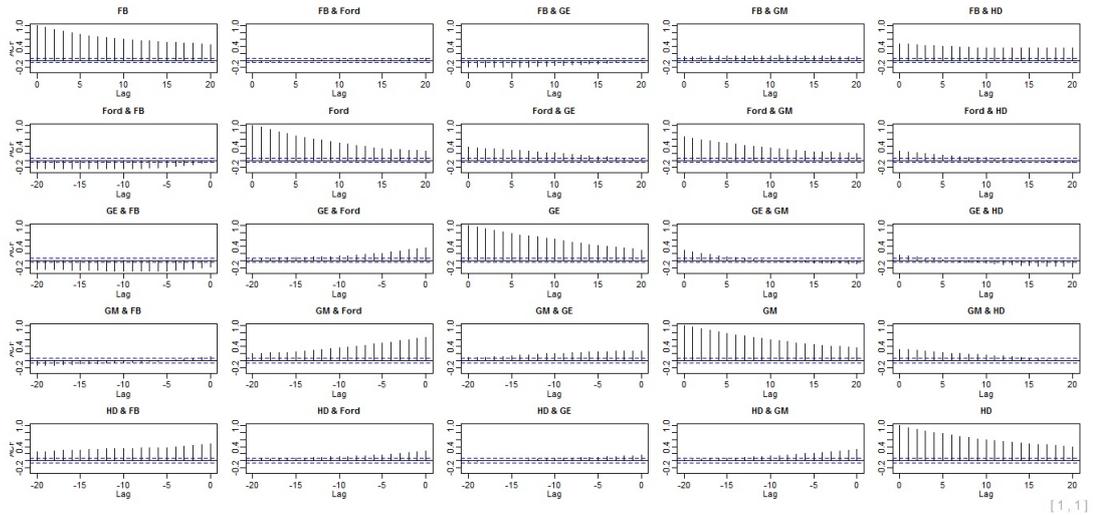


Figure 3.43: Sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the ten US companies before MDPCA is applied in Example 3.3.5 - Part 1.

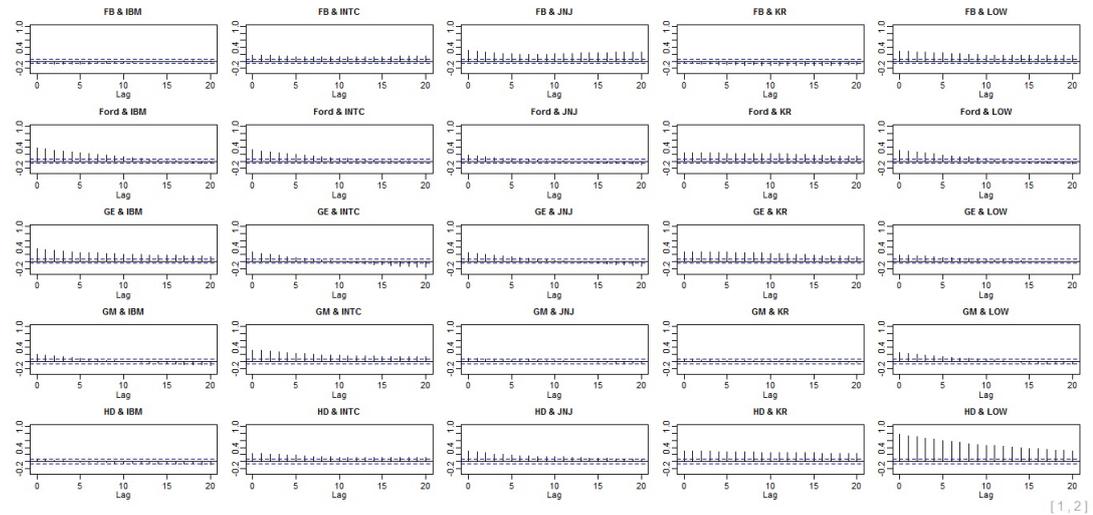


Figure 3.44: Sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the ten US companies before MDPCA is applied in Example 3.3.5 - Part 2.

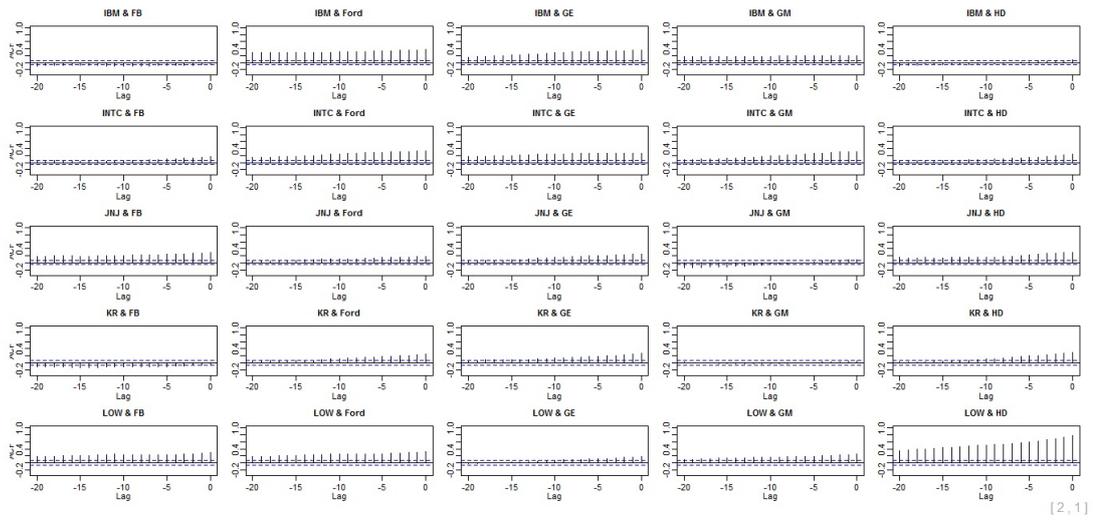


Figure 3.45: Sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the ten US companies before MDPCA is applied in Example 3.3.5 - Part 3.

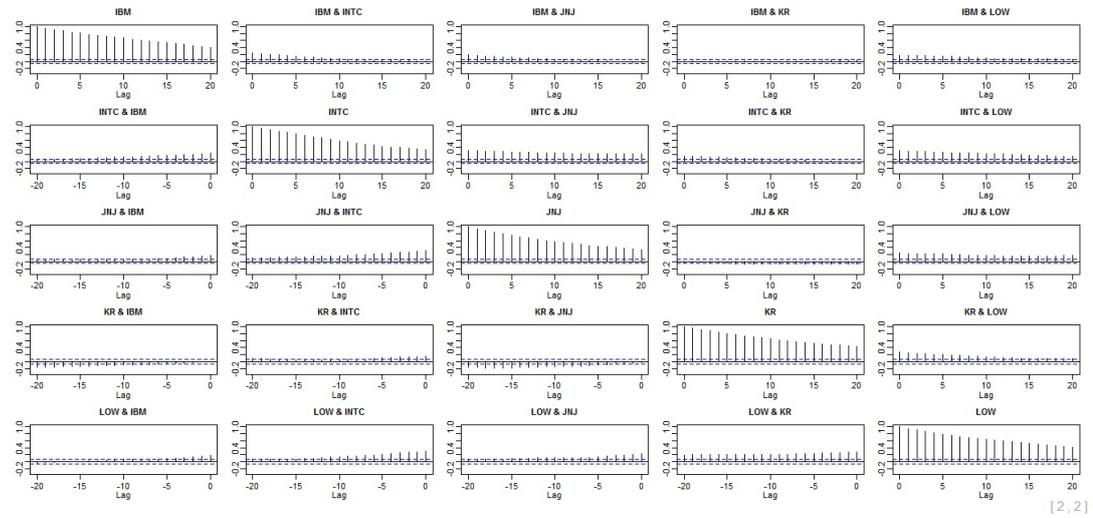


Figure 3.46: Sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the ten US companies before MDPCA is applied in Example 3.3.5 - Part 4.

Considering the stationarity of the data, MDPCA with $W = 101$ was used. The optimal number of lags was suggested to be $l = 1$, as shown in Figure 3.47, where six static and four dynamic relations were found. A plot of the eigenvalues is provided in Figure 3.48 along with the RCC criterion plot. The RCC of the first two MDPCs

had a value of 0.53, compared with 0.49 and 0.52 for the first three and the first four MDPCs respectively. Therefore, three MDPCs were suggested by the RCC criterion. The three retained MDPCs explained 84.54% of the total variation in the data and produced a reconstruction error of 292.28. The sample moving cross-correlation plots of the three MDPCs are shown in Figure 3.49, which suggested uncorrelated MDPCs.

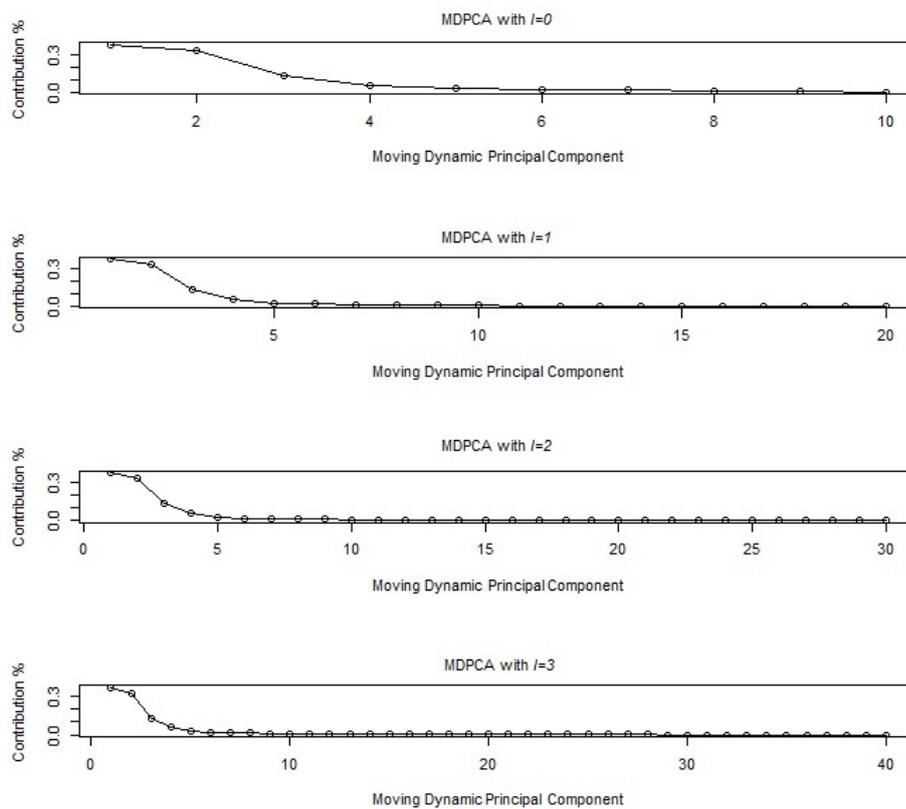


Figure 3.47: Eigenvalues' plots after applying MDPCA with $W = 101$ to the ten US companies in Example 3.3.5.

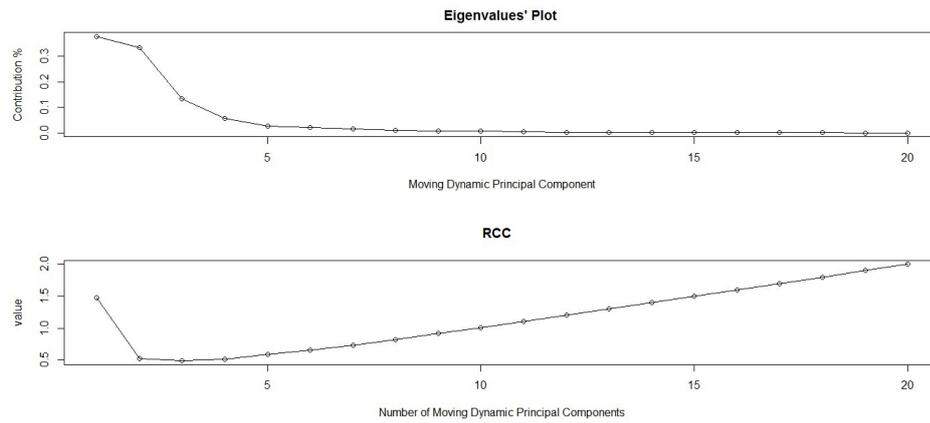


Figure 3.48: Top: An eigenvalues' plot of the ten US companies in Example 3.3.5 after applying MDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot.

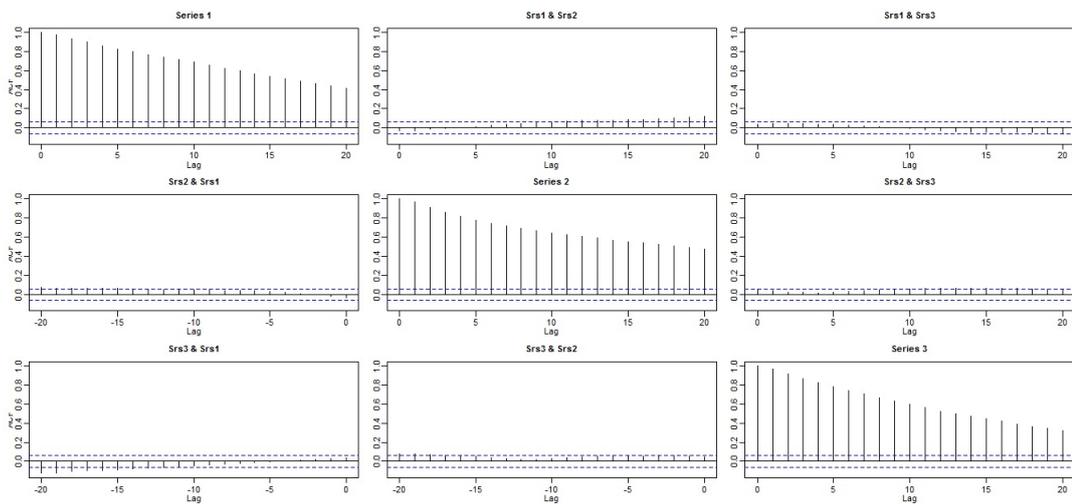


Figure 3.49: Sample moving cross-correlation plots of the three MDPCs of the ten US companies after MDPCA with $W=101$ and $l=1$ is applied in Example 3.3.5.

In conclusion, by applying MDPCA with $W = 101$ and $l = 1$ directly to the non-stationary series data of the ten US stock prices, we were able to reduce its dimension from 10 to 3.

Example 3.3.6 In this example, the MDPCA is applied to daily stock prices of 17 US companies in US Dollar from Nov 07, 2013 to Dec 18, 2017. The sample size of the data is 1036 days. The data set in this example is obtained by adding seven

new companies to the data in Example 3.3.5. The names of the 17 companies are shown in table 3.19. The data is available on *Yahoo!* Finance.

No.	Label	Name	No.	Label	Name
1	ESRX	Express Scripts Holding	10	KR	Kroger-The Kroger
2	FB	Facebook	11	LOW	Lowe's
3	F	Ford Motor	12	MCK	McKesson
4	GE	General Electric	13	MET	MetLife Insurance
5	GM	General Motor	14	MSFT	Microsoft
6	HD	The Home Depot	15	PEP	PepsiCo
7	IBM	IBM	16	PG	The Procter & Gamble
8	INTC	Intel	17	PRU	Prudential Financial
9	JNJ	Johnson & Johnson			

Table 3.19: Names and labels of the 17 US companies in Example 3.3.6.

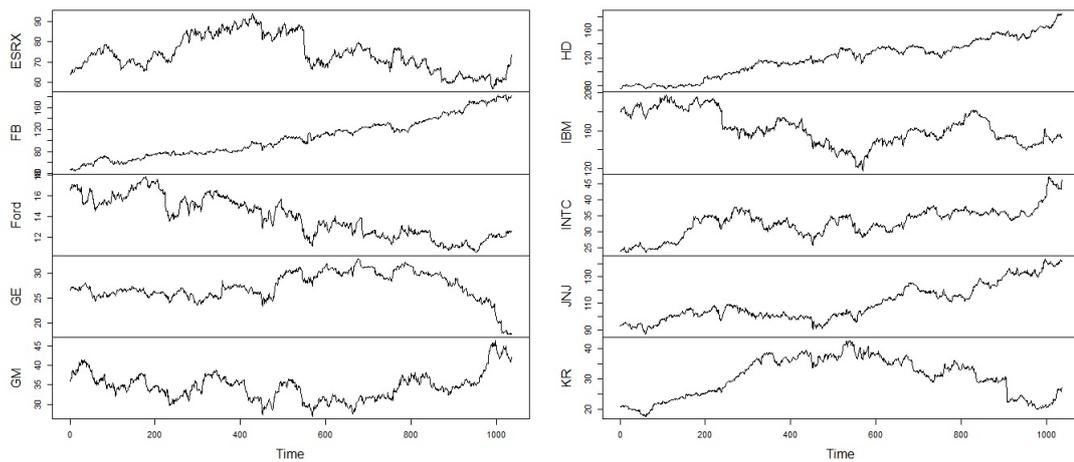


Figure 3.50: A time series plot of the daily stock prices of the 17 US companies in Example 3.3.6- Part 1

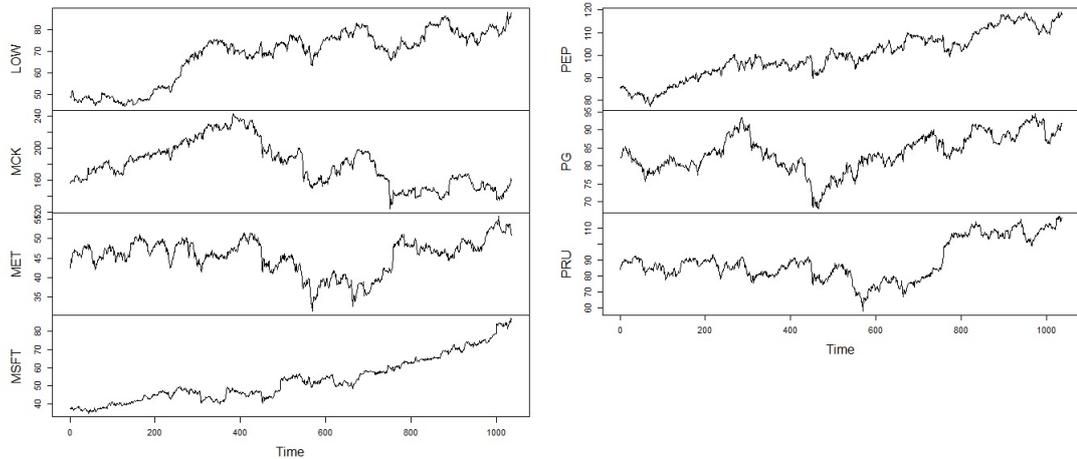


Figure 3.51: A time series plot of the daily stock prices of the 17 US companies in Example 3.3.6- Part 2

The time series plots of the daily stock prices of the 17 US companies revealed the non-stationary behaviour of the daily stock prices; See figures 3.50 and 3.51. Figure 3.52 shows the last 36 sample moving cross-correlation plots with $W = 101$ of the stock prices before the transformation. The daily stock prices of the 17 companies were moderately correlated. For example, the company MetLife Insurance was strongly correlated with Prudential Financial and weakly correlated with McKesson.

Considering the stationarity of the data, MDPCA with $W = 101$ could be used. The optimal number of lags was suggested to be $l = 1$ as shown in Figure 3.53, where ten static and seven dynamic relations were found. Figure 3.54 consists of the eigenvalues' plot along with the relative RCC criterion plot. The RCC had values of 0.508, 0.490, 0.489, 0.486 and 0.502 for the first three, four, five, six and seven MDPCs, respectively. Thus, the optimal number of MDPCs to retain is six MDPCs, as suggested by the RCC criterion. The six MDPCs explained 90.25% of the total variation in the data and produced a reconstruction error of 304.55. The retained six MDPCs were uncorrelated, as shown in the sample moving cross-correlation plots in Figure 3.55.

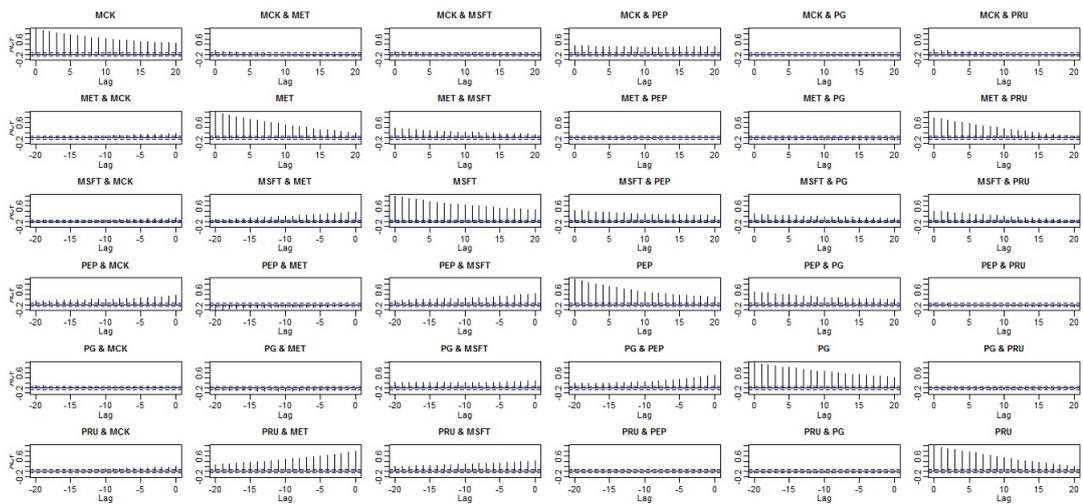


Figure 3.52: Last 36 sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the 17 US companies before MDPCA is applied in Example 3.3.6.

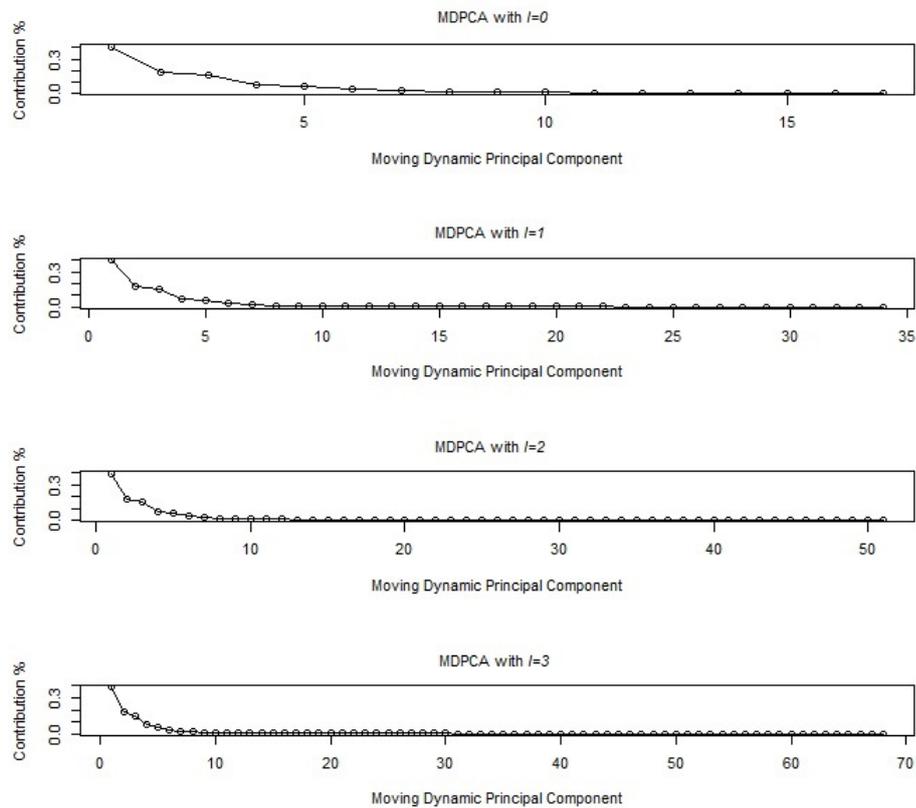


Figure 3.53: Eigenvalues' plots after applying MDPCA with $W = 101$ to the 17 US companies in Example 3.3.6.

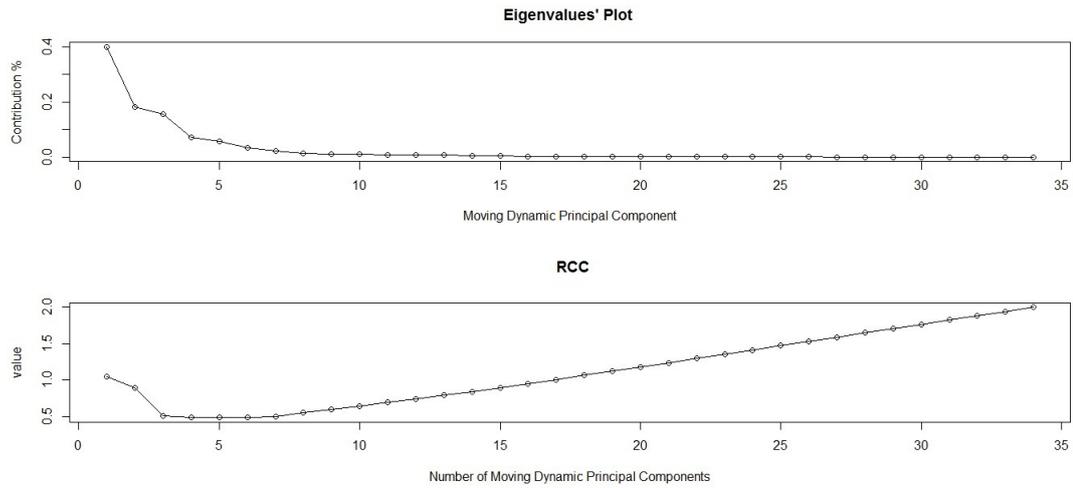


Figure 3.54: Top: An eigenvalues' plot of the 17 US companies in Example 3.3.6 after applying MDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot.

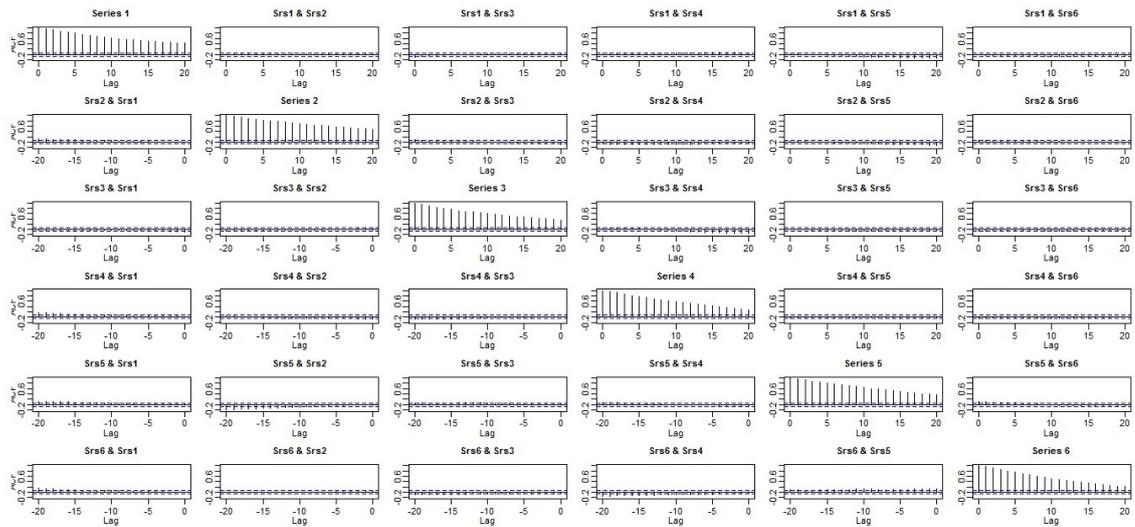


Figure 3.55: Sample moving cross-correlation plots of the MDPCs of the 17 US companies in Example 3.3.6 after MDPCA with $W=101$ and $l=1$ is applied.

To conclude, the MDPCA with one lag and a window of size 101 was able to reduce the dimension of the daily stock prices of the US companies from 17 to 6 by accounting for the non-stationarity and the dynamic dependence in the stock prices.

3.4 Summary and Remarks

In this chapter, we introduced MDPCA, which is a PCA-based dimension reduction method that is used to reduce the dimension of multivariate time series by transforming them into uncorrelated components. MDPCA is a generalization of DPCA method of Ku et al. (1995) to non-stationary series. DPCA can be considered as a special case of MDPCA when $W = N$. MDPCA can be used to reduce the dimension of both stationary and non-stationary series by accounting for the static, dynamic and non-stationary natures of time series data.

We used three methods to assess MDPCA's results. The moving cross-correlation function can evaluate the dynamic relationships between the final retained MDPCs. The MSE of reconstructed data and the percentage of explained variance are another two tools to evaluate retained MDPCs where lower reconstruction errors and higher percentages of contribution are indications of better results.

Choosing the window size for MDPCA depends on the stationarity of the data. Smaller windows are suitable for series with stronger non-stationary behaviour, and the opposite is true. The RCC criterion can be used to choose the optimal number of MDPCs to retain. The RCC criterion balances between the desire of reducing the dimension of the data and increasing the accuracy of the final results. Additionally, the RCC criterion can be employed to determine the optimal number of components to retain in PCA-based reduction methods.

We studied the asymptotic properties of our estimator $\hat{\mathbf{U}}$, the matrix that consists of the eigenvectors of the moving cross-covariance matrix of the data. Under some regularity assumptions, we show that $\hat{\mathbf{U}}$ is a consistent estimator of \mathbf{U} with $W^{-1/2}$ convergence rate.

We provided many simulation studies considering non-stationary series with different dimensional and sample sizes. MDPCA was able to reach dimension reduction and performed well even for reasonably moderate sample sizes. MDPCA was also

illustrated on real-life stock prices with different dimensions.

The performance of MDPCA was steady even though we increased the dimension of tested time series. It was noticed that MDPCA required larger sample sizes when applied to non-stationary series compared with stationary series. This due to the one-step movement in the moving cross-covariance matrix, which causes the retained MDPCs to have minor, but significant, correlations. These correlations become non-significant by increasing the sample size.

Chapter 4

Quadratic Moving Dynamic

Principal Component Analysis

In this chapter, we are going to introduce a new dimension reduction method that can be applied on time series that are allowed to be non-stationary. This method can be considered as an alternative to MDPCA that we introduced previously in Chapter 3. The method we propose in this chapter will use a quadratic order of the moving cross-covariance function that we introduced in Chapter 3. The new method will be named quadratic moving dynamic principal component analysis (QMDPCA). In what follows, we are going to show the building-structure of QMDPCA. We are also going to discuss the theoretical properties of the used estimators. The performance of QMDPCA will be tested on many simulated data sets. Additionally, our method will be illustrated on real data sets.

4.1 Methodology

Suppose an m -dimensional time series $\mathbf{z}_t = (z_{1,t}, z_{2,t}, \dots, z_{m,t})'$ that is allowed to be non-stationary. QMDPCA is aiming to reduce the dimension of \mathbf{z}_t by considering the dynamic relationships that exist between the variables of \mathbf{z}_t . These relationships can be captured by including lagged values of \mathbf{z}_t into the analysis. Denote by \mathbf{y}_t the extended data vector of \mathbf{z}_t , which consists of \mathbf{z}_t and its lags up to a specified lag l . Hence, \mathbf{y}_t is an $m(l+1)$ -dimensional vector. In specific, \mathbf{y}_t has the following

structure

$$\mathbf{y}_t = (\mathbf{z}'_t, \mathbf{z}'_{t-1}, \dots, \mathbf{z}'_{t-l})' = (z_{1,t}, \dots, z_{m,t}, z_{1,t-1}, \dots, z_{m,t-1}, \dots, z_{1,t-l}, \dots, z_{m,t-l})'$$

In the rest of its analyses, QMDPCA is going to use the extended data vector \mathbf{y}_t to extract the dynamic relationships of \mathbf{z}_t . Assume the series \mathbf{z}_t to be observed over T time points. Let $m(l+1) = M$ and $T-l = N$. Let \mathbf{Y} be the $M \times N$ extended data matrix whose columns are $\mathbf{y}_1, \dots, \mathbf{y}_N$.

QMDPCA seeks dimension reduction by applying eigenanalysis on a quadratic order of the moving cross-covariance matrix. In order to state a formal definition of the quadratic order moving cross-covariance matrix, recall the following definitions from the previous chapter. We assume that \mathbf{y}_i to be locally stationary. The cross-covariance matrix of \mathbf{y}_i is defined as

$$\Gamma_i = \text{Var}(\mathbf{y}_i). \quad (4.1)$$

Once the data is observed, the sample cross-covariance matrix defined over window \mathbf{y}_i with a pre-specified size $W = 2w + 1$ can be calculated as

$$\hat{\Gamma}_i = \frac{1}{2w+1} \sum_{t=i-w}^{i+w} (\mathbf{y}_t - \bar{\mathbf{y}}_i)(\mathbf{y}_t - \bar{\mathbf{y}}_i)' \quad (4.2)$$

where

$$\bar{\mathbf{y}}_i = \frac{1}{2w+1} \sum_{t=i-w}^{i+w} \mathbf{y}_t$$

where w is a positive integer. The matrix $\hat{\Gamma}_i$ is calculated over a moving local window \mathbf{y}_i of a pre-determined size, where i is the window's index. The window size is determined according to the stationarity of the data as we will discuss latter in this section. By moving the window \mathbf{y}_i one step in time, $\hat{\Gamma}_i$ can be recalculated to again to extract the dynamic dependence from the data located on the next window, and so on. These local matrices are used to build the moving cross-covariance matrix of \mathbf{y}_t , which is defined to be

$$\mathbf{M}\Gamma = \frac{1}{N-2w} \sum_{i=w+1}^{N-w} \Gamma_i. \quad (4.3)$$

Recall that the moving cross-covariance matrix $\mathbf{M}\Gamma$ is able to extract the dynamic relationships out of both stationary and non-stationary time series, as shown in Chapter 3. Based on observed data, the moving cross-covariance matrix of \mathbf{y}_t can be estimated by the sample moving cross-covariance matrix $\hat{\mathbf{M}}\Gamma$ as follows

$$\hat{\mathbf{M}}\Gamma = \frac{1}{N-2w} \sum_{i=w+1}^{N-w} \hat{\Gamma}_i . \quad (4.4)$$

where $\hat{\Gamma}_i$ is an estimate of Γ_i , as defined in (4.2).

Then, the quadratic order moving cross-covariance matrix of \mathbf{y}_t is calculated as

$$\mathbf{Q} = \mathbf{M}\Gamma[\mathbf{M}\Gamma]'. \quad (4.5)$$

Notice that \mathbf{Q} is an $M \times M$ symmetric matrix. Additionally, \mathbf{Q} can extract dynamic information from both stationary and non-stationary series as we will show in the examples' section. Based on sample data, the sample quadratic order moving cross-covariance matrix of \mathbf{y}_t can be calculated by

$$\hat{\mathbf{Q}} = \hat{\mathbf{M}}\Gamma[\hat{\mathbf{M}}\Gamma]', \quad (4.6)$$

where $\hat{\mathbf{M}}\Gamma$ is defined in (4.4).

Moreover, spectral decomposition of the quadratic moving cross-covariance matrix of \mathbf{y}_t yields

$$\mathbf{Q} = \mathbf{U}\Lambda\mathbf{U}', \quad (4.7)$$

where Λ is a diagonal $M \times M$ matrix whose diagonal entries are the eigenvalues of \mathbf{Q} , and \mathbf{U} is an orthogonal $M \times M$ matrix whose columns are the eigenvectors of \mathbf{Q} . Denote by λ_j , $1 \leq j \leq M$, the j^{th} eigenvalue of \mathbf{Q} , which is the $(j, j)^{\text{th}}$ element of matrix Λ in (4.7). Notice that the eigenvalues of \mathbf{Q} are ordered in along the diagonal such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_M$. Denote by \mathbf{u}_j the j^{th} eigenvector of \mathbf{Q} , which is the j^{th} column of \mathbf{U} that correspond to λ_j . Then, M uncorrelated quadratic moving dynamic principal components (QMDPCs) can be produced by left multiplying \mathbf{y}_t by the transpose matrix of \mathbf{U} (i.e. \mathbf{U}'). These QMDPCs are used to represent \mathbf{y}_t .

Let \mathbf{q}_j be the j^{th} QMDPC, then the amount of variation in \mathbf{y}_t that is explained by \mathbf{q}_j is given by $\frac{\lambda_j}{\lambda_1 + \dots + \lambda_M}$. Consequently, QMDPCA reduces the dimension of \mathbf{z}_t by retaining only the first k^{th} QMDPCs such that

$$\frac{\lambda_1 + \dots + \lambda_k}{\lambda_1 + \dots + \lambda_M} \simeq 1 . \quad (4.8)$$

These k QMDPCs represent \mathbf{z}_t with fewer dimensions ($k < m$) as they account for almost all variation in \mathbf{z}_t . For instance, if the first eigenvalue of \mathbf{Q} is large enough, then the dimension of \mathbf{z}_t can be reduced to one by projecting \mathbf{y}_t into \mathbf{q}_1 (i.e. the first QMDPC) as follows

$$\mathbf{q}_1 = \mathbf{u}'_1 \cdot \mathbf{y}_t.$$

Furthermore, the original data can be reconstructed by

$$\mathbf{y}_t^{\text{recon}} = \mathbf{u}_1 \cdot \mathbf{q}_1 \quad (4.9)$$

where $\mathbf{y}_t^{\text{recon}}$ is used to denote the reconstructed data after QMDPCA is applied. Notice that the optimal value for k is the minimum number of QMDPCs that consist of the maximum variation of the data and produced the minimum reconstruction error. Determining the optimal choice of k will be discussed in the next section.

4.1.1 Optimizing the Results of QMDPCA

This section provides useful tools to optimize the outcomes of QMDPCA. There are three aspects to consider in order to enhance QMDPCA's outcomes: determining the window size W , the number of lagged series l and the number of retained QMDPCs.

Choosing a size for W would affect the accuracy of QMDPCA's results. In particular, the size of W has a major effect on the ability of $\mathbf{M}\Gamma$ in (4.3) to extract accurate information from the data, as shown in Chapter 3. Hence, the size of W will also affect the accuracy of \mathbf{Q} in (4.7). As far as we know, there is no available objective criterion to determine the optimal size for W in literature, as this could be a subject of further research. Nevertheless, choosing the size of W can be done by

using the time series plot of the data and assessing its stationarity. Shorter windows are suitable for series with stronger non-stationarity, and wider windows are suitable for stationary series. For example, for a stationary series, one can use QMDPCA with one window that consists of the whole data (i.e. $W = N$).

In order to obtain informative results, one shall determine the optimal number of lagged series of \mathbf{z}_t that are included to build the extended data vector \mathbf{y}_t . Generally speaking, including more lagged series would add more dynamic information to the analysis. However, it would also lead to more complicated calculations as the number of QMDPCs will be increased. To this end, we are going to adapt the procedure introduced in Section 3.1 in the previous chapter to choose the number of lagged series to include in the analysis, which can be extended to QMDPCA as follows:

1. Start with $l = 0$.
2. Build the extended data vector \mathbf{y}_t by including l lagged series.
3. Apply QMDPCA to \mathbf{y}_t and obtain all QMDPCs.
4. Set $j = m(l + 1)$ and $r(l) = 0$, where r is the number of relations.
5. Determine if the j^{th} component \mathbf{q}_j provides a linear relation. If yes, go to next step, otherwise go to step 7.
6. Set $j = j - 1$ and $r = r(l) + 1$, then repeat step 5.
7. Calculate the number of new relations added by including one more lag, $r_{\text{new}}(l)$, by:

$$r_{\text{new}}(l) = r(l) - \sum_{k=0}^{l-1} (l - k + 1)r_{\text{new}}(k),$$

if $r_{\text{new}}(l) \leq 0$, go to step 9, otherwise go to next step.

8. Set $l = l + 1$, go to step 2.
9. Stop.

Notice that step 5 can be done by plotting the eigenvalues of $\hat{\mathbf{Q}}$, where the number of relations r can be obtained by subtracting the number of significant QMDPCs from the total number of QMDPCs. More information and examples about the usage of the above procedure are available in Section 3.1 in Chapter 3.

Another critical step after applying QMDPCA is to determine the optimal number of QMDPCs to retain (i.e. k). This can be done by balancing between the following desires: retaining the minimum number of QMDPCs, maximizing the amount of variation explained by the retained QMDPCs as expressed in (4.8) and minimizing the reconstruction error.

In order to determine the optimal number of QMDPCs objectively, we are going to use the Retained Component Criterion RCC proposed in Section 3.1 in the previous chapter. Recall that the amount of error of reconstructing \mathbf{y}_t by the first k QMDPCs can be calculated as follows:

$$\text{MSE}_k(\mathbf{y}_t, \mathbf{y}_t^{\text{recon}}) = \frac{1}{MN} \sum_{j=1}^M \sum_{t=1}^N (y_{j,t} - \sum_{v=1}^k u_{j,v} q_{v,t})^2 \quad (4.10)$$

where $\mathbf{y}_t^{\text{recon}}$ is the reconstructed data by the first k QMDPCs. Hence, we would like to minimize the reconstruction error in (4.10). Then, the RCC criterion of the first k QMDPCs can be calculated as

$$\text{RCC}_k = 2 - \left(\sum_{j=1}^k \lambda_j / \sum_{j=1}^M \lambda_j \right) - ((\text{MaxMSE} - \text{MSE}_k) / \text{MaxMSE}) + (2k/M) \quad (4.11)$$

where λ_j is the largest j^{th} eigenvalue of the matrix \mathbf{Q} defined in (4.5), and MSE_k is defined in (4.10). Furthermore, MaxMSE is the Maximum MSE of reconstruction that can be calculated using

$$\text{MaxMSE} = \frac{1}{MN} \sum_{j=1}^M \sum_{t=1}^N (y_{j,t})^2. \quad (4.12)$$

The RCC criterion can objectively balance between the above desires and therefore suggests the optimal number of QMDPCs to retain. See Section 3.1 in Chapter 3 for more information and examples about the RCC criterion.

4.1.2 QMDPCA Calculation Procedure

In order to reduce the dimension of an observed multivariate time series \mathbf{z}_t using QMDPCA, the following steps are used:

1. Build the extended data vector \mathbf{y}_t by including lagged series of \mathbf{z}_t up to lag l .
2. Calculate the sample quadratic order moving cross-covariance matrix $\hat{\mathbf{Q}}$ based on \mathbf{y}_t , as defined in (4.5).
3. Calculate the eigenvalues' matrix $\hat{\Lambda}$ and the corresponding eigenvectors' matrix $\hat{\mathbf{U}}$ based on $\hat{\mathbf{Q}}$ obtained in step 2.
4. Determine the optimal number of retained QMDPCs, k , by using the RCC criterion.
5. The transformed series with reduced dimension is obtained by left multiplying \mathbf{y}_t by the transpose of the first k columns of $\hat{\mathbf{U}}$ (i.e. first k eigenvectors of $\hat{\mathbf{Q}}$), where k is determined in step (4).

4.1.3 Evaluating Dynamic Relationships

Since QMDPCA deals with time series that can be non-stationary, then the moving cross-correlation function introduced in the previous chapter can be used to evaluate the dynamic dependence between its components. Recall, the lag l sample moving cross-correlation matrix of \mathbf{y}_t is defined as follows

$$\hat{\mathbf{M}}\rho(l) = \frac{1}{N - 2 \max(l, w)} \sum_{i=\max(l, w)+1}^{N-\max(l, w)} \hat{\mathbf{S}}_i^{-1} \hat{\Gamma}_i(l) \hat{\mathbf{S}}_{i-l}^{-1}, \quad (4.13)$$

where $\hat{\mathbf{S}}_i$ is the diagonal matrix of the standard deviations of \mathbf{y}_i , and $\hat{\Gamma}_i(l)$ is the sample lag l cross-covariance matrix defined over \mathbf{y}_i . Notice that the (j, j) th element of $\hat{\mathbf{S}}_i$ will be the square root of the (j, j) th element of $\hat{\Gamma}_i(0)$. Recall that $\hat{\mathbf{M}}\rho(l)$ can be used to evaluate the relationships between the components of both stationary and non-stationary series by adjusting the window size. More importantly, $\hat{\mathbf{M}}\rho(l)$ will be used to evaluate the relationships between the retained QMDPCs.

4.2 Theoretical Properties

The theoretical properties of the QMDPCA method are studied here. Recall that the estimated QMDPCs are obtained by left multiplying the extended data vector \mathbf{y}_t by the transpose of the eigenvectors' matrix $\hat{\mathbf{U}}$ of the matrix $\hat{\mathbf{Q}}$ in (4.6). Therefore, the convergence of QMDPCs is implied by the convergence of $\hat{\mathbf{U}}$. Hence, our goal is to show that $\hat{\mathbf{U}}$ is a consistent estimator of \mathbf{U} .

In order to derive the asymptotic properties of our estimator, we employ the distance measurement defined in (3.20) in Chapter 3. Recall, for some constants $c_1 < c_2$, let \mathbf{B}_1 and \mathbf{B}_2 be any $c_2 \times (c_2 - c_1)$ matrices satisfying the condition $\mathbf{B}_i' \mathbf{B}_i = \mathbf{I}_{(c_2 - c_1) \times (c_2 - c_1)}$, where $i \in \{1, 2\}$, then the distance between \mathbf{B}_1 and \mathbf{B}_2 is

$$D(\mathbf{B}_1, \mathbf{B}_2) = \sqrt{c_2 - c_1 - \text{tr}(\mathbf{B}_1 \mathbf{B}_1' \mathbf{B}_2 \mathbf{B}_2')} .$$

Here, we can approach the convergence of $\hat{\mathbf{U}}$ by showing that $D(\mathcal{M}(\hat{\mathbf{U}}), \mathcal{M}(\mathbf{U})) \rightarrow 0$ as $W \rightarrow \infty$, where $\mathcal{M}(\mathbf{U})$ is the linear space spanned by the columns of \mathbf{U} . Furthermore, an important assumption to obtain the results is the mixing assumption. We consider the mixing coefficient defined in (3.21) in Chapter 3:

$$\theta_l = \sup_c \sup_{A \in \mathcal{F}_{-\infty}^c, B \in \mathcal{F}_{c+l}^\infty} |P(A \cap B) - P(A)P(B)| .$$

Here, $\mathcal{F}_{c_3}^{c_4}$ is the σ -field generated by \mathbf{y}_t for $c_3 \leq t \leq c_4$, where c_3 and c_4 are constants. Recall that $\theta_l = 0$ if A and B are independent.

In order to investigate the theoretical properties of $\hat{\mathbf{U}}$ further, we shall notice that the convergence of $\hat{\mathbf{Q}}$ implies the convergence of $\hat{\mathbf{U}}$ as the latter consists of the eigenvectors of $\hat{\mathbf{Q}}$. Furthermore, since $\hat{\mathbf{Q}}$ is obtained by multiplying $\hat{\mathbf{M}}\hat{\Gamma}$ by its transpose, then the convergence of $\hat{\mathbf{Q}}$ is implied by the convergence of $\hat{\mathbf{M}}\hat{\Gamma}$. Recall that $\hat{\mathbf{M}}\hat{\Gamma}$ is calculated based on the cross-covariance matrices of the local windows of \mathbf{y}_i s as defined in (4.4). Hence, the convergence of $\hat{\mathbf{M}}\hat{\Gamma}$ is implied by the convergence $\hat{\Gamma}_i$.

In this section, we study the convergence of $\hat{\mathbf{U}}$, assuming the dimension M is fixed. We specify the required assumptions to obtain the convergence. We follow a similar framework to those in Section 3.2 in the previous chapter.

Assumption 4.1. Assume that

$$\sup_i \max_{1 \leq j \leq M} E|y_{j,i} - E(y_{j,i})|^{2q}$$

is upper bounded by a positive constant c for some constant $q > 2$, where $y_{j,i}$ is the j^{th} component of \mathbf{y}_i and $E(y_{j,i})$ is its expected value.

Assumption 4.1 requires the tail probabilities of the variables of \mathbf{y}_i to decay fast enough such that we obtain a fast convergence rate for $\hat{\Gamma}_i$.

Assumption 4.2. Assume that the mixing coefficient defined in (3.21) satisfies the following

$$\sum_{l=1}^{\infty} \theta_l^{1-2/q} < \infty$$

for q that is given in Assumption 4.1.

Assumption 4.2 above indicates that θ_l goes to zero fast enough in order to retain a fast rate of convergence for $\hat{\Gamma}_i$.

The following theorem states the rate of convergence of $\hat{\mathbf{U}}$ to \mathbf{U} when the dimension M is fixed.

Theorem 4.1. Under assumptions 4.1 and 4.2, given that the dimension M is fixed, then there exists $\hat{\mathbf{U}} = (\hat{\mathbf{U}}_1, \dots, \hat{\mathbf{U}}_M)$ such that

$$\max_{1 \leq j \leq M} D(\mathcal{M}(\hat{\mathbf{U}}_j), \mathcal{M}(\mathbf{U}_j)) = O_p(1/\sqrt{W})$$

as $W \rightarrow \infty$.

In order to prove Theorem 4.1, we need the following lemma, which is based on Lemma 3.1, Lemma 3.2 and Lemma 3.3 from Chapter 3.

Lemma 4.1. *Let the assumptions 4.1 and 4.2 hold. Assume the dimension M is fixed, then as $W \rightarrow \infty$*

$$\|\hat{\mathbf{Q}} - \mathbf{Q}\|_2 = O_p(1/\sqrt{W}).$$

Proof of Lemma 4.1. Notice that

$$\|\hat{\mathbf{M}}\Gamma[\hat{\mathbf{M}}\Gamma]' - \mathbf{M}\Gamma[\mathbf{M}\Gamma]'\|_2 \leq \|\hat{\mathbf{M}}\Gamma - \mathbf{M}\Gamma\|_2^2 + 2\|\hat{\mathbf{M}}\Gamma - \mathbf{M}\Gamma\|_2\|\mathbf{M}\Gamma\|_2$$

and from Lemma 3.3, we know that

$$\|\hat{\mathbf{M}}\Gamma - \mathbf{M}\Gamma\|_2 = O_p(1/\sqrt{W}),$$

then it follows that

$$\|\hat{\mathbf{M}}\Gamma[\hat{\mathbf{M}}\Gamma]' - \mathbf{M}\Gamma[\mathbf{M}\Gamma]'\|_2 = O_p(1/\sqrt{W})$$

as $W \rightarrow \infty$.

Now, we are going to provide the proof of Theorem 4.1.

Proof of Theorem 4.1. Under the assumptions 4.1 and 4.2. Assume the dimension M is fixed. By Theorem 8.1.10 of Golub and Van Loan (1996) and Remark 1 of Lemma 1 in Chang et al. (2018), then the following holds

$$\max_{1 \leq j \leq M} D(\mathcal{M}(\hat{\mathbf{U}}_j), \mathcal{M}(\mathbf{U}_j)) = O_p(\|\hat{\mathbf{Q}} - \mathbf{Q}\|_2).$$

Then by Lemma 4.1, we have

$$\max_{1 \leq j \leq M} D(\mathcal{M}(\hat{\mathbf{U}}_j), \mathcal{M}(\mathbf{U}_j)) = O_p(1/\sqrt{W})$$

as $W \rightarrow \infty$.

4.3 Simulations and Real Data Examples

In the following examples, the ability of the QMDPCA method to reduce the dimension of multivariate time series will be investigated on both simulated and real data sets with various dimensions. Both stationary and non-stationary cases will

be considered, although much of our focus will be spent on the non-stationary case. The performance of QMDPCA will be evaluated by the percentage of explained variation, the MSE of reconstructed data and the moving cross-correlation plots of the retained QMDPCs. Our analyses will be performed using R software. The necessary functions to produce and assess the results of QMDPCA can be found under the following R packages: QMDPCA by Alshammri (2020f) and RCC_QMDPCA by Alshammri (2020h).

4.3.1 Simulations

In the following studies, the behaviour of QMDPCA with different combinations of window and lag sizes will be investigated on simulated data sets of various dimensions and sample sizes. Each simulation will be replicated 500 times. The R command `arma.sim` will be used to generate the data sets.

Example 4.3.1 In this example, a stationary series \mathbf{z}_t with three variables and 1000 samples is considered. This example consists of three parts. The first part compares the results of applying QMDPCA and DPCA with different sizes of l on \mathbf{z}_t . The second part studies the effect of the size of T on the results of QMDPCA. The third part focuses on optimizing the results of QMDPCA. The variables of \mathbf{z}_t are simulated as follows. Let a_t, b_t and c_t be independent standard normal white noises, which are the innovation terms of the following three models, respectively, then:

$$\begin{cases} z_{1,t} = u_t \\ z_{2,t} = v_t \\ z_{3,t} = w_t \end{cases} \quad (4.14)$$

where $(u_t, v_t, w_t)'$ satisfies

$$\begin{cases} u_t = -0.8u_{t-1} + a_t \\ v_t = 0.25v_{t-1} + b_t \\ w_t = 0.65w_{t-1} + c_t \end{cases} \quad (4.15)$$

A time series plot of one simulated series is shown in Figure 4.1, where the data exhibited a stationary behaviour. Thus, QMDPCA shall use a wide window of size $W = N$.

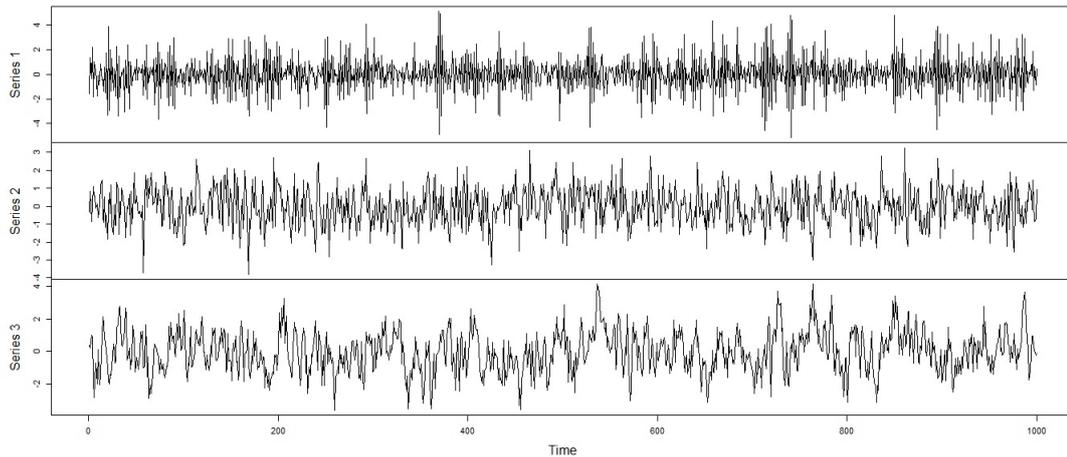


Figure 4.1: A time series plot of the randomly selected simulated series with three variables in Example 4.3.1.

The first part of this simulation study focuses on comparing the results of QMDPCA with those of DPCA of Ku et al. (1995)) using different number of lags. Based on 500 replicas, Table 4.1 is a comparison between the results of QMDPCA and DPCA using one and five lags, where two components are retained. In general, the mean percentages of explained variation by two QMDPCs of QMDPCA were higher than those explained by two DPCs of DPCA. For example, with $l = 1$, two QMDPCs explained 91.38% of the variation in the data compared with 70.57% when using two DPCs. Additionally, with $l = 5$, the mean percentage of explained variation was slightly reduced for QMDPCA and significantly reduced for DPCA. For example, the mean percentage of explained variation decreased from 91.38% to 86.05% for QMDPCA compared with 70.57% to 50.30% for DPCA. Additionally, the mean MSE of reconstructed data had similar values for both QMDPCA and DPCA. Both methods produced relatively small reconstruction errors. For example, when reconstructing the data with two components, both methods produced a mean MSE of 0.55 when $l = 1$ and 0.92 when $l = 5$. Notice that the standard deviations of both

the explained variation percentage and the MSE of reconstructed data for all four cases did not exceed 0.03, which indicates similar results to be obtained across the 500 replicas for the considered four cases.

Method Results	QMDPCA ($l=1$)	DPCA ($l=1$)
Contribution % Mean (Standard Deviation)	91.38 % (0.01)	70.57 % (0.02)
MSE Mean (Standard Deviation)	0.55 (0.02)	0.55 (0.02)
Method Results	QMDPCA ($l=5$)	DPCA ($l=5$)
Contribution % Mean (Standard Deviation)	86.05 % (0.02)	50.30 % (0.02)
MSE Mean (Standard Deviation)	0.92 (0.03)	0.92 (0.03)

Table 4.1: A comparison between QMDPCA and DPCA with different sizes of l applied to the simulated stationary series in Example 4.3.1.

The correlations between the retained components were revealed by plotting the mean of the absolute values of the autocorrelation function (i.e. acf function) as the series is stationary. As shown in figures 4.2 and 4.3, uncorrelated components were obtained for all four cases. Also, the standard deviations of the absolute values of the acf function are reported in Table 4.2, where the values ranged from 0 to 0.08, which is very close to zero. Therefore, similar acf plots were obtained across the 500 replicas.

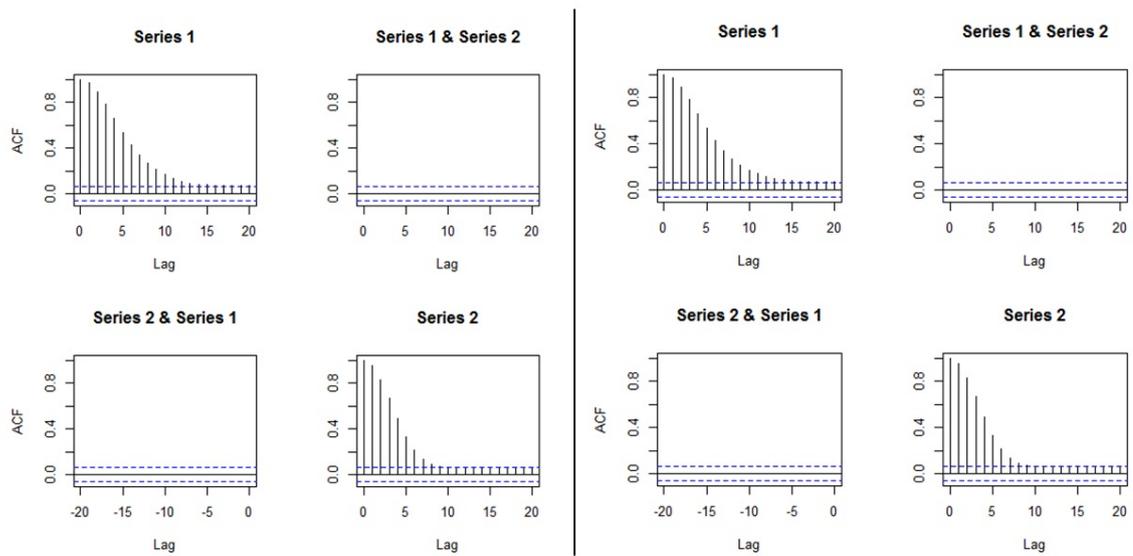


Figure 4.2: Plots of the mean of the absolute value of the moving cross-correlations between two components in Example 4.3.1 [Left: QMDPCA ($l = 1$), Right: DPCA($l = 1$)].

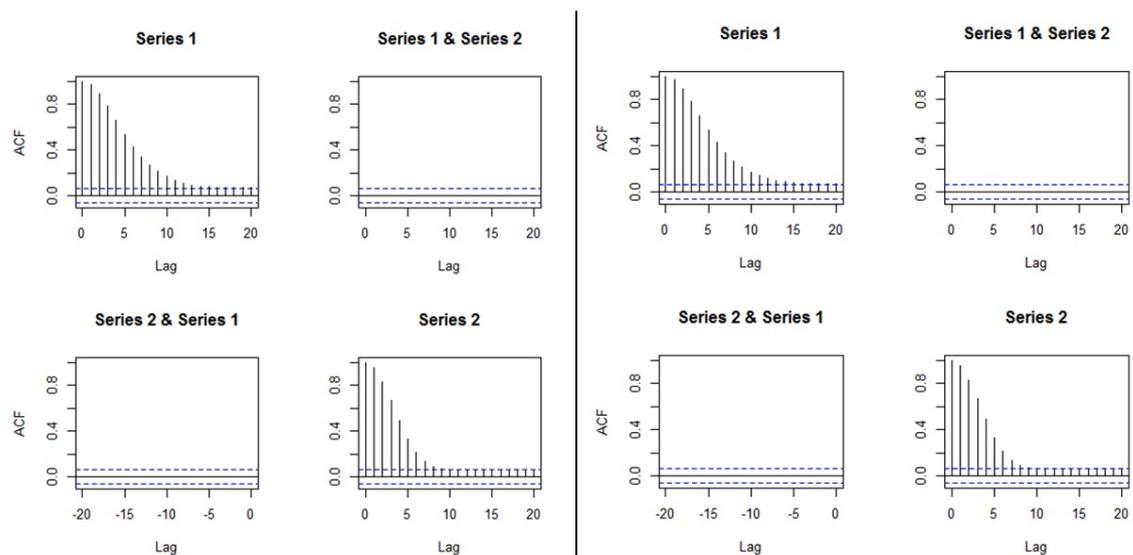


Figure 4.3: Plots of the mean of the absolute value of the moving cross-correlations between two components in Example 4.3.1 [Left: QMDPCA ($l = 5$), Right: DPCA($l = 5$)].

		QMDPCA($l=1$)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.01	0.03	0.04	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.04	0.04
QMDPC1 vs QMDPC2	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
QMDPC2 vs QMDPC1	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
QMDPC2 vs QMDPC2	0.00	0.01	0.03	0.04	0.05	0.05	0.05	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.03	0.03	0.03	0.03	0.03	0.03	0.03
		DPCA($l=1$)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
DPC1 vs DPC1	0.00	0.01	0.03	0.04	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.04	0.04	0.04
DPC1 vs DPC2	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DPC2 vs DPC1	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DPC2 vs DPC2	0.00	0.01	0.03	0.04	0.05	0.05	0.05	0.04	0.04	0.04	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04
		QMDPCA($l=5$)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.03	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.07	0.07	0.07	0.07	0.06	0.06	0.05	0.05	0.05	0.06	0.06
QMDPC1 vs QMDPC2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
QMDPC2 vs QMDPC1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
QMDPC2 vs QMDPC2	0.00	0.00	0.02	0.03	0.05	0.06	0.07	0.07	0.07	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.04	0.05	0.05	0.04	0.04
		DPCA($l=5$)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
DPC1 vs DPC1	0.00	0.00	0.01	0.02	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06
DPC1 vs DPC2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DPC2 vs DPC1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DPC2 vs DPC2	0.00	0.00	0.02	0.03	0.05	0.06	0.07	0.07	0.07	0.06	0.05	0.05	0.05	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05

Table 4.2: Standard deviations of the absolute value of the acf function between two components with different sizes of l in Example 4.3.1.

Therefore, based on the above results, the dimension of the simulated stationary series with three variables could be reduced by QMDPCA with $l = 1$. Alternatively, with slightly less accuracy, QMDPCA with $l = 5$ can be used, where more dynamic information could be added to the analysis. The case where the optimal number of lagged series is used will be considered latter in this example.

The second part of this example studies the effect of the sample size T on the results of QMDPCA. In particular, we used the three models introduced in the first part of this example to generate series with sample sizes $T = \{200, 400, 600, 800\}$. Based on 500 replicas, Table 4.3 is a comparison between the results of applying QMDPCA with $W = N$ (i.e. $W = T - l$) and $l = 1$ to the generated series, where two QMDPCs are retained. It can be seen that the results in Table 4.3 are close to what we observed when QMDPCA with $l = 1$ was applied to the simulated series with $T = 1000$. In specific, the mean percentages of explained variation were above 91% with a standard deviations less than 0.04 for all cases. Also, the mean MSE of reconstructed data had values around 0.55 with standard deviations less than 0.05 in all cases.

Method	QMDPCA ($W=T-l, l=1$)			
Sample size	T=200	T=400	T=600	T=800
Contribution %	91.07 %	91.20 %	91.33 %	91.50 %
Mean (Standard Deviation)	(0.03)	(0.02)	(0.02)	(0.01)
MSE	0.54	0.55	0.55	0.55
Mean (Standard Deviation)	(0.04)	(0.03)	(0.02)	(0.02)

Table 4.3: Comparison between the results of QMDPCA with different sizes of T in Example 4.3.1.

Moreover, the mean of the absolute value of the acf function between two QMDPCs of series of different sample sizes are plotted in figures 4.4 and 4.5, where the two QMDPCs were uncorrelated in all cases. We noticed that the standard deviations of the absolute value of the acf function between the considered two QMDPCs decreased as we increased the sample size; See Table 4.4. For example, the standard deviations ranged between 0 to 0.13 when $T = 200$ comparing with 0 to 0.07 when

$T = 800$, where the latter is similar to what we observed when $T = 1000$ earlier.

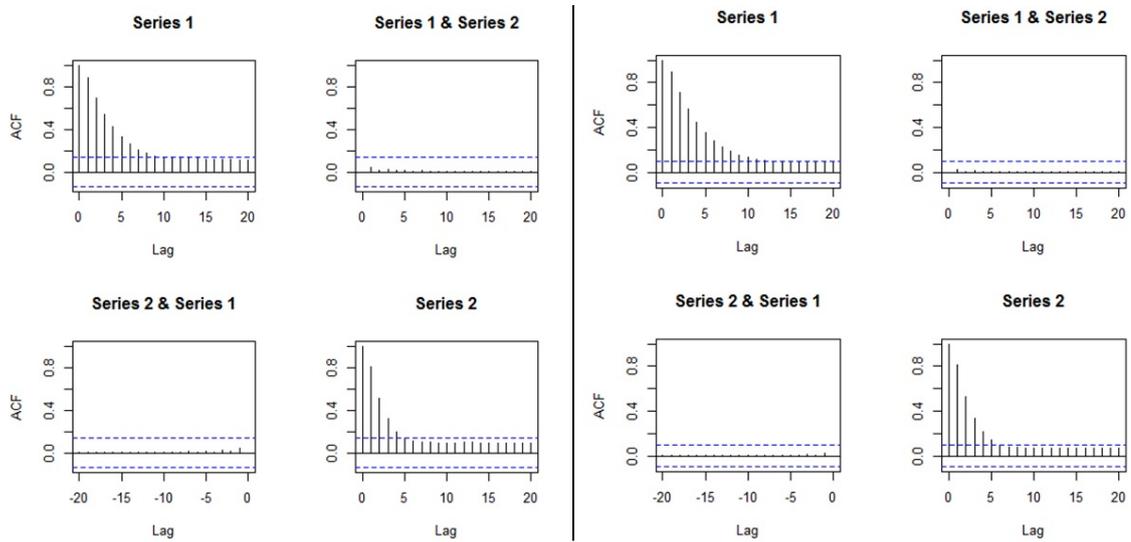


Figure 4.4: Plots of the mean of the absolute value of the acf function between two QMDPCs with different sizes of T in Example 4.3.1 [Left: $T = 200$, Right: $T = 400$].

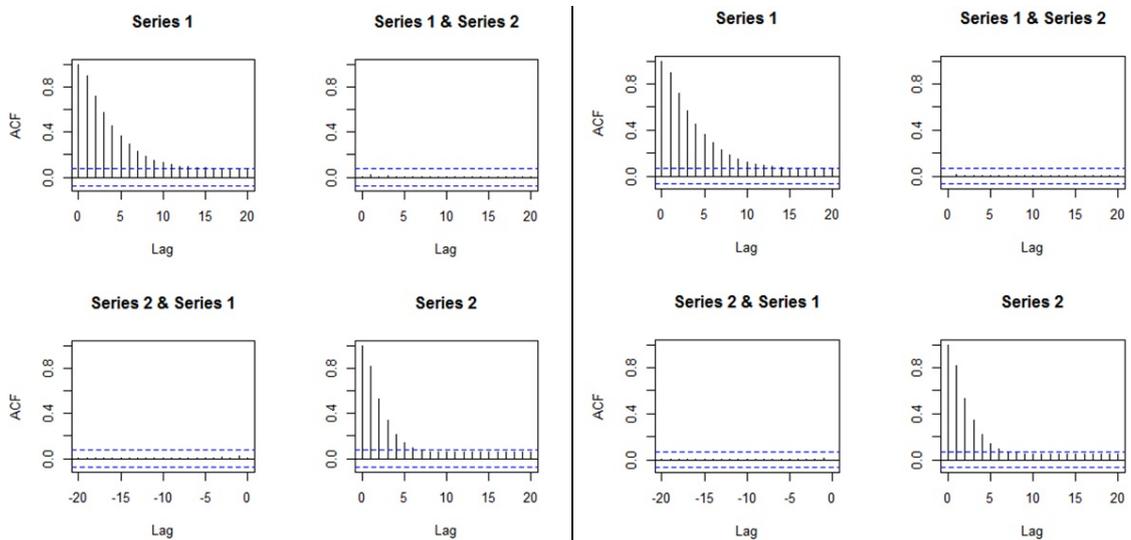


Figure 4.5: Plots of the mean of the absolute value of the acf function between two QMDPCs with different sizes of T in Example 4.3.1 [Left: $T = 600$, Right: $T = 800$].

		T=200																				
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lag	0	0.00	0.05	0.06	0.10	0.11	0.13	0.13	0.13	0.12	0.12	0.11	0.11	0.10	0.10	0.09	0.09	0.09	0.09	0.09	0.09	0.09
	QMDPC1 vs QMDPC1	0.00	0.08	0.01	0.04	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	QMDPC1 vs QMDPC2	0.00	0.08	0.01	0.04	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	QMDPC2 vs QMDPC1	0.00	0.05	0.08	0.10	0.11	0.09	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
	QMDPC2 vs QMDPC2	0.00	0.05	0.08	0.10	0.11	0.09	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
		T=400																				
lag	0	0.00	0.02	0.05	0.07	0.08	0.09	0.10	0.10	0.10	0.09	0.09	0.08	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
	QMDPC1 vs QMDPC1	0.00	0.04	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	QMDPC1 vs QMDPC2	0.00	0.04	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	QMDPC2 vs QMDPC1	0.00	0.02	0.06	0.07	0.08	0.08	0.07	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
	QMDPC2 vs QMDPC2	0.00	0.02	0.06	0.07	0.08	0.08	0.07	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
		T=600																				
lag	0	0.00	0.02	0.04	0.05	0.06	0.07	0.08	0.08	0.08	0.08	0.08	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
	QMDPC1 vs QMDPC1	0.00	0.03	0.01	0.01	0.01	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	QMDPC1 vs QMDPC2	0.00	0.03	0.01	0.01	0.01	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	QMDPC2 vs QMDPC1	0.00	0.02	0.05	0.06	0.07	0.07	0.06	0.05	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
	QMDPC2 vs QMDPC2	0.00	0.02	0.05	0.06	0.07	0.07	0.06	0.05	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
		T=800																				
lag	0	0.00	0.01	0.03	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05
	QMDPC1 vs QMDPC1	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	QMDPC1 vs QMDPC2	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	QMDPC2 vs QMDPC1	0.00	0.01	0.01	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	QMDPC2 vs QMDPC2	0.00	0.01	0.04	0.05	0.06	0.06	0.05	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04

Table 4.4: Standard deviations of the absolute value of the acf function between two QMDPCs with different sizes of T in Example 4.3.1.

Therefore, increasing the sample size of the data improves the results of QMDPCA to reduce the dimension of the simulated stationary series. We notice that the performance of QMDPCA was reasonable on series with sizes as small as $T = 200$.

The last part of this study includes only one randomly selected replica from the above simulations with 1000 samples. In this part, we focus on optimizing the results of QMDPCA. Recall that the optimal window size for a stationary series is $W = N$. The optimal number of lags is suggested to be $l = 1$ based on Figure 4.6, where one static and two dynamic relations were found. Therefore, QMDPCA with $W = N$ and $l = 1$ is suggested to reduce the dimension of the simulated series. Figure 4.7 shows the eigenvalues' plot and RCC plot after applying QMDPCA, where the RCC criterion suggested two optimal QMDPCs to retain. These two components explained 93.01% of the variation in the data and produced an MSE of 0.56 when used to reconstruct the data. The sample acf plots in Figure 4.8 indicated uncorrelated QMDPCs. In conclusion, QMDPCA reduced the dimension of the simulated stationary series from three to two stationary components that explained a high percentage of the variation of the data and produced a low reconstruction error.

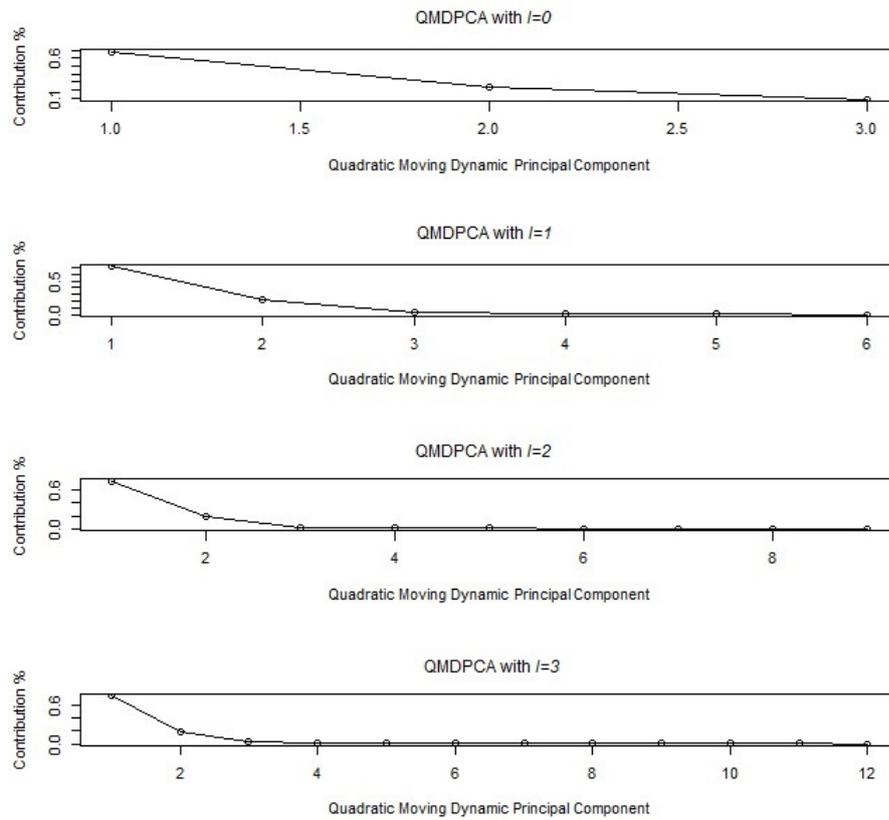


Figure 4.6: Eigenvalues' plots after applying QMDPCA with $W = N$ and different sizes of l to the simulated series with three variables in Example 4.3.1.

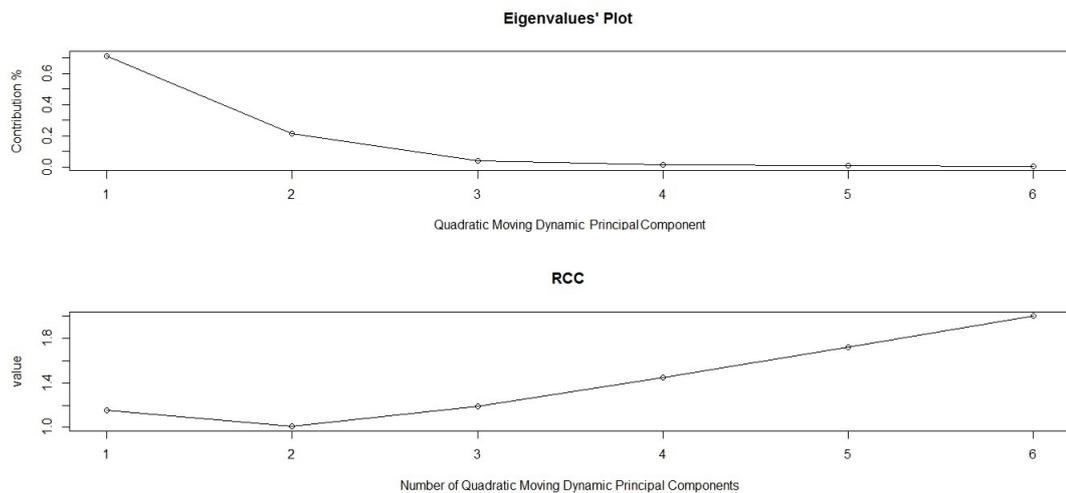


Figure 4.7: Top: An eigenvalues' plot of the simulated series with three variables in Example 4.3.1 after applying QMDPCA with $W = N$ and $l = 1$. Bottom: RCC plot.

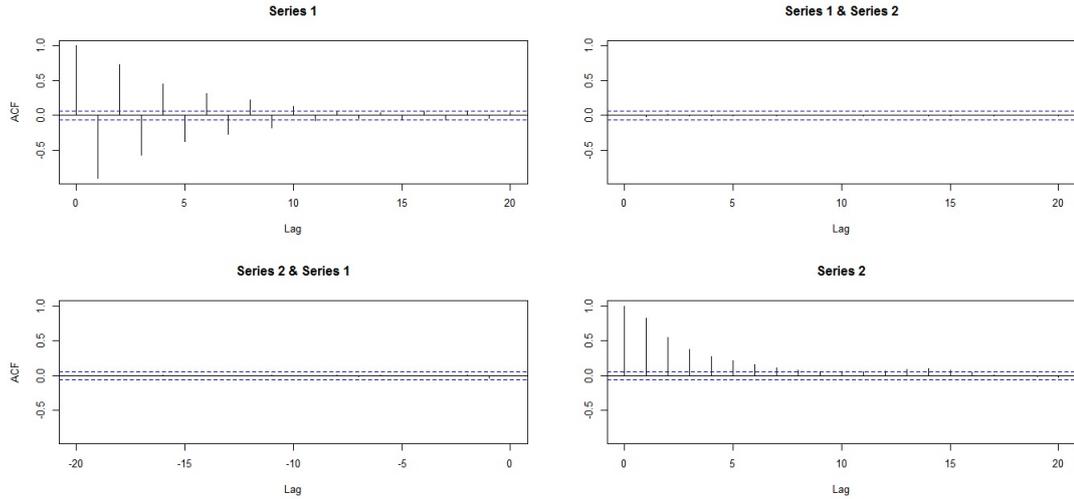


Figure 4.8: Sample moving cross-correlation plots of the retained two QMDPCs of the simulated series in Example 4.3.1.

Example 4.3.2 In this example, we test the ability of QMDPCA on a non-stationary data set \mathbf{z}_t that consists of five variables and 1000 samples. This example consists of four parts. The first part compares the results of applying QMDPCA and MDPCA with different sizes of l on \mathbf{z}_t . The second part investigates the results of QMDPCA when using different combinations of W and l . The third part investigates the effect of the size of T on the results of QMDPCA. The fourth part focuses on optimizing the results of QMDPCA. The series \mathbf{z}_t is generated using five different models as described below.

Let a_t, b_t, c_t, d_t and e_t be independent standard normal white noises, which are the innovation terms of the following five models, respectively, then:

$$\left\{ \begin{array}{l} z_{1,t} = u_t \\ z_{2,t} = v_t \\ z_{3,t} = w_t \\ z_{4,t} = x_t \\ z_{5,t} = q_t \end{array} \right. \quad (4.16)$$

where $(u_t, v_t, w_t, f_t, q_t)'$ satisfies

$$\begin{cases} \nabla u_t = 0.55 \nabla u_{t-1} + a_t - 0.8a_{t-1} \\ \nabla v_t = -0.65 \nabla v_{t-1} + b_t + 0.45b_{t-1} \\ \nabla w_t = 0.45 \nabla w_{t-1} + c_t + 1.6c_{t-1} \\ \nabla x_t = -0.8 \nabla x_{t-1} + d_t - 0.9d_{t-1} \\ \nabla q_t = 0.85 \nabla q_{t-1} + e_t - 2.2e_{t-1} \end{cases} \quad (4.17)$$

A time series plot of the simulated data is shown in Figure 4.9, where the variables exhibited apparent non-stationary behaviours.

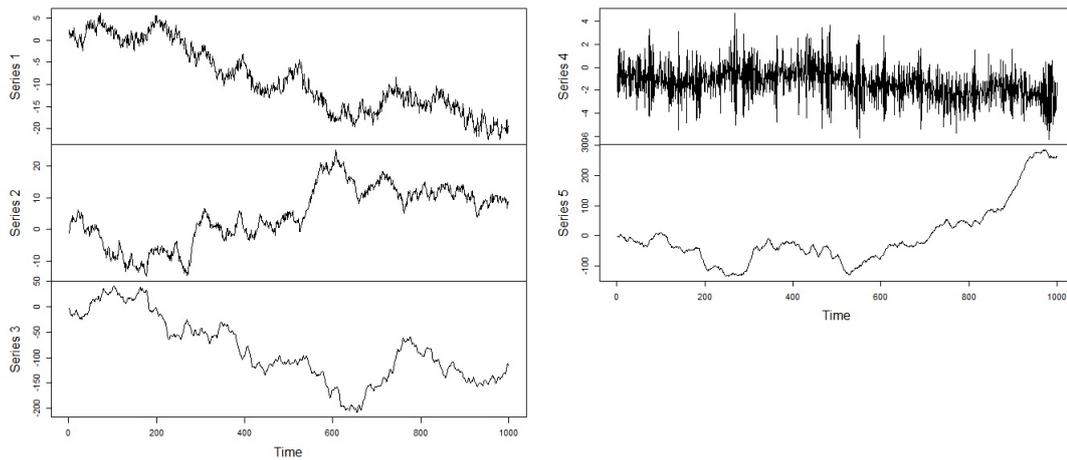


Figure 4.9: A time series plot of the simulated non-stationary series with five variables in Example 4.3.2.

In the first part of this simulation study, we compare the results of QMDPCA with those of MDPCA introduced in Chapter 3. Both methods use a window of size $W = 101$ in order to obtain a fair comparison. Based on 500 replicas, Table 4.5 is a comparison between the results of QMDPCA and MDPCA with $l = 1$ and $l = 5$, where two components are retained. It can be seen that the mean percentages of explained variation by two QMDPCs were slightly higher than those explained by two MDPCs. For example, with $l = 1$, two QMDPCs explained 99.98% of the variation in the data compared with 98.22% when two MDPCs were used. When $l = 5$ was

used, the mean percentages of explained variation decreased slightly to 99.95% for QMDPCA, and to 96.44% for MDPCA. Steady results were obtained from the 500 replicas as both methods had a standard deviation of explained variation around zero. In terms of the MSE of reconstructed data, both QMDPCA and MDPCA produced similar values. In particular, both methods produced a mean error of 85.55 with a standard deviation of 79.51 when $l = 1$, and 93.36 with a standard deviation of 84.13 when $l = 5$. The large standard deviations of the MSE of reconstructed data indicated different reconstruction errors for different replicas.

Method Results	QMDPCA ($l=1$)	MDPCA ($l=1$)
Contribution % Mean (Standard Deviation)	99.98 % (0.00)	98.22 % (0.00)
MSE Mean (Standard Deviation)	85.55 (79.51)	85.55 (79.51)
Method Results	QMDPCA ($l=5$)	MDPCA ($l=5$)
Contribution % Mean (Standard Deviation)	99.95 % (0.00)	96.44 % (0.01)
MSE Mean (Standard Deviation)	93.36 (84.13)	93.36 (84.13)

Table 4.5: A comparison between QMDPCA and MDPCA with $W = 101$ and different sizes of l in Example 4.3.2 where two components are retained.

Additionally, the mean of the absolute value of the moving cross-correlations with $W = 101$ between the retained two components of both methods are plotted in figures 4.10 and 4.11. Notice that the retained two components were uncorrelated for both methods in all cases. The standard deviations of the absolute value of the moving cross-correlations with $W = 101$ between the two components are reported in Table 4.6, where their values did not exceed 0.11 in all cases. Hence, similar results were obtained for both methods in terms of the correlation between the retained components.

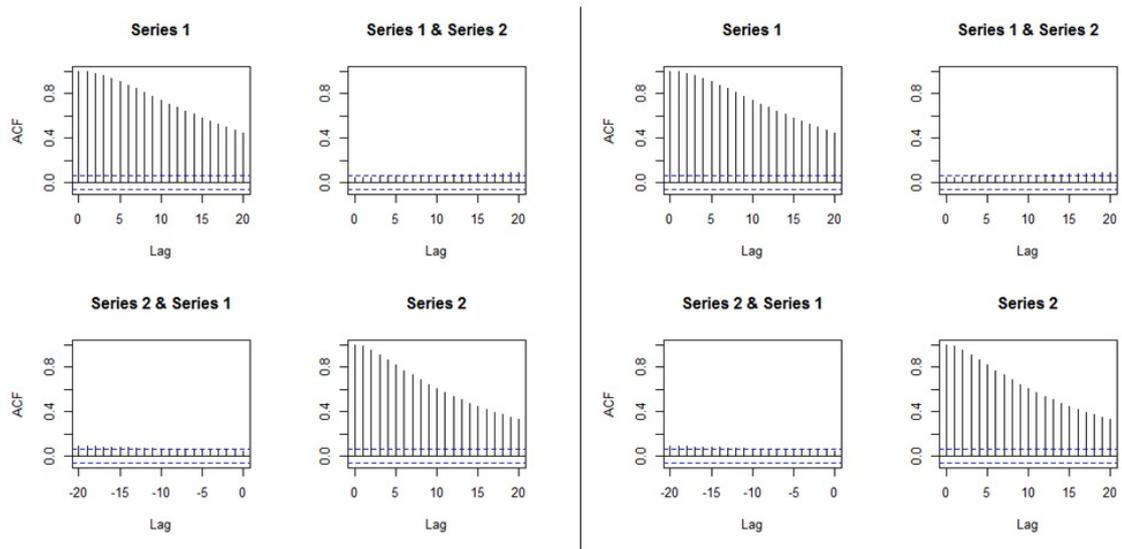


Figure 4.10: Plots of the mean of the absolute value of the moving cross-correlations between two components in Example 4.3.2 [Left: QMDPCA ($W = 101, l = 1$), Right: MDPKA ($W = 101, l = 1$)].

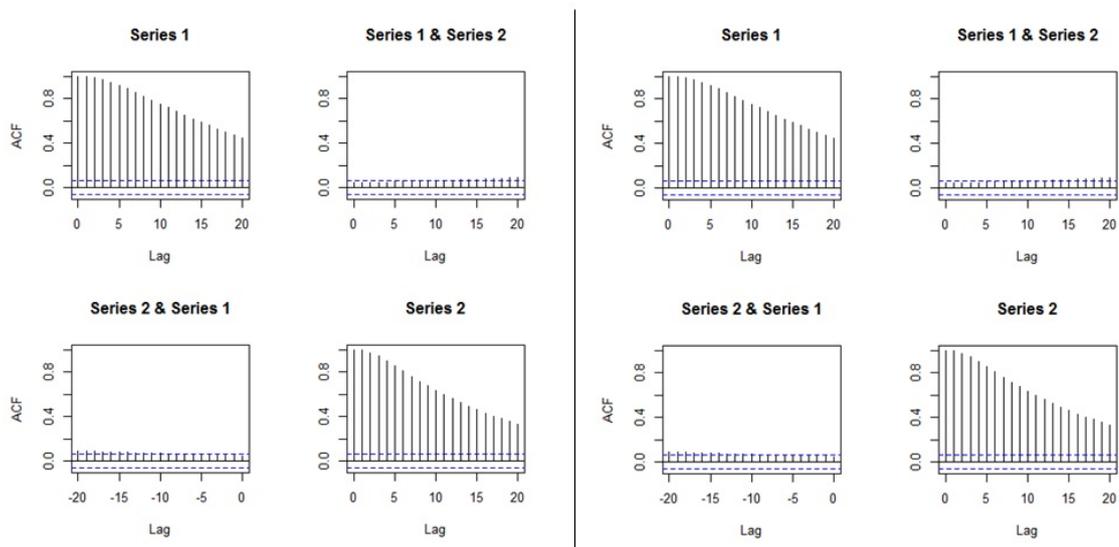


Figure 4.11: Plots of the mean of the absolute value of the moving cross-correlations between two components in Example 4.3.2 [Left: QMDPCA ($W = 101, l = 5$), Right: MDPKA ($W = 101, l = 5$)].

		QMDPCA($l=1$)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.01	0.02	0.02	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.08	0.09	0.09
QMDPC1 vs QMDPC2	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
QMDPC2 vs QMDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.04	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09	0.10	0.10
		MDPCA($l=1$)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.01	0.01	0.02	0.02	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.08	0.09	0.09
MDPC1 vs MDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
MDPC2 vs MDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
MDPC2 vs MDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.04	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09	0.10	0.10
		QMDPCA($l=5$)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.00	0.01	0.01	0.01	0.02	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09
QMDPC1 vs QMDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
QMDPC2 vs QMDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.05	0.05	0.06	0.07	0.07	0.08	0.08	0.09	0.09	0.10	0.10	0.10	0.11	0.11
		MDPCA($l=5$)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MDPC1 vs MDPC1	0.00	0.00	0.00	0.01	0.01	0.01	0.02	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09
MDPC1 vs MDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
MDPC2 vs MDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
MDPC2 vs MDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.05	0.05	0.06	0.07	0.07	0.08	0.08	0.09	0.09	0.10	0.10	0.10	0.11	0.11

Table 4.6: Standard deviations of the absolute value of the moving cross-correlation between two components with $W = 101$ and different sizes of l in Example 4.3.2.

Therefore, both QMDPCA and MDPCA produced similar results when applied to reduce the dimension of the simulated non-stationary series with five variables in terms of reconstruction error and the dynamic relationship between the retained components. We noticed that the QMDPCA showed a higher percentage of explained variation.

The second part of this simulation study focuses on comparing the results of QMDPCA using different sizes of W and l . Based on 500 replicas, Table 4.7 is a comparison between the results of applying QMDPCA with different combinations of W and l , where two QMDPCs are retained. The mean percentages of explained variation by two QMDPCs were about 99.9% for all cases and did not differ much by changing the size of W or l . For example, a mean percentage of 99.98% was obtained when using QMDPCA with $W = 101$ and $l = 1$, compared with 99.95% when using $W = 101$ and $l = 5$. Also, a mean percentage of 99.98% was obtained when using QMDPCA with $W = 301$ for both cases where $l = 1$ and $l = 5$. Also, we noticed that these percentages did not change across the 500 replicas, as they have zero standard deviations when rounded to two decimal points. In terms of the mean of MSE of reconstructed data, its values varied from 73.17 to 87.35 for the considered combinations of W and l , where it had its lowest when using $W = 301$ and $l = 1$, and it had its highest when using $W = 101$ and $l = 5$. We observed large values for the standard deviations of the MSE of reconstructed data for all cases, which is an indication of a non-steady reconstruction errors across the replicas. For example, a mean MSE of 75.77 was obtained when using $W = 301$ and $l = 5$ with a standard deviation of 62.02.

Window Size (W)	$W=101$	$W=201$	$W=301$
Results	($l=1$)	($l=1$)	($l=1$)
Contribution %	99.98 %	99.98 %	99.98 %
Mean (Standard Deviation)	(0.00)	(0.00)	(0.00)
MSE	86.09	79.59	73.17
Mean (Standard Deviation)	(72.47)	(75.76)	(75.24)
Window Size (W)	$W=101$	$W=201$	$W=301$
Results	($l=5$)	($l=5$)	($l=5$)
Contribution %	99.95 %	99.97 %	99.98 %
Mean (Standard Deviation)	(0.00)	(0.00)	(0.00)
MSE	87.35	81.78	75.77
Mean (Standard Deviation)	(71.03)	(76.13)	(62.02)

Table 4.7: A comparison between the results of QMDPCA with different combinations of W and l in Example 4.3.2.

For the mean of the absolute value of the moving cross-correlations with $W = 101$ between the retained QMDPCs, we noticed that the components were uncorrelated for the cases where $W = 101$ was used, as plotted in figures 4.12 - 4.14. The correlations became larger and significant as we increased the window size to 201 and 301. The standard deviations of the absolute value of the moving cross-correlations with $W = 101$ between the two components are reported in Table 4.8, where their values were reasonable and did not exceed 0.1 for the considered six cases.

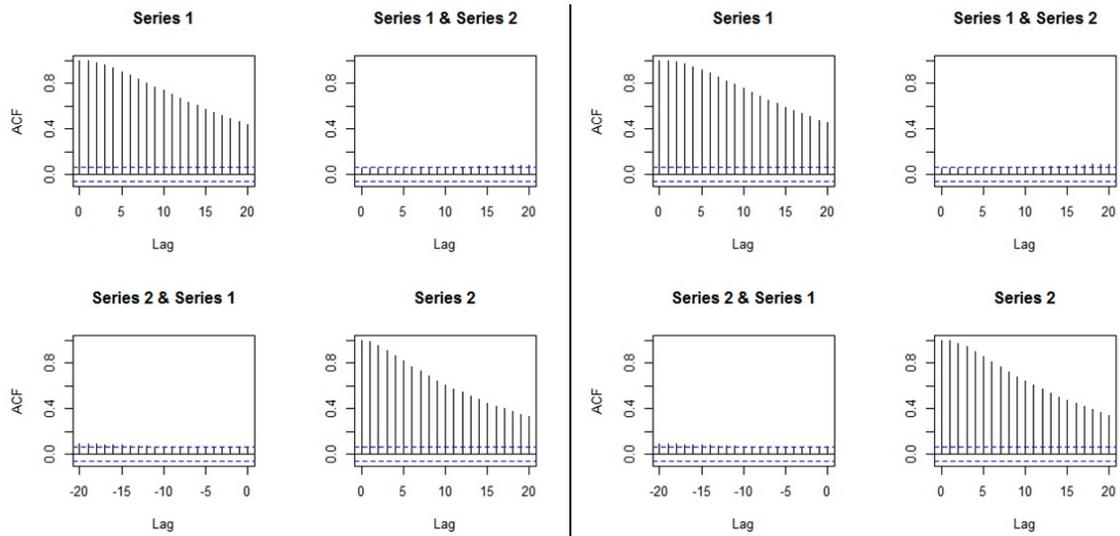


Figure 4.12: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.2 [Left: QMDPCA ($W = 101, l = 1$), Right: QMDPCA($W = 101, l = 5$)].

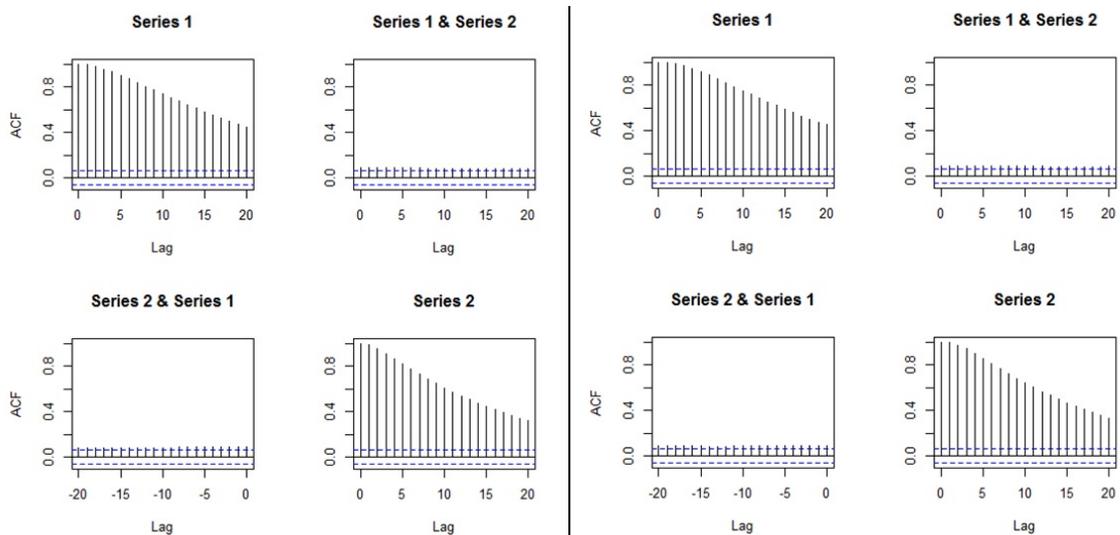


Figure 4.13: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.2 [Left: QMDPCA ($W = 201, l = 1$), Right: QMDPCA($W = 201, l = 5$)].

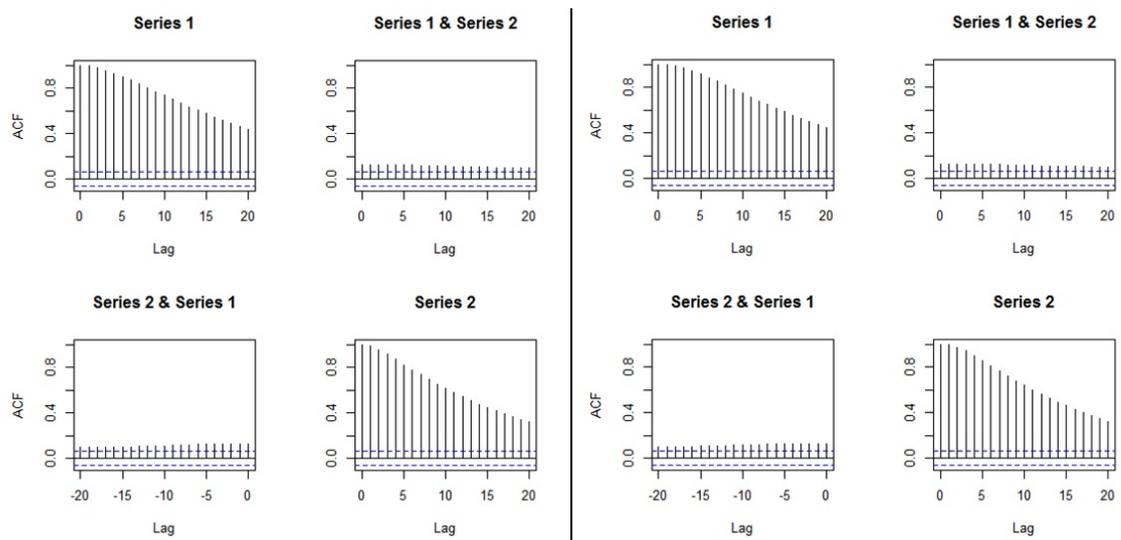


Figure 4.14: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.2 [Left: QMDPCA ($W = 301, l = 1$), Right: QMDPCA ($W = 301, l = 5$)].

		QMDPCA(W=101, l=1)																			
Lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.10
QMDPC1 vs QMDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07
QMDPC2 vs QMDPC1	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09	0.10	0.10
		QMDPCA(W=201, l=1)																			
Lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09
QMDPC1 vs QMDPC2	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07
QMDPC2 vs QMDPC1	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09	0.10	0.10
		QMDPCA(W=301, l=1)																			
Lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09
QMDPC1 vs QMDPC2	0.10	0.10	0.10	0.10	0.09	0.09	0.09	0.09	0.09	0.09	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.07	0.07	0.07
QMDPC2 vs QMDPC1	0.10	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.07	0.07	0.07
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.10
		QMDPCA(W=101, l=5)																			
Lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.07	0.07	0.08	0.08	0.09	0.09	0.10
QMDPC1 vs QMDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07
QMDPC2 vs QMDPC1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.10
		QMDPCA(W=201, l=5)																			
Lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.07	0.07	0.08	0.08	0.09	0.09	0.09
QMDPC1 vs QMDPC2	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
QMDPC2 vs QMDPC1	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.10
		QMDPCA(W=301, l=5)																			
Lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.09	0.09
QMDPC1 vs QMDPC2	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09
QMDPC2 vs QMDPC1	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.11

Table 4.8: Standard deviations of the absolute value of the moving cross-correlations between two components with different combinations of W and l in Example 4.3.2.

Therefore, changing the window size had a significant effect on the results of QMDPCA when applied to the non-stationary series in this example. Additionally, similar results were obtained when including one or five lagged series to the analysis, where slightly better results were obtained when using one lag in this example.

The third part of this study focuses on the effect of the sample size T on the results of QMDPCA. In the previous parts of this study, the five models introduced at the beginning of this example were used to simulate the series \mathbf{z}_t with 1000 samples. In this part, we simulate \mathbf{z}_t using the same five models, but with sample sizes $T = \{200, 400, 600, 800\}$, then apply QMDPCA with $W = 101$ and $l = 1$ and compare the results. Based on 500 replicas, Table 4.9 is a comparison between QMDPCA's results for the above cases, where two QMDPCs are retained in all cases. We observed a mean percentage of explained variation of 99.97% with zero standard deviation for all different sample size. The mean and the standard deviation of the MSE of reconstruction data increased as we increased the sample size. For example, a mean MSE of 16.25 was obtained when $T = 200$ compared with 68.72 when $T = 800$. This can be justified as less reconstruction error is expected for smaller number of observation, and the error increases as more data is required to be reconstructed. The large standard deviations of the reconstruction error indicate the value of the reconstruction error varies from replica to the other. A smaller reconstruction error could be obtained by normalizing the data. Consequently, the standard deviation of the reconstruction error would also be reduced.

By investigating the plots of the mean of the absolute value of the moving cross-correlation between two QMDPCs with different sample sizes in figures 4.15 and 4.16, we noticed that significant correlation between the two components appeared after the first ± 10 lags when $T = 200$. As we increased the sample size to 400, 600 and 800, the correlations became non-significant and acted similar to what we observed when $T = 1000$ earlier. Furthermore, the standard deviations of the absolute value of the moving cross-correlation between two QMDPCs are reported in Table 4.10, where smaller values were obtained when using $T \geq 400$. For example,

the standard deviations ranged from 0 to 0.24 when $T = 200$, compared with 0 to 0.16 when $T = 400$, and 0 to 0.11 when $T = 800$.

Method	QMDPCA (W=101, l=1)			
Sample size	T=200	T=400	T=600	T=800
Contribution %	99.97 %	99.97 %	99.97 %	99.97 %
Mean (Standard Deviation)	(0.00)	(0.00)	(0.00)	(0.00)
MSE	16.25	32.40	46.62	68.72
Mean (Standard Deviation)	(14.91)	(32.87)	(42.00)	(64.17)

Table 4.9: A comparison between the results of QMDPCA with different sizes of T in Example 4.3.2.

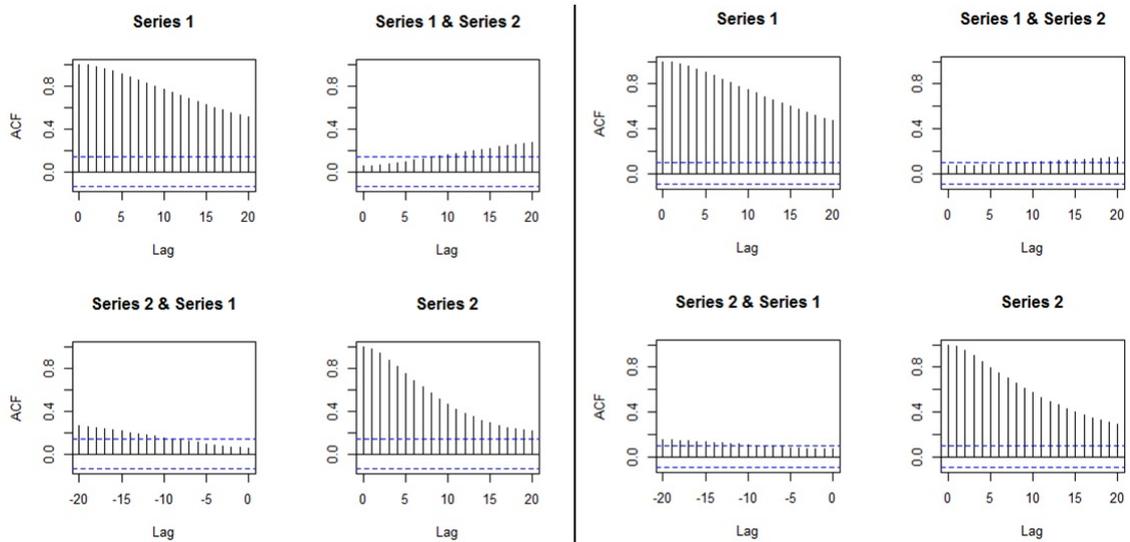


Figure 4.15: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.2 [Left: $T = 200$, Right: $T = 400$].

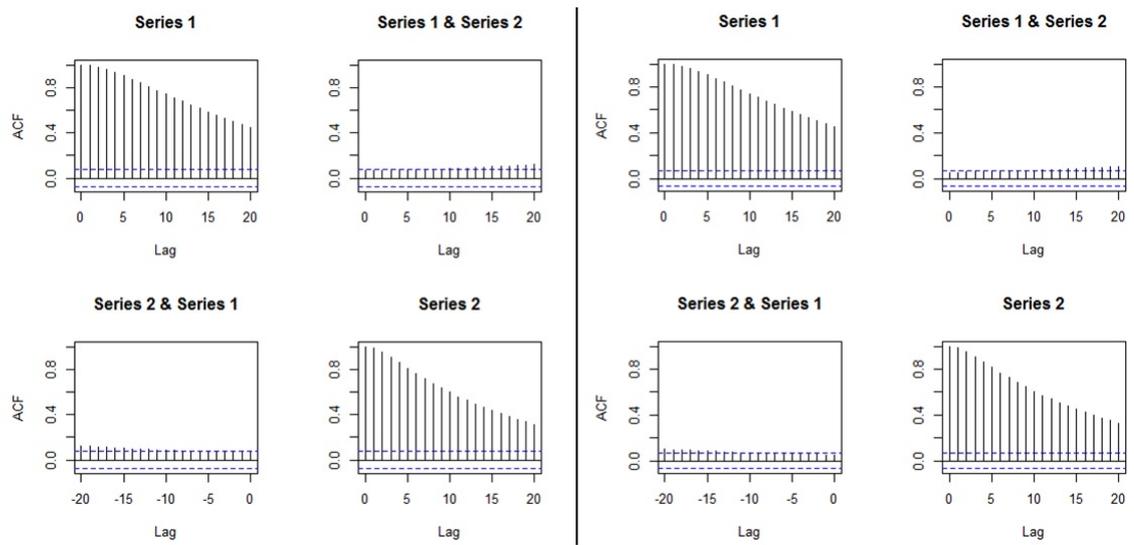


Figure 4.16: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.2 [Left: $T = 600$, Right: $T = 800$].

		T=200																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.02	0.03	0.05	0.06	0.08	0.09	0.11	0.13	0.14	0.16	0.17	0.18	0.19	0.21	0.21	0.22	0.23	0.23	0.24
QMDPC1 vs QMDPC2	0.05	0.05	0.05	0.06	0.06	0.07	0.08	0.09	0.10	0.11	0.12	0.12	0.13	0.14	0.15	0.15	0.16	0.16	0.17	0.17	0.18	0.18
QMDPC2 vs QMDPC1	0.05	0.05	0.05	0.06	0.07	0.07	0.08	0.09	0.10	0.11	0.12	0.13	0.13	0.14	0.15	0.15	0.16	0.16	0.17	0.17	0.18	0.18
QMDPC2 vs QMDPC2	0.00	0.01	0.03	0.06	0.09	0.12	0.14	0.16	0.17	0.19	0.20	0.20	0.20	0.20	0.20	0.19	0.19	0.18	0.18	0.17	0.17	0.17
		T=400																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.02	0.03	0.03	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.12	0.12	0.13	0.14	0.15	0.15	0.16
QMDPC1 vs QMDPC2	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.10	0.11	0.11
QMDPC2 vs QMDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.10	0.11	0.11	0.11	0.11
QMDPC2 vs QMDPC2	0.00	0.01	0.02	0.03	0.05	0.07	0.08	0.09	0.10	0.11	0.12	0.13	0.14	0.14	0.15	0.15	0.15	0.16	0.16	0.16	0.16	0.16
		T=600																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.05	0.06	0.06	0.07	0.08	0.08	0.09	0.09	0.10	0.11	0.11	0.12	0.12
QMDPC1 vs QMDPC2	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09
QMDPC2 vs QMDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.03	0.04	0.05	0.05	0.06	0.07	0.08	0.09	0.10	0.10	0.11	0.11	0.12	0.12	0.12	0.13	0.13	0.13	0.13
		T=800																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.09	0.10	0.10	0.11
QMDPC1 vs QMDPC2	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07
QMDPC2 vs QMDPC1	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.09	0.09	0.09	0.09	0.09	0.10	0.10	0.10	0.11

Table 4.10: Standard deviations of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.2.

Therefore, by applying QMDPCA with $W = 101$ and $l = 1$ on the simulated series in this example with different sample sizes, the two QMDPCs explained more than 99% of the variation in the data. However, the value of reconstruction error differed across simulations. Also, the two QMDPCs showed some correlation when using small sample size and became uncorrelated when increasing the sample size. In particular, a reasonable performance was obtained when $T \geq 400$.

In the last part of this example we optimize the results of QMDPCA based on one randomly selected replica with 1000 samples. Considering the non-stationarity of the simulated series, as shown in Figure 4.9, then a window of size $W = 101$ can be used. The optimal number of lags is suggested to be $l = 1$, where three static and two dynamic relations were found according to the eigenvalues' plots in Figure 4.17. Thus, we apply QMDPCA with $W = 101$ and $l = 1$ to reduce the dimension of the simulated series in this part. Figure 4.18 shows the eigenvalues' plot and RCC plot after QMDPCA was applied to the simulated series. The RCC value was 0.94 for the first QMDPC, compared with 0.41 for the first two QMDPCs, 0.60 for the first three QMDPCs, and then continued to increase afterward. Therefore, two QMDPCs to be retained were suggested to be optimal by the RCC criterion. The retained two components were able to explain almost all variation in the data, where the percentage of explained variation was 99.98%. Also, the retained two components were able to reconstruct the data with an MSE of 24.07, which relatively small value considering the size of the data. Additionally, the sample moving cross-correlation plots between the two retained QMDPCs indicated uncorrelated components, as shown in Figure 4.19.

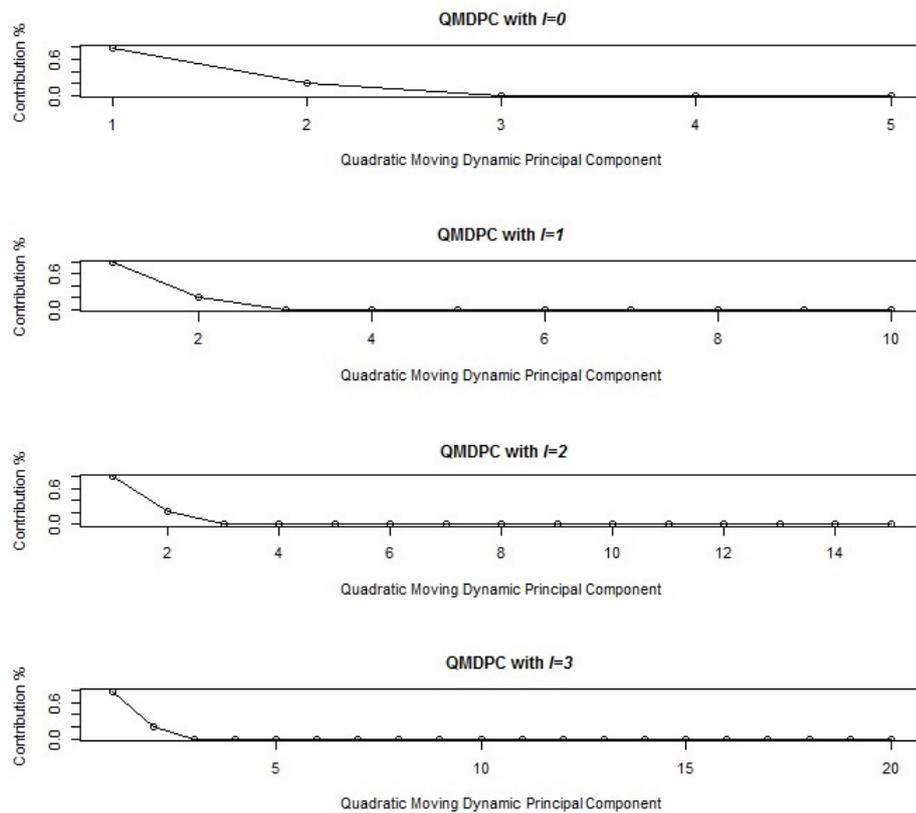


Figure 4.17: Eigenvalues' plots after applying QMDPCA with $W = 101$ and different sizes of l to the simulated series with five variables in Example 4.3.2.

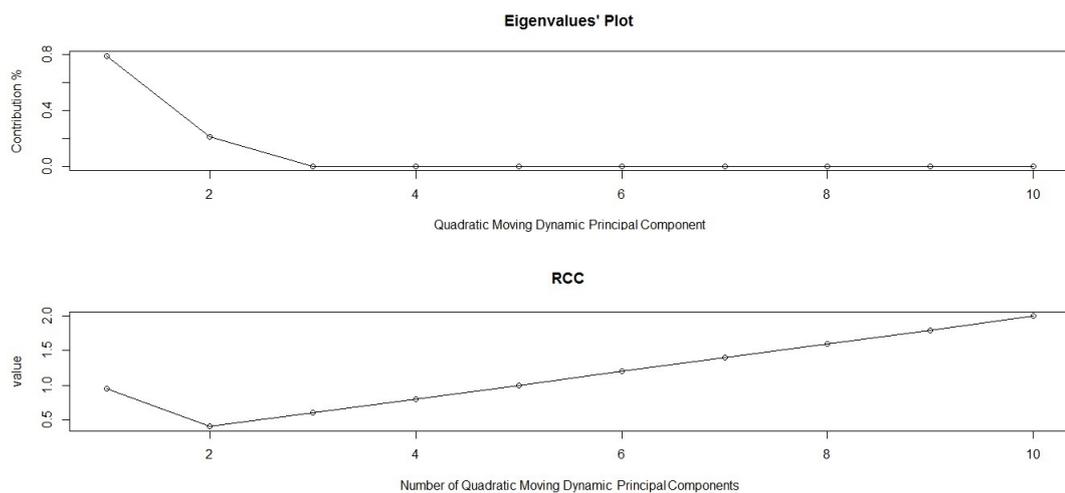


Figure 4.18: Top: An eigenvalues' plot of the simulated series with five variables in Example 4.3.2 after applying QMDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot.

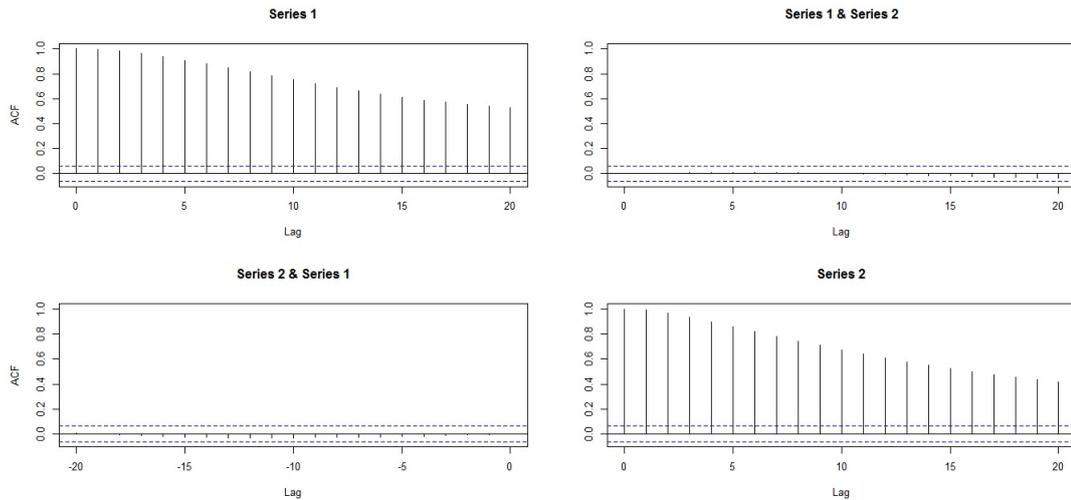


Figure 4.19: Sample moving cross-correlation plots of the retained two QMDPCs of the simulated series in Example 4.3.2.

Therefore, we conclude that the QMDPCA with $W = 101$ and $l = 1$ was able to reduce the dimension of the simulated non-stationary series from five to two. The retained two components explained almost all variation in the data and could be used to reconstruct the original data with a small error.

Example 4.3.3 In the following study, a non-stationary series \mathbf{z}_t with ten variables and 1500 samples. This study consists of three parts. The first part studies the results of QMDPCA when using different combinations of W and l . The second part investigates the effect of the size of T on the results of QMDPCA. The third part focuses on optimizing the results of QMDPCA. The series \mathbf{z}_t is simulated using five different models, such that each model generates two correlated variables. Let a_t, b_t, c_t, d_t and e_t be independent standard normal white noises, which are the innovation terms of the following five models, respectively, then:

$$\left\{ \begin{array}{ll} z_{j,t} = u_{t+j-1}, & j = 1 \text{ and } 2 \\ z_{j,t} = v_{t+j-3}, & j = 3 \text{ and } 4 \\ z_{j,t} = w_{t+j-5}, & j = 5 \text{ and } 6 \\ z_{j,t} = x_{t+j-7}, & j = 7 \text{ and } 8 \\ z_{j,t} = q_{t+j-9}, & j = 9 \text{ and } 10 \end{array} \right. \quad (4.18)$$

where $(u_t, v_t, w_t, f_t, q_t)'$ satisfies

$$\begin{cases} \nabla u_t = -0.6 \nabla u_{t-1} + a_t - 1.8a_{t-1} \\ \nabla v_t = 0.8 \nabla v_{t-1} + b_t - 0.7b_{t-1} \\ \nabla w_t = 0.5 \nabla w_{t-1} + c_t + 0.9c_{t-1} \\ \nabla x_t = 0.7 \nabla x_{t-1} + d_t \\ \nabla q_t = -0.9 \nabla q_{t-1} + e_t \end{cases} \quad (4.19)$$

Figure 4.20 is a time series plot of a simulated series. It can be seen that the variables exhibited non-stationary behaviours.

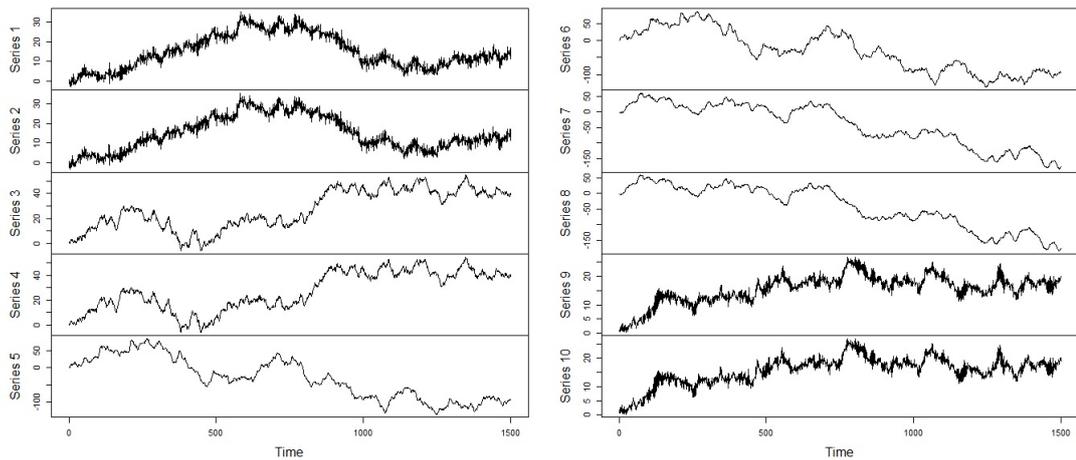


Figure 4.20: A time series plot of the simulated non-stationary series with ten variables in Example 4.3.3.

First, we compare the results of QMDPCA with different window and lag sizes on the simulated series. Based on 500 replicas, Table 4.11 is a comparison between the results of applying QMDPCA with different combinations of W and l , where two QMDPCs are considered. The mean percentages of explained variation by two QMDPCs were about 98.5% and did not differ much by changing the size of W and l . For example, a mean percentage of 98.57% was obtained when using QMDPCA with $W = 101$ and $l = 1$, compared with 98.44% when using $W = 101$ and $l = 5$. Also, a mean percentage of 98.60% was obtained when using QMDPCA with $W = 301$

and $l = 1$, compared with 98.59% when using $W = 301$ and $l = 5$. Also, we noticed that these percentages were similar for all 500 replicas as they had small standard deviations of 0.01 in all cases. In terms of the mean of MSE of reconstructed data, we noticed that its value decreased as we increased the window size. In particular, its values ranged from 325.57 to 374.66 for the considered combinations of W and l , where it had its lowest when using $W = 301$ and $l = 5$, and it had its highest when using $W = 101$ and $l = 1$. We observed large values for the standard deviations of MSE of reconstructed data for all cases, which indicates non-steady reconstruction errors across the replicas. For example, a mean MSE of 337.23 was obtained when using $W = 201$ and $l = 1$ with a standard deviation of 313.96.

Window Size (W)	W=101 ($l=1$)	W=201 ($l=1$)	W=301 ($l=1$)
Results			
Contribution % Mean (Standard Deviation)	98.57 % (0.01)	98.55 % (0.01)	98.60 % (0.01)
MSE Mean (Standard Deviation)	374.66 (349.63)	337.23 (313.96)	334.47 (331.91)
Window Size (W)	W=101 ($l=5$)	W=201 ($l=5$)	W=301 ($l=5$)
Results			
Contribution % Mean (Standard Deviation)	98.44 % (0.01)	98.53 % (0.01)	98.59 (0.01)
MSE Mean (Standard Deviation)	369.04 (337.30)	345.57 (315.86)	325.57 (319.38)

Table 4.11: A comparison between the results of QMDPCA with different combinations of W and l in Example 4.3.3.

Also, the mean of the absolute value of the moving cross-correlations with $W = 101$ between the retained QMDPCs are plotted in figures 4.21 - 4.23. The plots indicate uncorrelated QMDPCs for the cases where $W = 101$ was used. The correlations became significant as we increased the W to 201 and 301. The standard deviations of the absolute value of the moving cross-correlations with $W = 101$ between the retained components had small values as reported in Table 4.12, where their values ranged from 0 to 0.1 in all of the considered six cases.

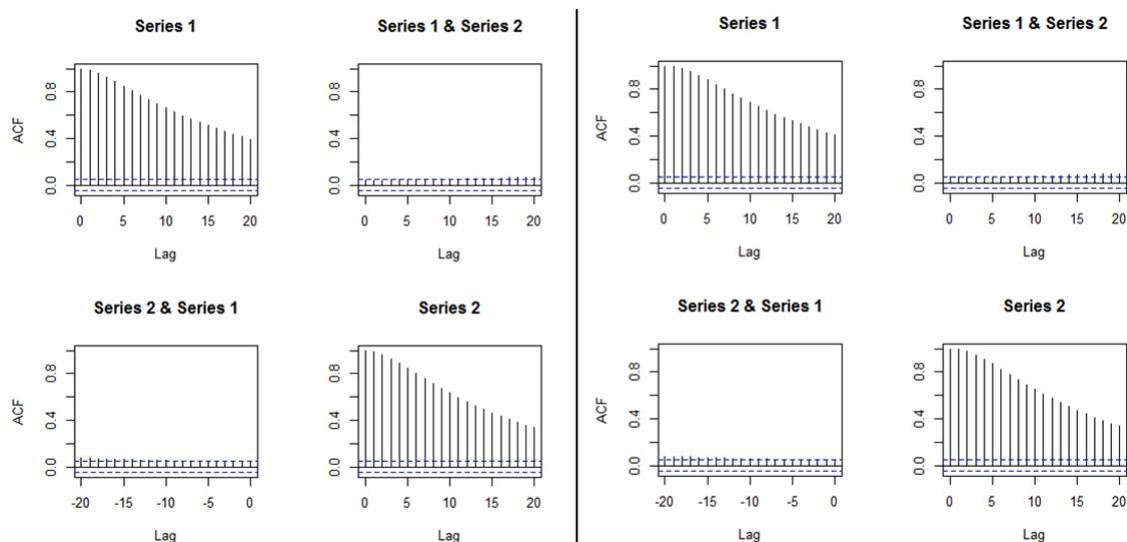


Figure 4.21: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.3 [Left: QMDPCA ($W = 101$, $l = 1$), Right: QMDPCA($W = 101$, $l = 5$)].

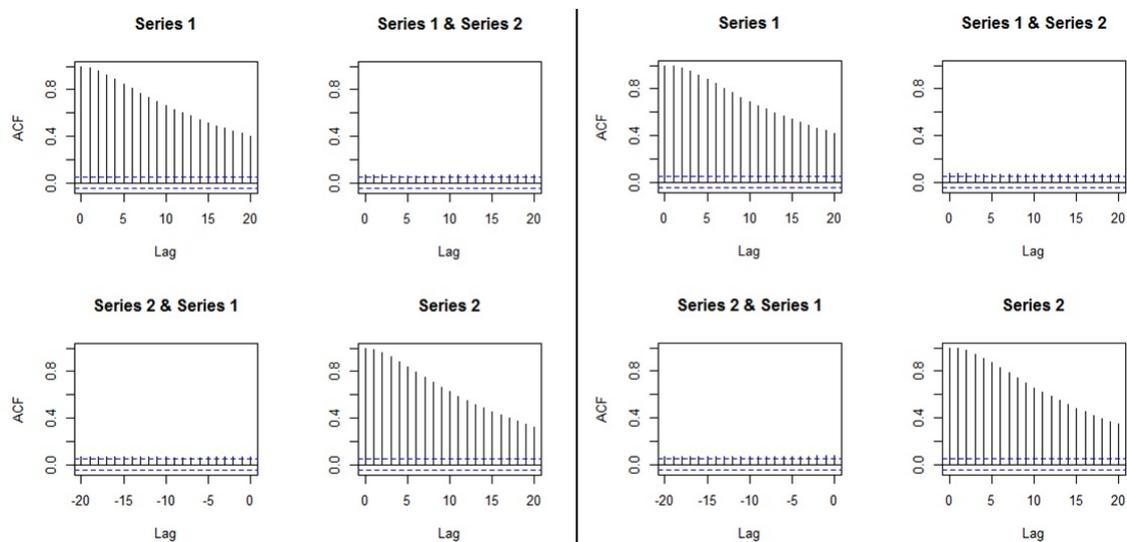


Figure 4.22: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.3 [Left: QMDPCA ($W = 201$, $l = 1$), Right: QMDPCA($W = 201$, $l = 5$)].

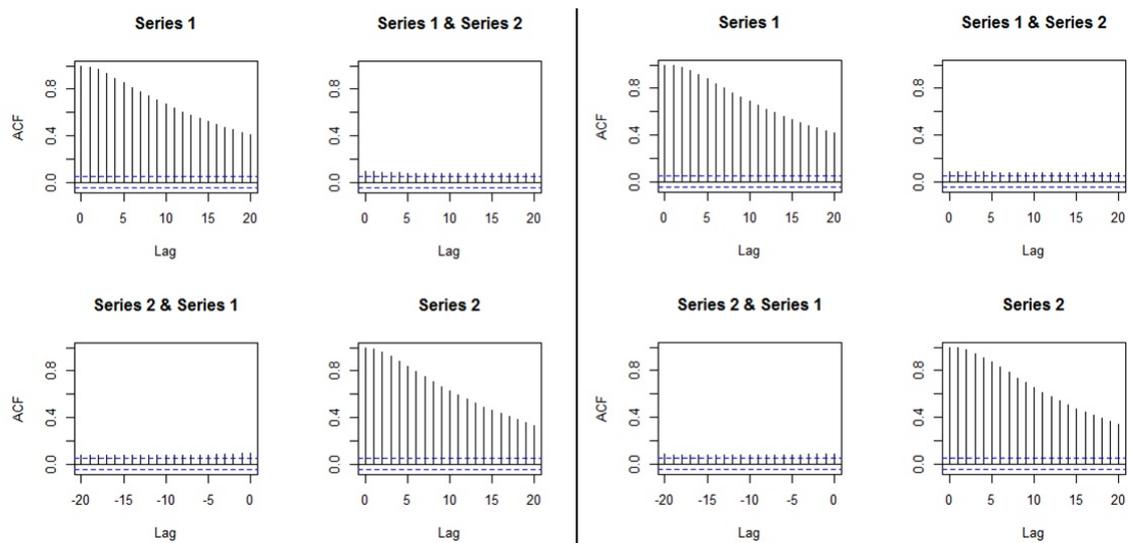


Figure 4.23: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.3 [Left: QMDPCA ($W = 301$, $l = 1$), Right: QMDPCA($W = 301$, $l = 5$)].

		QMLDPCA(W=101, l=1)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMLDPC1 vs QMLDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08
QMLDPC1 vs QMLDPC2	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06
QMLDPC2 vs QMLDPC1	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06
QMLDPC2 vs QMLDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.08
		QMLDPCA(W=201, l=1)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMLDPC1 vs QMLDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08
QMLDPC1 vs QMLDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
QMLDPC2 vs QMLDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
QMLDPC2 vs QMLDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.08	0.08
		QMLDPCA(W=301, l=1)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMLDPC1 vs QMLDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07
QMLDPC1 vs QMLDPC2	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
QMLDPC2 vs QMLDPC1	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
QMLDPC2 vs QMLDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.08
		QMLDPCA(W=101, l=5)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMLDPC1 vs QMLDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.08
QMLDPC1 vs QMLDPC2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05
QMLDPC2 vs QMLDPC1	0.04	0.04	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05
QMLDPC2 vs QMLDPC2	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.08
		QMLDPCA(W=201, l=5)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMLDPC1 vs QMLDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08
QMLDPC1 vs QMLDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
QMLDPC2 vs QMLDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
QMLDPC2 vs QMLDPC2	0.00	0.00	0.00	0.01	0.02	0.02	0.02	0.03	0.03	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08
		QMLDPCA(W=301, l=5)																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMLDPC1 vs QMLDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08
QMLDPC1 vs QMLDPC2	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
QMLDPC2 vs QMLDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
QMLDPC2 vs QMLDPC2	0.00	0.00	0.00	0.01	0.02	0.02	0.02	0.03	0.03	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.08

Table 4.12: Standard deviations of the absolute value of the moving cross-correlations between two components with different combinations of W and l in Example 4.3.3.

Therefore, based on the above results, the dimension of the simulated non-

stationary series in this example can be reduced by using QMDPCA with $W = 101$ and either $l = 1$ or $l = 5$. Using larger window sizes caused the QMDPCA to produce correlated components.

Second, we study the effect of changing the sample size of the simulated series with ten variables on the results of QMDPCA. Based on 500 replicas, Table 4.13 is a comparison between the results of QMDPCA with $W = 101$ and $l = 5$ applied to the simulated series with sample sizes $T = \{200, 400, 600, 800\}$, where two QMDPCs are retained. Notice that slight changes in the values of the mean and standard deviation of the percentage of explained variation. In particular, the mean percentage of explained variation decreased slightly as we increased T . The standard deviation of the percentage of explained variation increased slightly as we increased T . For example, a mean percentage of 99.03% with 0.02 standard deviation was obtained when $T = 200$, compared with 98.52% and 0.01 when $T = 800$. Additionally, the mean and the standard deviation of the MSE of reconstruction data increased as we increased T . For example, a mean MSE of 41.56 with a standard deviation of 42.21 was obtained when $T = 200$, compared with 188.22 and 187.19 when $T = 800$. As the data in this example is not normalized, the values of the reconstruction error and its standard deviations could be reduced further by normalizing the data.

Method	QMDPCA (W=101, l=5)			
Sample size	T=200	T=400	T=600	T=800
Contribution % Mean (Standard Deviation)	99.03 % (0.02)	98.65 % (0.02)	98.65 % (0.01)	98.52 % (0.01)
MSE Mean (Standard Deviation)	41.56 (42.21)	81.37 (74.55)	132.23 (120.28)	188.22 (187.19)

Table 4.13: A comparison between the results of QMDPCA with different sizes of T in Example 4.3.3.

Furthermore, the plots of the mean of the absolute value of the moving cross-correlation between the retained QMDPCs with different sample sizes are provided in figures 4.24 and 4.25. It can be seen that some significant correlation between the retained two components appeared after the first ± 10 lags when $T = 200$. These

correlations decreased as we increased T to 400. The retained QMDPCs became uncorrelated when increasing T to 600 and 800. Also, the standard deviations of the absolute value of the moving cross-correlation between the retained QMDPCs are reported in Table 4.14, where smaller values were obtained when using $T \geq 400$, which indicates steady results for different replicas. For example, the standard deviations ranged from 0 to 0.24 when $T = 200$, compared with 0 to 0.16 when $T = 400$, and 0 to 0.12 when $T = 800$.

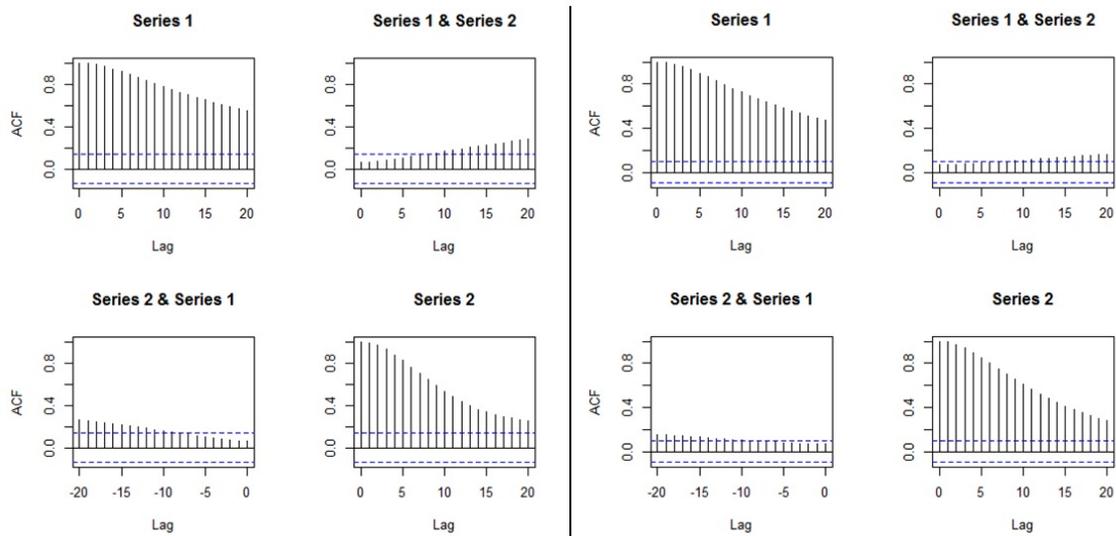


Figure 4.24: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.3 [Left: $T = 200$, Right: $T = 400$].

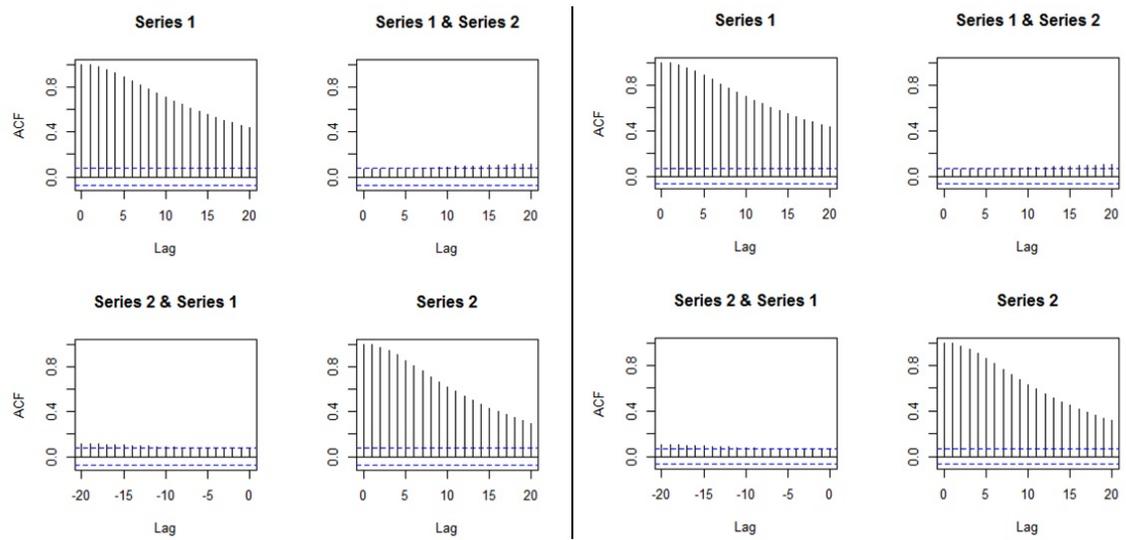


Figure 4.25: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.3 [Left: $T = 600$, Right: $T = 800$].

		T=200																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.02	0.04	0.05	0.07	0.09	0.10	0.12	0.13	0.15	0.16	0.17	0.18	0.20	0.21	0.21	0.21	0.22	0.23	0.24
QMDPC1 vs QMDPC2	0.06	0.06	0.06	0.07	0.07	0.08	0.09	0.09	0.10	0.11	0.11	0.12	0.13	0.14	0.15	0.15	0.16	0.16	0.16	0.17	0.18	0.18
QMDPC2 vs QMDPC1	0.06	0.06	0.06	0.06	0.07	0.08	0.08	0.09	0.10	0.11	0.12	0.13	0.14	0.14	0.15	0.15	0.16	0.16	0.17	0.17	0.18	0.18
QMDPC2 vs QMDPC2	0.00	0.00	0.02	0.03	0.05	0.08	0.10	0.13	0.15	0.17	0.19	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.19	0.19	0.19	0.18
		T=400																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.02	0.03	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.11	0.12	0.13	0.13	0.13	0.14	0.14	0.15
QMDPC1 vs QMDPC2	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.10	0.11	0.11	0.11
QMDPC2 vs QMDPC1	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.09	0.09	0.10	0.10	0.10	0.11	0.11	0.11	0.12
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.02	0.03	0.05	0.06	0.08	0.09	0.10	0.11	0.12	0.13	0.14	0.15	0.16	0.16	0.16	0.16	0.16	0.16	0.16
		T=600																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.08	0.08	0.09	0.09	0.10	0.10	0.11	0.11	0.11	0.12	0.12
QMDPC1 vs QMDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.09
QMDPC2 vs QMDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.09
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.10	0.10	0.11	0.11	0.12	0.12	0.13	0.13	0.13	0.13
		T=800																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.04	0.04	0.05	0.06	0.06	0.07	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.10	0.10
QMDPC1 vs QMDPC2	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.08
QMDPC2 vs QMDPC1	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.08
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.10	0.11	0.11	0.12	0.12	0.13	0.13	0.13	0.13

Table 4.14: Standard deviations of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.3.

Therefore, by applying QMDPCA with $W = 101$ and $l = 5$ to the simulated series in this example with different sample sizes, we were able to obtain high performance when using large sample size (i.e. $T \geq 600$). QMDPCA showed a reasonable performance when applied to series with sample of sizes as small as $T = 400$.

In the last part of this study, we optimize the results of QMDPCA by considering only one randomly selected replica of the above simulations with 1500 samples. A small window of size $W = 101$ is used considering the non-stationarity of the simulated series; See Figure 4.20. Also, eight static and two dynamic relations were observed by investigating the eigenvalues' plots in Figure 4.26. Hence, $l = 1$ is suggested to be optimal. Therefore, QMDPCA with $W = 101$ and $l = 1$ can be used to reduce the dimension of the simulated series. After applying QMDPCA to the data, Figure 4.27 shows the eigenvalues' plot and RCC plot. Two QMDPCs were suggested to be optimal with an RCC value of 0.31. The retained two QMDPCs explained almost all variation in the data with a percentage of 99.5%, and produced an MSE of 273.21 when they used to reconstruct the data, which is a reasonable value considering the size of the data. The sample moving cross-correlation plots between the two retained QMDPCs indicated uncorrelated components, as shown in Figure 4.28.

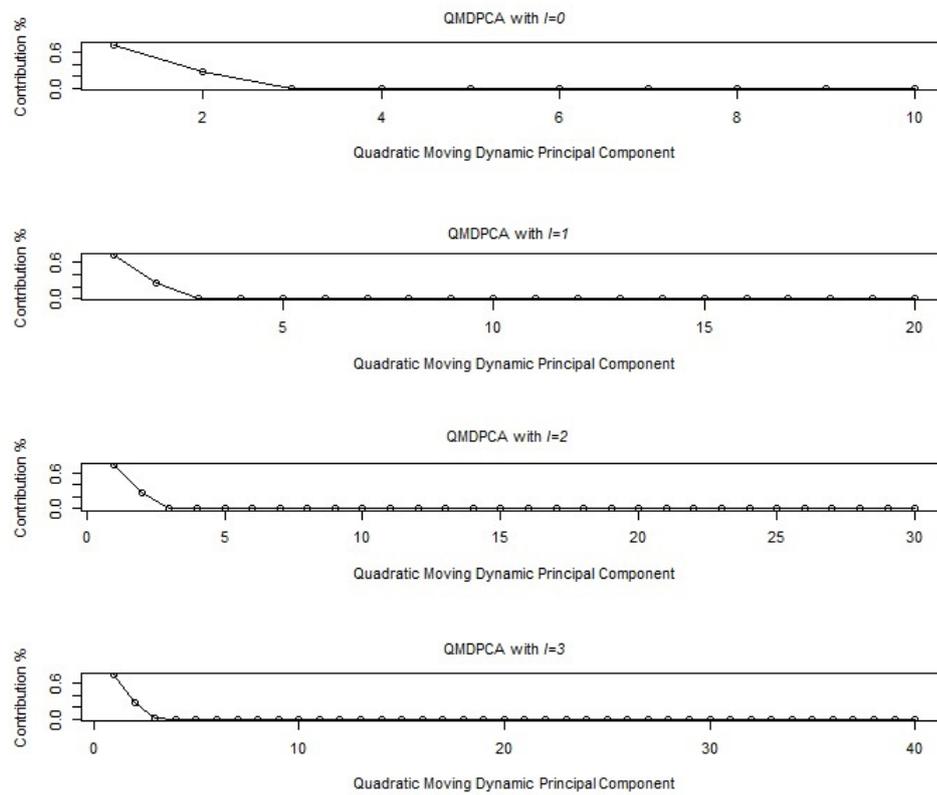


Figure 4.26: Eigenvalues' plots after applying QMDPCA with $W = 101$ and different sizes of l to the simulated series with ten variables in Example 4.3.3.

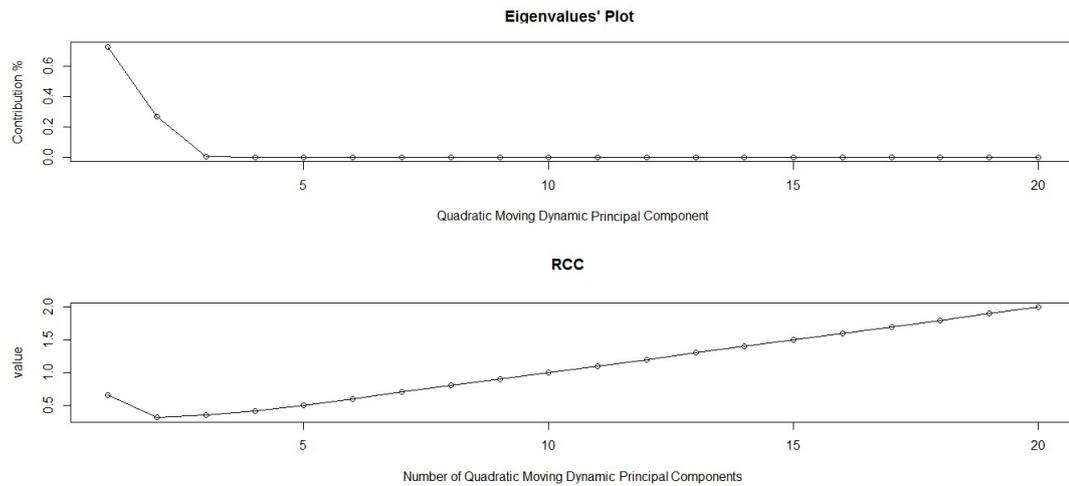


Figure 4.27: Top: An eigenvalues' plot of the simulated series with ten variables in Example 4.3.3 after applying QMDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot.

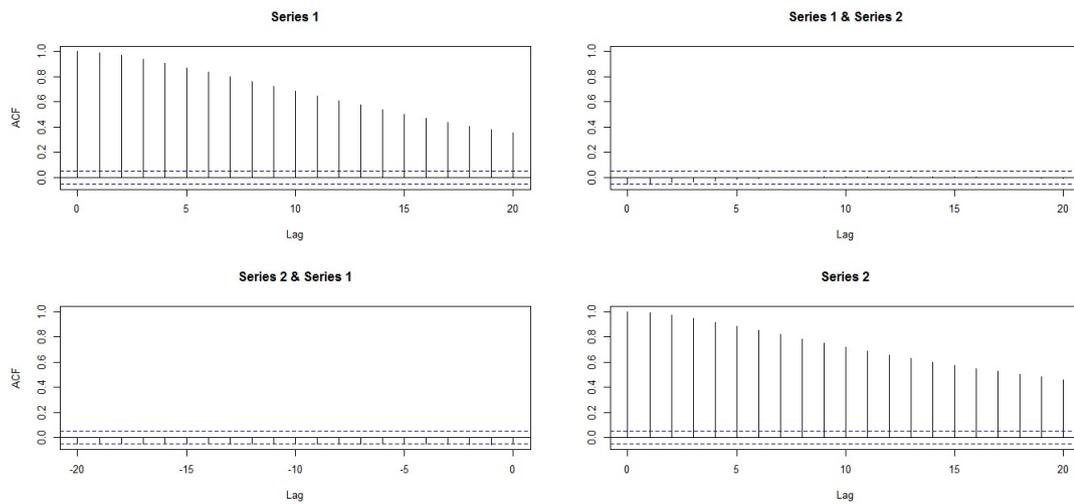


Figure 4.28: Sample moving cross-correlation plots of the retained two QMDPCs of the simulated series in Example 4.3.3.

Based on the above discussion, the dimension of the simulated series in this example was reduced from 10 to 2 by using QMDPCA with $W = 101$ and $l = 1$. The produced two QMDPCs explained almost all variation in the non-stationary series and produced a relatively small error when used to reconstruct the data.

Example 4.3.4 In this study, QMDPCA is applied to a non-stationary series \mathbf{z}_t that consists of 15 variables. This study includes two parts. The first part investigates the effect of applying QMDPCA with different combinations of W and l on the simulated series \mathbf{z}_t . The second part studies the effect of changing the sample size of \mathbf{z}_t on the results of QMDPCA. The series \mathbf{z}_t is generated using five different models, such that each model produces three dependent variables as described below.

Let a_t, b_t, c_t, d_t and e_t be independent standard normal white noises, which are the innovation terms of the following five models, respectively, then:

$$\left\{ \begin{array}{ll} z_{j,t} = u_{t+j-1}, & j = 1, 2 \text{ and } 3 \\ z_{j,t} = v_{t+j-4}, & j = 4, 5 \text{ and } 6 \\ z_{j,t} = w_{t+j-7}, & j = 7, 8 \text{ and } 9 \\ z_{j,t} = x_{t+j-10}, & j = 10, 11 \text{ and } 12 \\ z_{j,t} = q_{t+j-13}, & j = 13, 14 \text{ and } 15 \end{array} \right. \quad (4.20)$$

where $(u_t, v_t, w_t, f_t, q_t)'$ satisfies

$$\left\{ \begin{array}{l} \nabla u_t = -0.8 \nabla u_{t-1} + a_t - 1.2a_{t-1} \\ \nabla v_t = 0.55 \nabla v_{t-1} + b_t - 0.9b_{t-1} \\ \nabla w_t = 0.6 \nabla w_{t-1} + c_t + 2.2c_{t-1} \\ \nabla x_t = -0.75 \nabla x_{t-1} + d_t + 0.45d_{t-1} \\ \nabla q_t = 0.8 \nabla q_{t-1} + e_t + 1.5e_{t-1} \end{array} \right. \quad (4.21)$$

A time series plot of the simulated data is provided in figures 4.29 and 4.30, where the variables exhibited non-stationary behaviours.

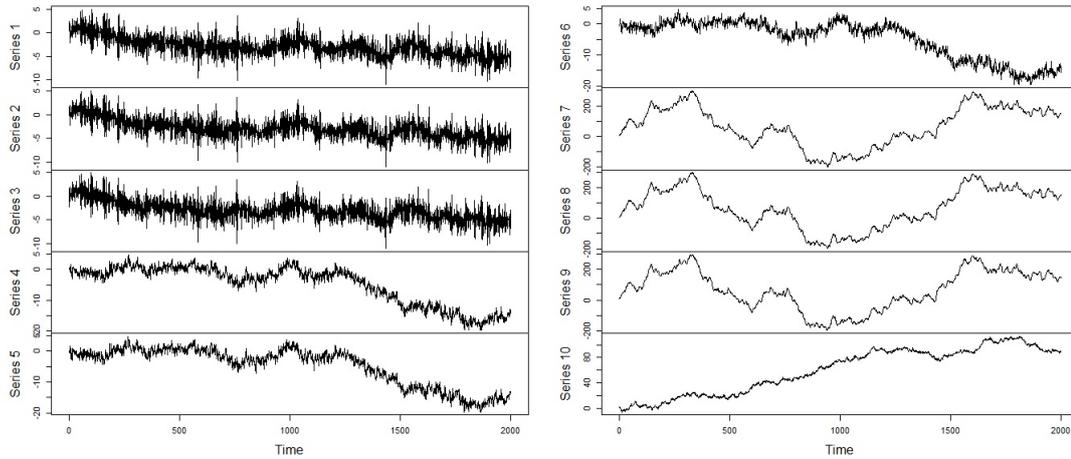


Figure 4.29: A time series plot of the simulated non-stationary series with 15 variables in Example 4.3.4 - Part 1.

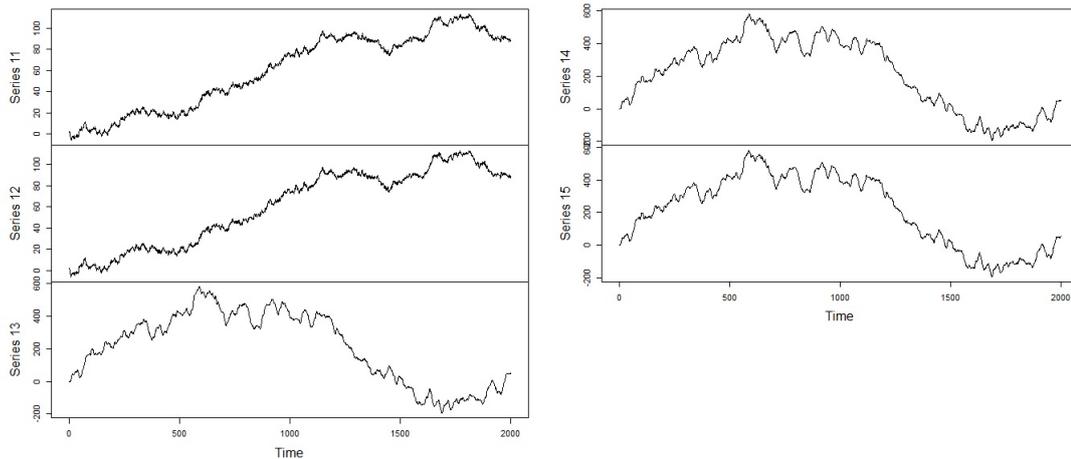


Figure 4.30: A time series plot of the simulated non-stationary series with 15 variables in Example 4.3.4 - Part 2.

Based on 500 replicas, Table 4.15 is a comparison between the results of applying QMDPCA using different combinations of W and l applied to \mathbf{z}_t with 2000 samples, where two QMDPCs are retained. For the mean percentage of explained variation by the retained QMDPCs, about 99.9% of the variability was explained in all cases. This percentage did not change significantly by using different combinations of W or l . For example, a mean percentage of 99.96% was obtained when using QMDPCA

with $W = 101$ and $l = 5$, compared with 99.99% for all other cases. These percentages did not change among the 500 replicas as they have zero standard deviations (when rounded to two decimals). For the mean of the MSE of reconstructed data, its values ranged from 134.93 to 158.99 for the considered combinations of W and l , where the lowest value was obtained when using $W = 301$ and $l = 1$, and the highest when using $W = 201$ and $l = 5$. These values are reasonable considering the dimensions of the simulated series.

Window Size (W)	W=101 ($l=1$)	W=201 ($l=1$)	W=301 ($l=1$)
Results			
Contribution % Mean (Standard Deviation)	99.99 % (0.00)	99.99 % (0.00)	99.99 % (0.00)
MSE Mean (Standard Deviation)	151.05 (62.96)	137.45 (138.84)	134.93 (137.02)
Window Size (W)	W=101 ($l=5$)	W=201 ($l=5$)	W=301 ($l=5$)
Results			
Contribution % Mean (Standard Deviation)	99.96 % (0.00)	99.99 % (0.00)	99.99 % (0.00)
MSE Mean (Standard Deviation)	156.91 (135.43)	158.99 (147.19)	146.37 (137.12)

Table 4.15: A comparison between the results of QMDPCA with different combinations of W and l in Example 4.3.4.

For the relationship between the retained QMDPCs, the plots of the mean of the absolute value of the moving cross-correlation indicated uncorrelated components for the cases where QMDPCA with $W = 101$ was used; See Figure 4.31. However, barely significant correlations were observed between the two components for the cases where $W = 201$ was used; See Figure 4.32. The correlations increased slightly when using QMDPCA with $W = 301$ compared with those of $W = 201$; See Figure 4.33. The standard deviations of the absolute value of the moving cross-correlation are reported in Table 4.16, where their values ranged between 0 to 0.07 in all cases. The latter indicates steady relationship between the retained QMDPCs across the 500 replicas.

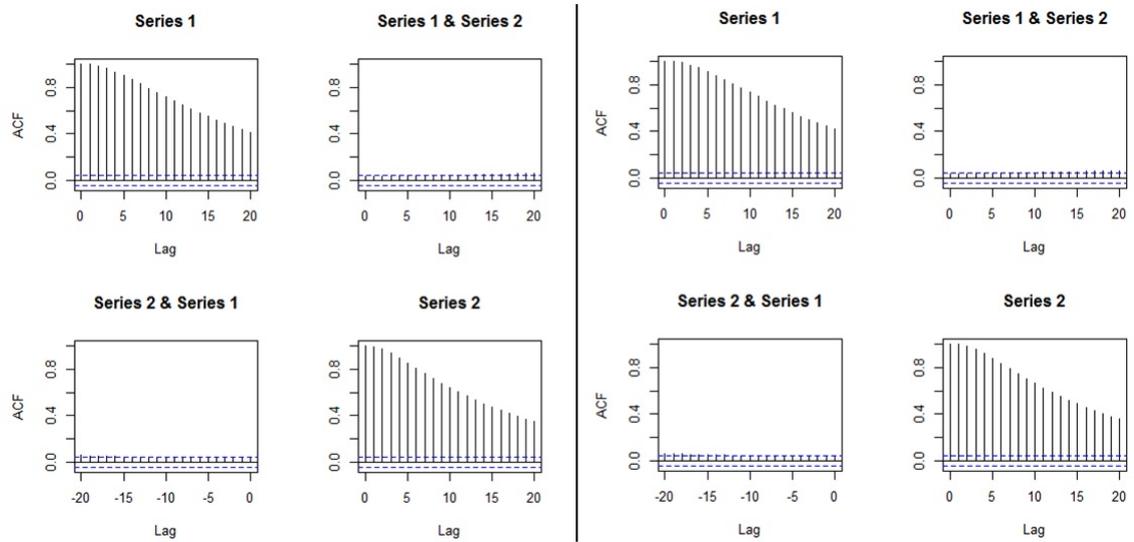


Figure 4.31: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.4 [Left: QMDPCA ($W = 101$, $l = 1$), Right: QMDPCA($W = 101$, $l = 5$)].

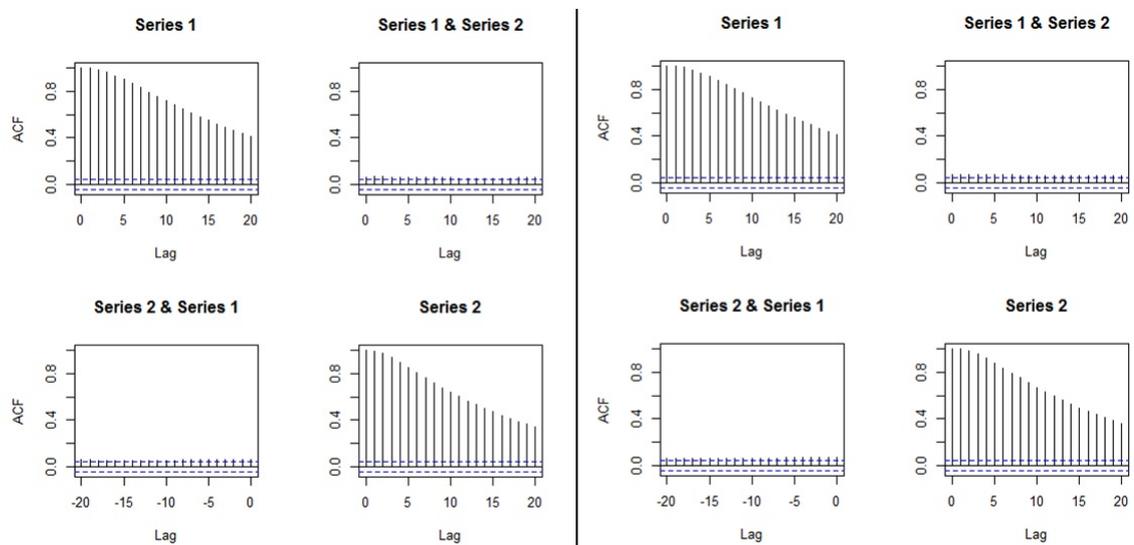


Figure 4.32: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.4 [Left: QMDPCA ($W = 201$, $l = 1$), Right: QMDPCA($W = 201$, $l = 5$)].

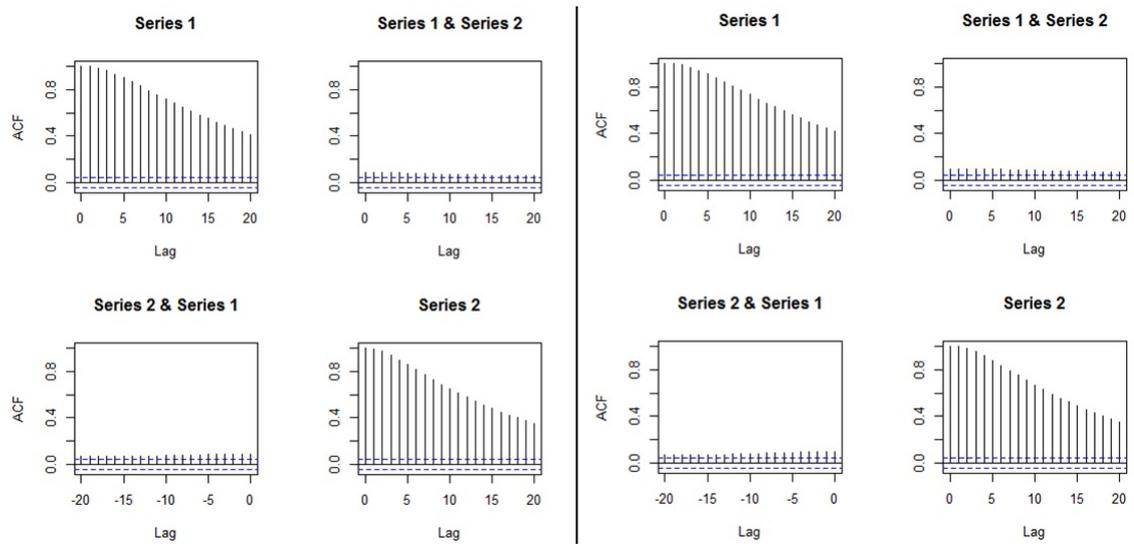


Figure 4.33: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs in Example 4.3.4 [Left: QMDPCA ($W = 301$, $l = 1$), Right: QMDPCA($W = 301$, $l = 5$)].

		QMDPCA(W=101, l=1)																			
Lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.00	0.01	0.01	0.01	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07
QMDPC1 vs QMDPC2	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
QMDPC2 vs QMDPC1	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07
		QMDPCA(W=201, l=1)																			
Lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.06	0.06
QMDPC1 vs QMDPC2	0.03	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
QMDPC2 vs QMDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
QMDPC2 vs QMDPC2	0.00	0.00	0.00	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07
		QMDPCA(W=301, l=1)																			
Lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06
QMDPC1 vs QMDPC2	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05
QMDPC2 vs QMDPC1	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
QMDPC2 vs QMDPC2	0.00	0.00	0.00	0.01	0.01	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07
		QMDPCA(W=101, l=5)																			
Lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.00	0.01	0.01	0.01	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07
QMDPC1 vs QMDPC2	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05
QMDPC2 vs QMDPC1	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05
QMDPC2 vs QMDPC2	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07
		QMDPCA(W=201, l=5)																			
Lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.00	0.01	0.01	0.01	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07
QMDPC1 vs QMDPC2	0.06	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
QMDPC2 vs QMDPC1	0.06	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
QMDPC2 vs QMDPC2	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07
		QMDPCA(W=301, l=5)																			
Lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.00	0.01	0.01	0.01	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07
QMDPC1 vs QMDPC2	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
QMDPC2 vs QMDPC1	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
QMDPC2 vs QMDPC2	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07

Table 4.16: Standard deviations of the absolute value of the moving cross-correlations between two QMDPCs with different combinations of W and l in Example 4.3.4.

Therefore, we can reduce the dimension of the simulated non-stationary series with 15 variables in this example by using QMDPCA with $W = 101$ and either $l = 1$ or $l = 5$.

In order to investigate the effect of changing the sample size of \mathbf{z}_t on QMDPCA's results, we apply QMDPCA with $W = 101$ and $l = 1$ on \mathbf{z}_t with sample sizes $T = \{200, 400, 600, 800\}$. Based on 500 replicas, Table 4.17 is a summary of the results, where two QMDPCs are selected. For all the considered sample sizes, we obtained a mean percentage of explained variation of 99.99% with zero standard deviations. Also, we notice that the mean MSE of reconstructed data and the corresponding standard deviation increased as we increased T . For example, a mean MSE of 18.15 with a standard deviation of 11.76 were obtained when $T = 200$, compared with 55.69 and 53.02 when $T = 800$. Additionally, the mean of the absolute value of the moving cross-correlation with $W = 101$ are plotted in figures 4.34 and 4.35. The retained two QMDPCs had significant correlations when $T = 200$. These correlations decreased with some minor, but significant, correlations for $|l| \geq 10$ when increasing the dimension to $T = 400$. Uncorrelated QMDPCs were obtained when using $T = 600$ and $T = 800$. The standard deviations of the absolute value of the moving cross-correlation are reported in Table 4.18, where the values improved (i.e. decreased) by increasing T . These standard deviations ranged between 0 to 0.24 when $T = 200$, compared with 0 to 0.12 when $T = 800$. Notice that the data in this example are not normalized, and as mentioned in the previous examples, normalizing the data would reduce the values of the reconstruction error and its standard deviations further.

Method	QMDPCA (W=101, l=5)			
Sample size	T=200	T=400	T=600	T=800
Contribution %	99.99 %	99.99 %	99.99 %	99.99 %
Mean (Standard Deviation)	(0.00)	(0.00)	(0.00)	(0.00)
MSE	18.15	29.53	40.56	55.69
Mean (Standard Deviation)	(11.76)	(25.63)	(33.74)	(53.02)

Table 4.17: A comparison between the results of QMDPCA with different sizes of T in Example 4.3.4.

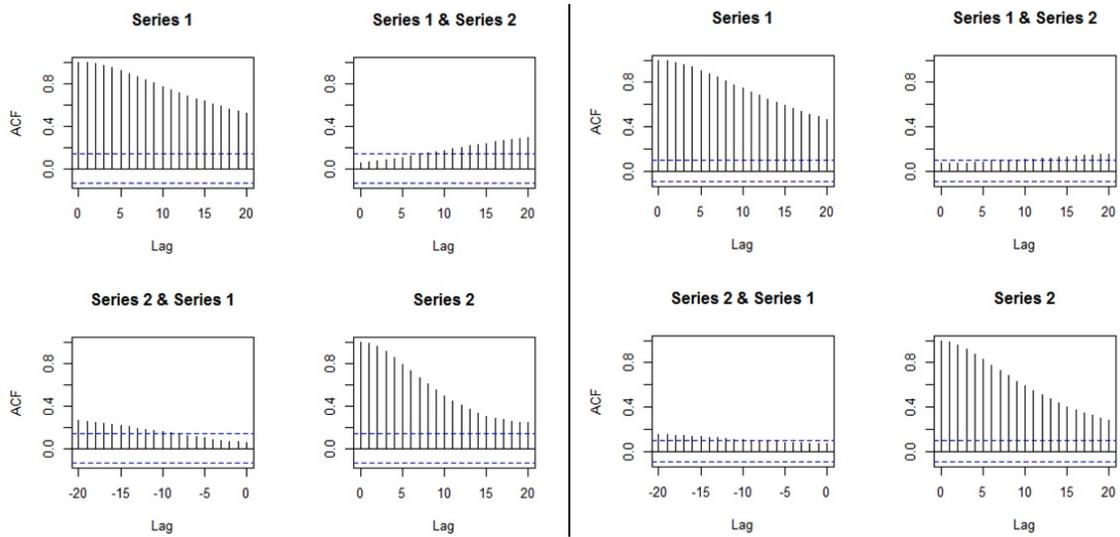


Figure 4.34: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.4 [Left: $T = 200$, Right: $T = 400$].

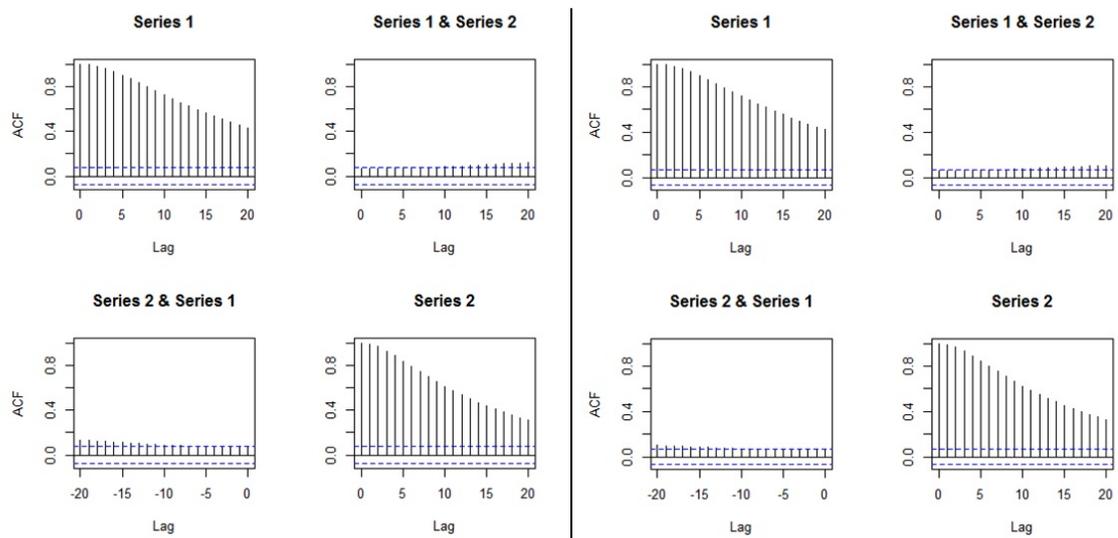


Figure 4.35: Plots of the mean of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.4 [Left: $T = 600$, Right: $T = 800$].

		T=200																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.02	0.03	0.05	0.06	0.08	0.10	0.11	0.13	0.15	0.16	0.17	0.19	0.20	0.20	0.21	0.21	0.22	0.23	0.24
QMDPC1 vs QMDPC2	0.06	0.06	0.06	0.06	0.07	0.07	0.09	0.08	0.09	0.09	0.10	0.11	0.12	0.12	0.13	0.14	0.15	0.15	0.16	0.17	0.18	0.18
QMDPC2 vs QMDPC1	0.06	0.06	0.06	0.06	0.07	0.08	0.08	0.09	0.10	0.11	0.12	0.13	0.13	0.14	0.15	0.16	0.16	0.17	0.17	0.18	0.18	0.18
QMDPC2 vs QMDPC2	0.00	0.01	0.02	0.04	0.06	0.09	0.11	0.13	0.16	0.17	0.19	0.20	0.21	0.21	0.21	0.21	0.21	0.20	0.19	0.19	0.18	0.18
		T=400																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.11	0.11	0.12	0.13	0.14	0.14	0.15	0.15
QMDPC1 vs QMDPC2	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.09	0.09	0.09	0.09	0.10	0.10	0.10	0.11	0.11	0.11	0.12
QMDPC2 vs QMDPC1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.10	0.11	0.11
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.02	0.04	0.05	0.07	0.08	0.10	0.11	0.12	0.13	0.14	0.14	0.15	0.15	0.16	0.16	0.16	0.16	0.17	0.17
		T=600																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.04	0.05	0.06	0.07	0.08	0.08	0.08	0.09	0.10	0.10	0.11	0.12	0.12	0.13	0.13
QMDPC1 vs QMDPC2	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.09	0.09	0.09	0.09
QMDPC2 vs QMDPC1	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.09	0.09	0.09
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.10	0.10	0.11	0.11	0.12	0.12	0.13	0.13	0.13	0.13
		T=800																				
Lag		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
QMDPC1 vs QMDPC1	0.00	0.00	0.00	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.06	0.06	0.07	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.11
QMDPC1 vs QMDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08
QMDPC2 vs QMDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.08	0.08	0.09	0.10	0.11	0.12	0.12	0.13	0.13	0.13	0.13
QMDPC1 vs QMDPC1	0.00	0.00	0.00	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.06	0.06	0.07	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.11
QMDPC1 vs QMDPC2	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08
QMDPC2 vs QMDPC1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08
QMDPC2 vs QMDPC2	0.00	0.00	0.01	0.02	0.03	0.04	0.05	0.05	0.06	0.06	0.07	0.08	0.08	0.09	0.10	0.11	0.12	0.12	0.13	0.13	0.13	0.13

Table 4.18: Standard deviations of the absolute value of the moving cross-correlations between two QMDPCs with different sizes of T in Example 4.3.4.

Therefore, by using QMDPCA with $W = 101$ and $l = 1$ in this example, we were able to reduce the dimension of \mathbf{z}_t from 15 to 2. QMDPCA performed well when applied to series with sample sizes $T \geq 600$. A reasonable performance was obtained when using sample sizes as small as $T = 400$.

By the end of the above simulation studies, we conclude that the proposed QMDPCA was able to reduce the dimension of stationary and non-stationary multivariate time series. The proposed method showed steady performance as we increased the dimension of the data in the simulation studies.

Determining the window size enables QMDPCA to extract accurate results from non-stationary series. We noticed that using a window size that is larger than required could cause QMDPCA to produce correlated QMDPCs. However, the opposite does not have to be true, where QMDPCA can produce accurate results using a window size that is smaller than required.

Including more lagged series in the analysis might add more dynamic information to the results of QMDPCA. We noticed that by including more lagged series in the analysis, QMDPCA produced QMDPCs with a slightly lower contribution percentage and slightly higher reconstruction error compared to those when fewer lagged series were included.

It was noticed that QMDPCA could produce a small number of components that explain almost all variation of the data and produce a small reconstruction error when applied on non-stationary series with small sample sizes. However, the retained QMDPCs might have some correlations appeared in larger time lags due to the continuous movement in the moving cross-covariance matrix. These correlations became non-significant when increasing the sample size. Therefore, it is not sufficient to evaluate the retained QMDPCs based on the contribution percentage and the reconstruction error only. It is important to investigate the relationship between the retained components to obtain uncorrelated QMDPCs.

4.3.2 Real Data Examples

Example 4.3.5 The data in this example consists of the daily stock prices of 10 US companies in US Dollar observed in 1036 days from Nov 07, 2013 to Dec 18, 2017. The variables in this example represent the stock prices of the following companies: Boeing (BA), Bank of America (BAC), Citigroup (C), Facebook (FB), Ford (F), General Electric (GE), General Motor(GM), The Home Depot (HD), IBM (IBM) and Intel (INTC). The data is available at *Yahoo!* Finance. In the following, we are going to apply QMDPCA to see if the dimension of the above series can be reduced.

Figure 4.36 is a time series plot of the daily stock prices of the US companies where all ten companies exhibited non-stationary behaviours over time. A window of $W = 101$ is suggested to be used in the following analysis considering the non-stationarity of the data. The dynamic relationships among the ten companies is revealed by the sample moving cross-correlation plots before applying QMDPCA; See the first 36 plots in Figure 4.37. In general, a moderated dynamic relationship was observed between the ten companies. For example, the companies Bank of America and Citigroup exhibited a strong correlation in all lags. Also, Facebook and Boeing had a moderate relationship. Moreover, a weak relationship existed between the companies Facebook and IBM.

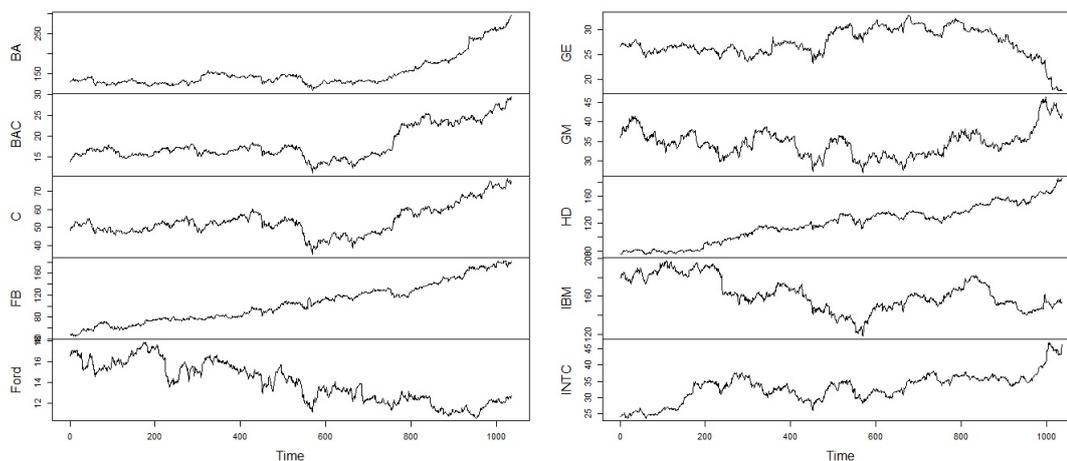


Figure 4.36: A time series plot of the daily stock prices of the ten US companies in Example 4.3.5

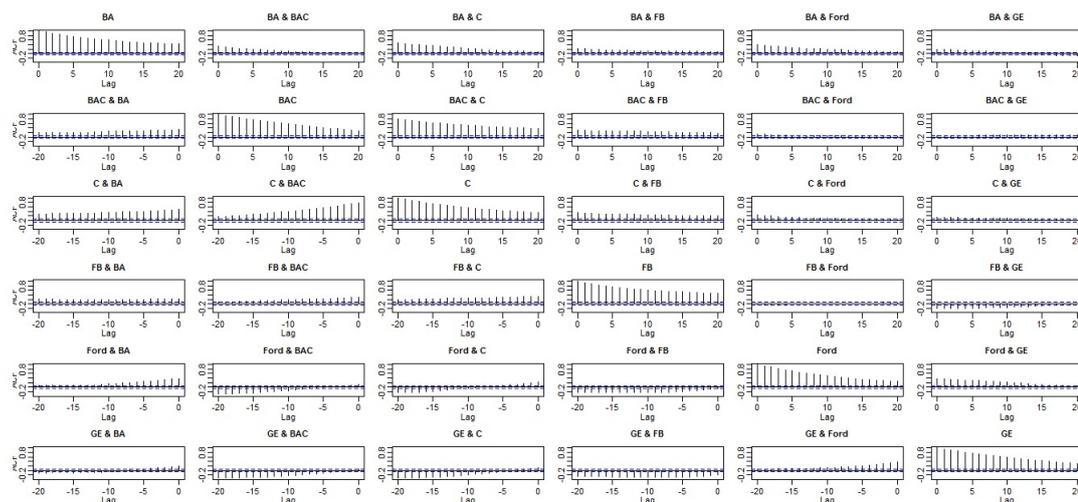


Figure 4.37: First 36 sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the ten US companies in Example 4.3.5 before QMDPCA is applied.

By investigating Figure 4.38, the optimal number of lags is suggested to be $l = 1$, where seven static and three dynamic relations were found. The eigenvalues and the RCC values are plotted in Figure 4.39, where the first three QMDPCs are suggested to be optimal with an RCC value of 0.33 (rounded to two decimals). The retained three QMDPCs explained 98.63% of the variation of the data and produced an

MSE of 134.26 when used to reconstruct the data. Additionally, the sample moving cross-correlation plots of the three QMDPCs indicated uncorrelated components; See Figure 4.40.

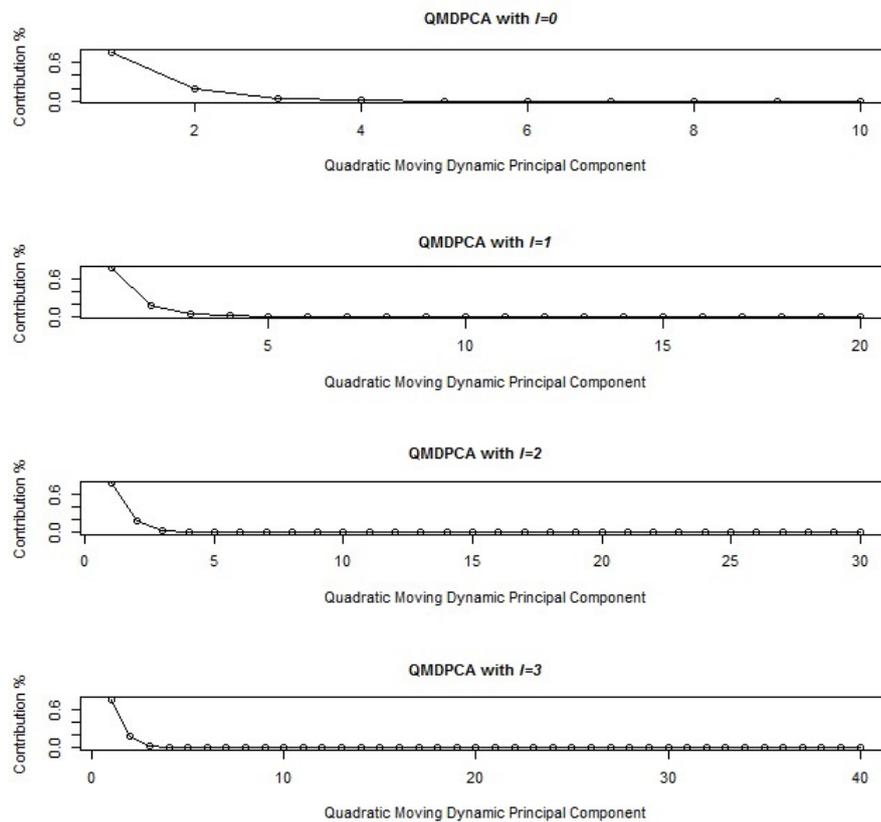


Figure 4.38: Eigenvalues' plots after applying QMDPCA with $W = 101$ to the ten US companies in Example 4.3.5.

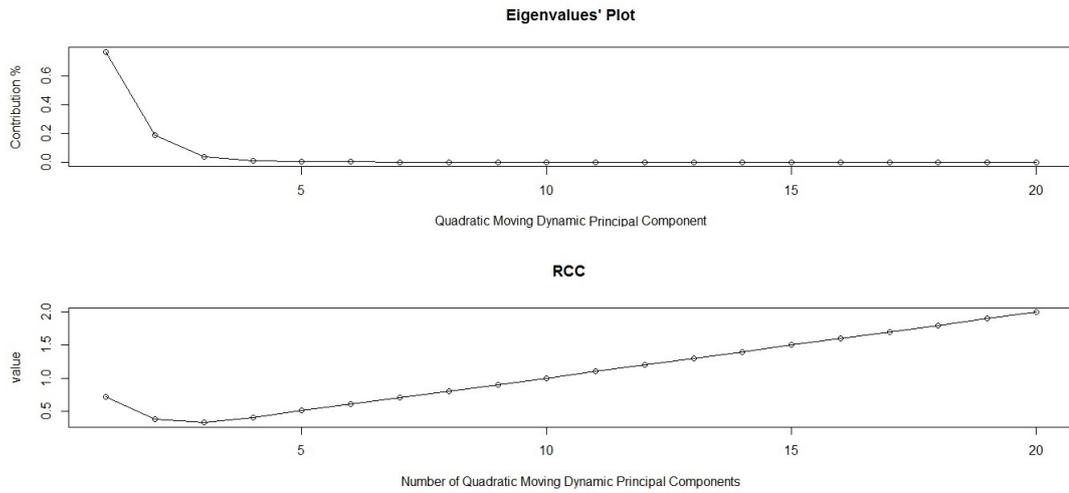


Figure 4.39: Top: An eigenvalues' plot of the ten US companies in Example 4.3.5 after applying QMDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot.

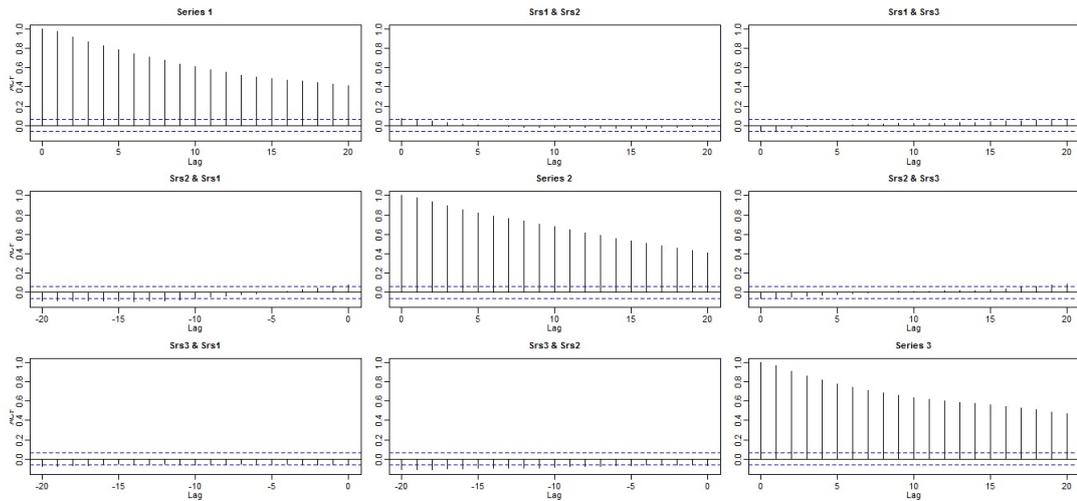


Figure 4.40: Sample moving cross-correlation plots of the three QMDPCs of the ten US companies in Example 4.3.5 after QMDPCA with $W=101$ and $l=1$ is applied.

In conclusion, QMDPCA with $W = 101$ and $l = 1$ was able to reduce the dimension of the US stock prices data from 10 to 3. The retained three QMDPCs explained almost all variation in the data and were able to reconstruct back the original data with a relatively small error.

Example 4.3.6 The data in this example consists of daily stock prices of eighteen US companies in US Dollar observed in 1036 days from Nov 07, 2013 to Dec 18, 2017. The eighteen variables are the ten variables in Example 4.3.5 additional to eight new variables. The names and labels of the eighteen US companies are summarized in Table 4.19. The data is available at *Yahoo!* Finance.

No.	Label	Name	No.	Label	Name
1	ADM	Archer-Daniels-Midland	10	IBM	IBM
2	BA	Boeing-The Boeing	11	INTC	Intel
3	BAC	Bank of America	12	JNJ	Johnson & Johnson
4	C	Citigroup	13	VLO	Valero Energy
5	FB	Facebook	14	VZ	Verizon
6	F	Ford Motor	15	WBA	Walgreens Boots Alliance
7	GE	General Electric	16	WFC	Wells Fargo & Company
8	GM	General Motor	17	WMT	Walmart
9	HD	The Home Depot	18	OXM	Exxon Mobil

Table 4.19: Names and labels of the 18 US companies in Example 4.3.6.

The non-stationary behaviours of daily stock prices of the eighteen US companies are shown in figures 4.41 and 4.42. Thus, a window of size $W = 101$ is used in the analyses. The sample moving cross-correlation plots revealed moderate relationships among the eighteen companies before applying QMDPCA; See the last 36 plots in Figure 4.43. For example, the company Walmart was strongly correlated with Walgreens Boots Alliance and weakly correlated with Verizon.

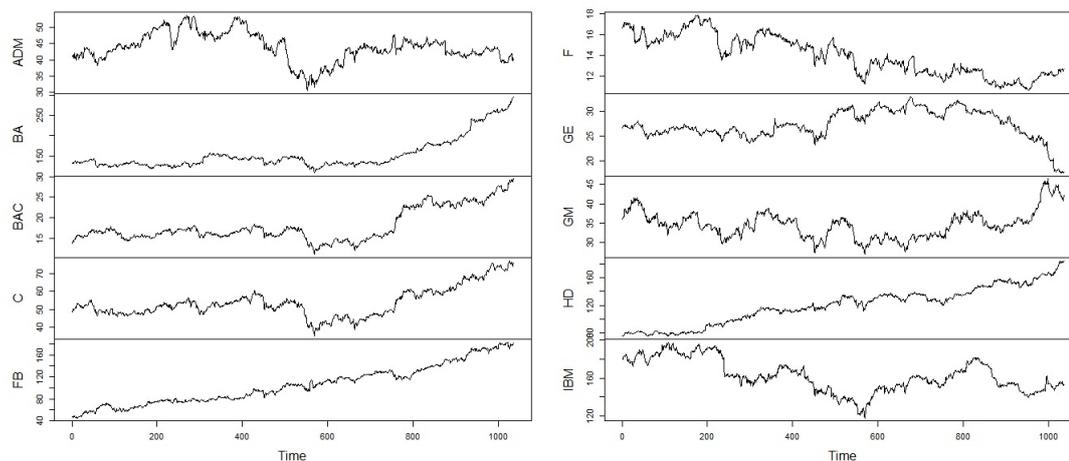


Figure 4.41: A time series plot of the daily stock prices of the 18 US companies in Example 4.3.6 - Part 1.

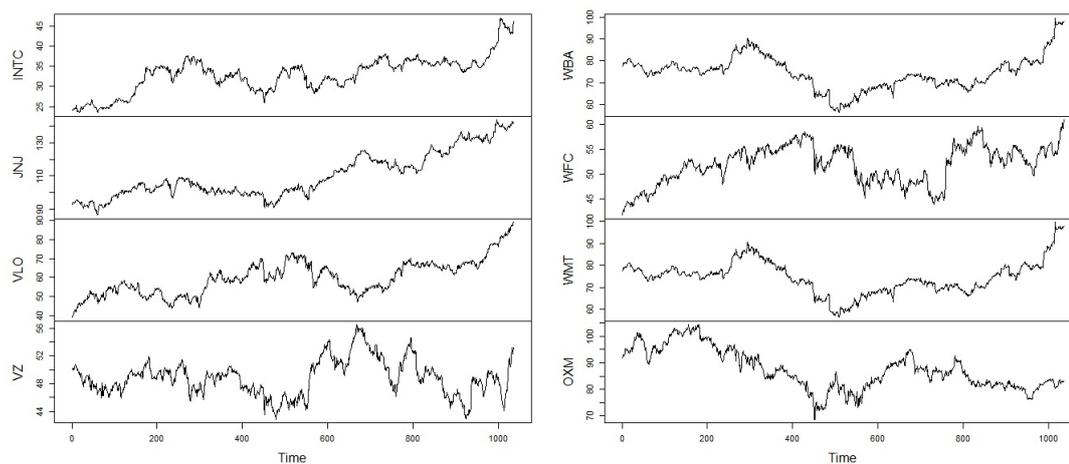


Figure 4.42: A time series plot of the daily stock prices of the 18 US companies in Example 4.3.6 - Part 2.

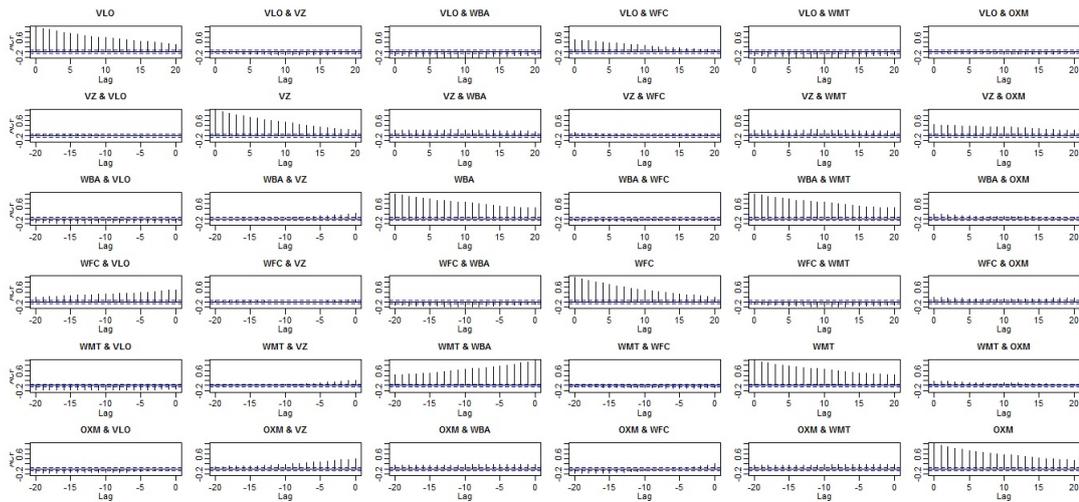


Figure 4.43: Last 36 sample moving cross-correlation plots with $W = 101$ of the daily stock prices of the 18 US companies in Example 4.3.6 before QMDPCA is applied.

The optimal number of lags is suggested to be $l = 1$ based on Figure 4.44, where fourteen static and four dynamic relations were observed. The eigenvalues and the RCC values are plotted in Figure 4.45. The RCC had values of 0.425, 0.305, 0.303 and 0.322 for the first two, three, four and five QMDPCs, respectively. Therefore, four QMDPCs were considered to be optimal to represent the data after transformation. The retained four QMDPCs explained 97.66% of the variation of the data and produced an MSE of 408.58 when used to reconstruct the data. Also, the sample moving cross-correlation plots of the four QMDPCs suggested uncorrelated components, as shown in Figure 4.46.

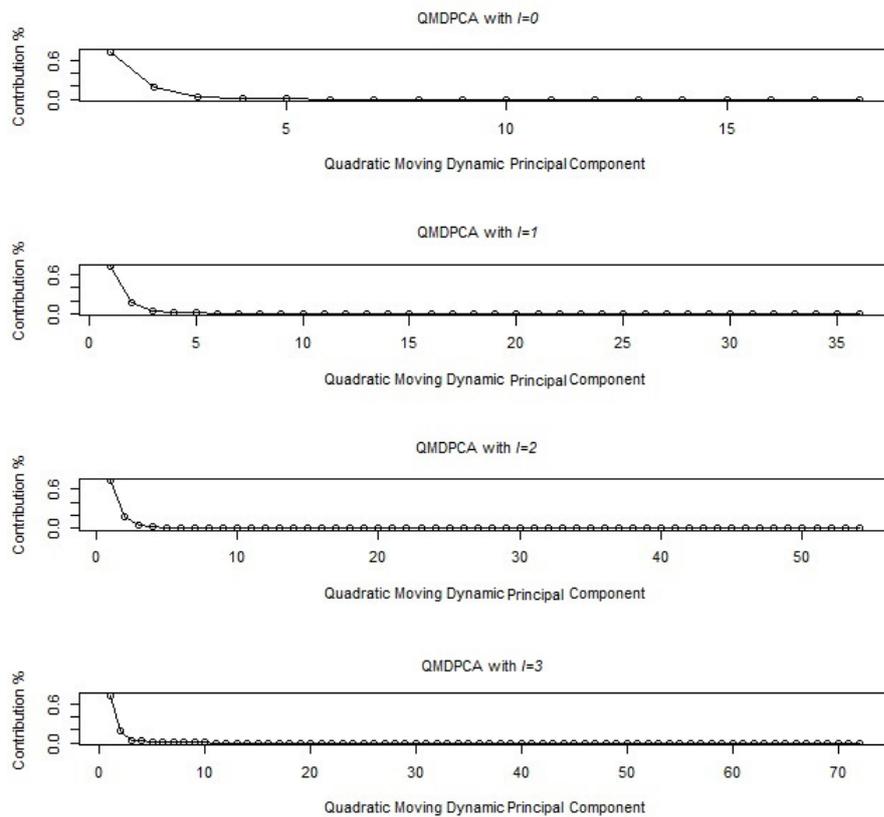


Figure 4.44: Eigenvalues' plots after applying QMDPCA with $W = 101$ to the 18 US companies in Example 4.3.6.

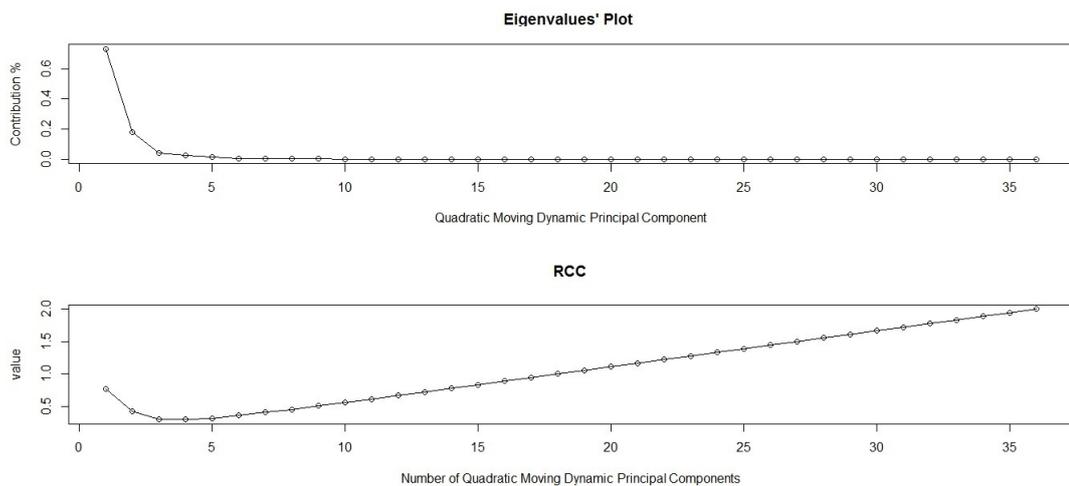


Figure 4.45: Top: An eigenvalues' plot of the 18 US companies in Example 4.3.6 after applying QMDPCA with $W = 101$ and $l = 1$. Bottom: RCC plot.

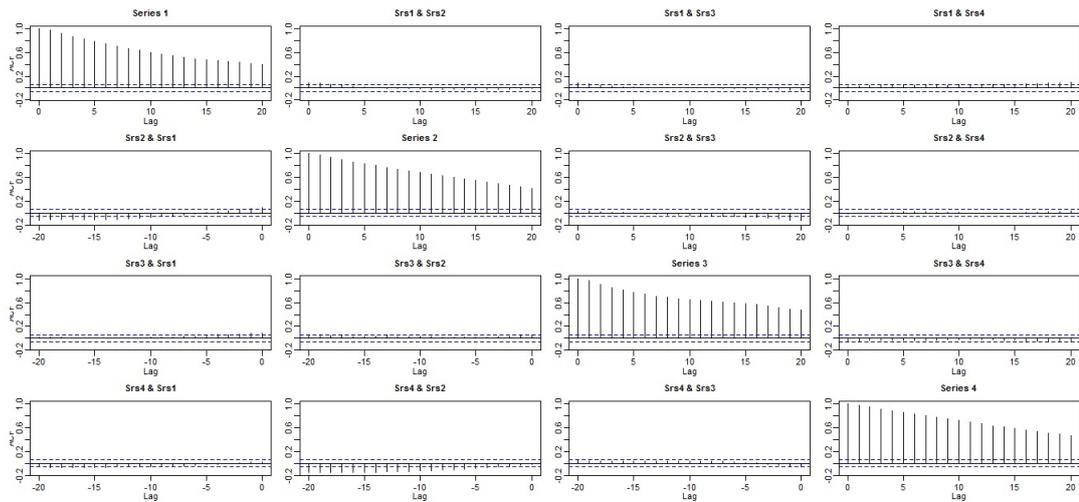


Figure 4.46: Sample moving cross-correlation plots of the QMDPCs of the 18 US companies in Example 4.3.6 after QMDPCA with $W=101$ and $l=1$ is applied.

To conclude, QMDPCA with $W = 101$ and $l = 1$ was able to reduce the dimension of the US daily stock prices data in this example from eighteen to four. The retained four components explained a high percentage of the variation in the data and were able to reconstruct the original data with a relatively small error.

4.4 Summary and Remarks

In this chapter, we extended the PCA to non-stationary time series data by presenting QMDPCA. This dimension reduction method is general in the sense that it can be applied on both stationary and non-stationary time series data by adjusting the window size to extract static and dynamic information from the data. The main difference between MDPCA from Chapter 3 and QMDPCA is that the former is based on eigenanalysis of the moving cross-covariance matrix of the data, where the latter is based on eigenanalysis of the quadratic moving cross-covariance matrix of the data.

We used three methods to evaluate the results of QMDPCA. The static and dynamic relationships between the final retained QMDPCs are investigated by using

the moving cross-correlation function. We also used reconstruction error (i.e. MSE of reconstructed data) and the percentage of contribution to evaluating the retained QMDPCs, where lower reconstruction errors and higher percentages of contribution are indications of better results.

In order to optimize QMDPCA's results, we used the RCC criterion that we introduced in Chapter 3 to determine the optimal number of retained QMDPCs. Choosing the window size for QMDPCA depends on the stationarity of the data. Shorter windows are suitable for series with stronger non-stationary movements, and the opposite is true.

We studied the theoretical properties of the estimator of \mathbf{U} , the matrix that consists of the eigenvectors of the quadratic moving cross-covariance matrix of the data. We showed that $\hat{\mathbf{U}}$ is a consistent estimator of \mathbf{U} with a convergence rate $W^{-1/2}$ under some regularity assumptions.

Many simulations were carried out to show the numerical properties of QMDPCA on data with different dimensions and sample sizes. QMDPCA showed consistent performance on both stationary and non-stationary time series data. QMDPCA was also illustrated on real data sets. It was noticed that QMDPCA could produce QMDPCs that are correlated in larger time lags (i.e. $|l| \leq 15$) when applied to a non-stationary series. These correlations appeared due to the continuous movement in the moving cross-covariance matrix. One can overcome this problem by increasing the sample size. Therefore, it is not sufficient to evaluate QMDPCA's results based on the contribution percentage and the reconstruction error only. It is also important to investigate the relationship between the retained components and obtain QMDPCs that are uncorrelated. Similar conclusion was reported for MDPCA in Chapter 3, as it also uses the moving cross-covariance matrix in its calculations.

We noticed that QMDPCA and MDPCA produced similar results when applied on non-stationary series with higher percentages of explained variation by the re-

tained components of QMDPCA. A similar conclusion was obtained for the stationary case when comparing the results of QMDPCA and DPCA of Ku et al. (1995). The reason for this is because QMDPCA produces eigenvalues that have a quadratic order (i.e. squared eigenvalues) of those produced by MDPCA and DPCA.

Chapter 5

Generalized Principal Component Analysis for Time Series

In this chapter, a new dimension reduction tool will be proposed. The method we propose is general in the sense that it can be applied to a wide range of both stationary and non-stationary time series data with large dimensions. This method can be considered as a generalization of the stationary TS-PCA of Chang et al. (2018) to non-stationary data and, therefore, will be called the generalised principal component analysis of time series (GTS-PCA). Similar to TS-PCA, the results of GTS-PCA are also uncorrelated subseries (i.e. subgroups) with smaller dimensions that can be analysed individually. In this chapter, we are going to provide a full description of the GTS-PCA method and describe how its model is derived. Also, a new columns' rearrangement tool called the maximum moving cross-correlation method will be proposed to enhance the performance of GTS-PCA. The theoretical properties of the used estimators will be studied. The performance of the GTS-PCA will be tested on both simulated and real data.

5.1 Methodology

Consider an m -dimensional time series $\mathbf{z}_t = (z_{1,t}, z_{2,t}, \dots, z_{m,t})'$ that can be non-stationary. We are seeking a linear transformation

$$\mathbf{z}_t = \mathbf{A}\mathbf{x}_t \tag{5.1}$$

where \mathbf{A} is an $m \times m$ unknown constant matrix and \mathbf{x}_t is a hidden m -dimensional time series that is segmented into n subseries such that $\mathbf{x}_t = (x_{1,t}, x_{2,t}, \dots, x_{n,t})'$, where $1 < n \leq m$. These n subseries are uncorrelated with each other. Hence the autocorrelation matrices of \mathbf{x}_t are block-diagonal matrices with n being the number of blocks.

Since \mathbf{z}_t might be non-stationary, the classical cross-covariance function will not be able to truly measure the linear dynamic dependence of \mathbf{z}_t as it uses a fixed mean vector for the calculation across the whole series. To overcome this problem, we shall use another function that can be updated as we move in time. Hence, we are going to use the moving cross-covariance matrix that we introduced in Chapter 3 as it can measure the linear dynamic dependence of both stationary and non-stationary time series. Before we proceed further, the following definitions are needed.

We assume that \mathbf{z}_i is locally stationary. Define the lag l cross-covariance matrix of \mathbf{z}_i as

$$\Gamma_{z,i}(l) = Cov(\mathbf{z}_i, \mathbf{z}_{i-l}) \quad (5.2)$$

where l is a non-negative integer. Once the series \mathbf{z}_t is observed, the sample lag l cross-covariance matrix over window \mathbf{z}_i , with a pre-specified size of $2w + 1$, can be used to estimate $\Gamma_{z,i}(l)$ as follows

$$\hat{\Gamma}_{z,i}(l) = \frac{1}{2w + 1} \sum_{t=i-w}^{i+w} (\mathbf{z}_t - \bar{\mathbf{z}}_i)(\mathbf{z}_{t-l} - \bar{\mathbf{z}}_i)' \quad (5.3)$$

where

$$\bar{\mathbf{z}}_i = \frac{1}{2w + 1} \sum_{t=i-w}^{i+w} \mathbf{z}_t,$$

and w is a positive integer. Then, the lag l moving cross-covariance matrices of \mathbf{z}_t and \mathbf{x}_t , respectively, are defined as

$$\mathbf{M}\Gamma_z(l) = \frac{1}{T - 2 \max(l, w)} \sum_{i=\max(l,w)+1}^{T-\max(l,w)} \Gamma_{z,i}(l) \quad (5.4)$$

and

$$\mathbf{M}\Gamma_x(l) = \frac{1}{T - 2 \max(l, w)} \sum_{i=\max(l,w)+1}^{T-\max(l,w)} \Gamma_{x,i}(l),$$

where $\Gamma_{x,i}(l)$ is defined for \mathbf{x}_t similarly to that for \mathbf{z}_t . The sample lag l moving cross-covariance matrix of \mathbf{z}_t is calculated as

$$\hat{\mathbf{M}}\Gamma_z(l) = \frac{1}{T - 2 \max(l, w)} \sum_{i=\max(l,w)+1}^{T-\max(l,w)} \hat{\Gamma}_{z,i}(l) \quad (5.5)$$

where $\hat{\mathbf{M}}\Gamma_x(l)$ is defined for \mathbf{x}_t similarly to that for \mathbf{z}_t . When $l = 0$, we use the notation $\mathbf{M}\Gamma_z$ and $\mathbf{M}\Gamma_x$ to refer to the moving cross-covariance matrix of the series \mathbf{z}_t and \mathbf{x}_t , respectively. Notice that $\hat{\mathbf{M}}\Gamma(l)$ extracts its information from the sample lag l cross-covariance matrices defined over moving local windows, similar to $\hat{\mathbf{M}}\Gamma$ defined in (3.6). However, $\hat{\mathbf{M}}\Gamma(l)$ defined in (5.5) is slightly different from $\hat{\mathbf{M}}\Gamma$. The former uses lagged series in its calculations, where the latter does not use lags as they are already included in the extended data vector.

We assume that the moving cross-covariance matrix between any two of the n uncorrelated subseries $\mathbf{x}_{j,t}$ and $\mathbf{x}_{k,t}$ of \mathbf{x}_t is essentially the zero matrix at any time

$$\mathbf{M}\Gamma(\mathbf{x}_{j,t}, \mathbf{x}_{k,s}) = \mathbf{0} \quad \text{for all } t, s \text{ and } j \neq k. \quad (5.6)$$

For identifiability, we assume without loss of generality that

$$\mathbf{M}\Gamma_z = \mathbf{I}_m \quad \text{and} \quad \mathbf{M}\Gamma_x = \mathbf{I}_m. \quad (5.7)$$

This can be achieved by replacing \mathbf{z}_t by $[\hat{\mathbf{M}}\Gamma_z]^{-1/2} \mathbf{z}_t$. The above assumption will not affect the block structure of \mathbf{x}_t . Under the above assumptions, the constant matrix \mathbf{A} is orthonormal since

$$\mathbf{M}\Gamma_z = \mathbf{A} \mathbf{M}\Gamma_x \mathbf{A}' = \mathbf{A} \mathbf{A}' = \mathbf{I}_m \quad (5.8)$$

and therefore, dimension reduction using GTS-PCA will lead to block-diagonal moving autocorrelation matrices of \mathbf{x}_t with n blocks. Since \mathbf{A} is orthonormal, then equation (5.1) can be re-written as

$$\mathbf{x}_t = \mathbf{A}' \mathbf{z}_t. \quad (5.9)$$

For stationary time series, in order to uniquely define \mathbf{x}_t and \mathbf{A} , Chang et al. (2018) proposed the use of the sum of a quadratic order of the cross-correlation matrices of \mathbf{z}_t up to a pre-specified lag l_1 . The result is a positive definite matrix whose eigenvectors' matrix is equivalent to \mathbf{A} after rearranging its columns. Once \mathbf{A} is found, the hidden segmentations can be found as the matrix \mathbf{x}_t can be easily calculated using equation (5.9) above. This can be extended to account for non-stationary data based on the definitions and the assumptions we have so far as follows. Define the quadratic order moving cross-covariance matrices of \mathbf{z}_t and \mathbf{x}_t , respectively, as

$$\mathbf{M}\mathbf{W}_z = \mathbf{I}_m + \sum_{l=1}^{l_1} \mathbf{M}\Gamma_z(l) \mathbf{M}\Gamma_z(l)' \quad (5.10)$$

and

$$\mathbf{M}\mathbf{W}_x = \mathbf{I}_m + \sum_{l=1}^{l_1} \mathbf{M}\Gamma_x(l) \mathbf{M}\Gamma_x(l)'.$$

Thus, we have the following results:

1. The matrices $\mathbf{M}\mathbf{W}_z$ and $\mathbf{M}\mathbf{W}_x$ will be the generalized symmetric quadratic order of the cross-covariance matrices used in the stationary TS-PCA and they are also positive-definite matrices.
2. The matrices $\mathbf{M}\Gamma_x(l)$ and $\mathbf{M}\mathbf{W}_x$ are block-diagonal $m \times m$ matrices consist of n blocks.
3. It follows from equation (5.1) that

$$\mathbf{M}\mathbf{W}_z = \mathbf{A} \mathbf{M}\mathbf{W}_x \mathbf{A}' \quad (5.11)$$

Once the series \mathbf{z}_t is observed, the matrix $\mathbf{M}\mathbf{W}_z$ can be estimated as follows

$$\hat{\mathbf{M}}\mathbf{W}_z = \mathbf{I}_m + \sum_{l=1}^{l_1} \hat{\mathbf{M}}\Gamma_z(l) \hat{\mathbf{M}}\Gamma_z(l)' \quad (5.12)$$

where $\hat{\mathbf{M}}\Gamma_z(l)$ is defined in (5.5).

Spectral decomposition of the symmetric matrix $\mathbf{M}\mathbf{W}_x$ yields that

$$\mathbf{M}\mathbf{W}_x = \mathbf{U}_x \Lambda \mathbf{U}_x'$$

where \mathbf{U}_x is an orthogonal matrix whose columns are the eigenvectors of $\mathbf{M}\mathbf{W}_x$ and Λ is a diagonal matrix whose diagonal elements are the eigenvalues of $\mathbf{M}\mathbf{W}_x$. It follows from equation (5.11) that

$$\mathbf{M}\mathbf{W}_z \mathbf{A} \mathbf{U}_x = \mathbf{A} \mathbf{U}_x \Lambda \quad (5.13)$$

which means that $\mathbf{U}_z \equiv \mathbf{A} \mathbf{U}_x$ is the orthogonal matrix whose columns are the eigenvectors of $\mathbf{M}\mathbf{W}_z$ after rearranging its columns. Furthermore, left-multiplying \mathbf{z}_t by the transposed matrix of \mathbf{U}_z and noting that \mathbf{A} is orthogonal we have

$$\mathbf{U}'_z \mathbf{z}_t = \mathbf{U}'_z \mathbf{A} \mathbf{x}_t = \mathbf{U}'_x \mathbf{A}' \mathbf{A} \mathbf{x}_t = \mathbf{U}'_x \mathbf{x}_t. \quad (5.14)$$

The expression in (5.14) exposes the hidden segments, which are the components of the transformed series $\mathbf{U}'_z \mathbf{z}_t$. Here, \mathbf{U}'_z is the transformation matrix used to project the original data into a new space where the transformed data is segmented into n uncorrelated subseries with lower dimensions that sum up to m . We assume that different blocks of the quadratic matrix $\mathbf{M}\mathbf{W}_x$ have different eigenvalues. In the event where different blocks of $\mathbf{M}\mathbf{W}_x$ have at least one common eigenvalue, then the prospective blocks should be merged together to form one block. Also, following Proposition 1 in Chang et al. (2018), $\mathbf{U}'_z \mathbf{z}_t$ is equivalent to $\mathbf{A}' \mathbf{z}_t$ subject to columns' rearrangement.

Since GTS-PCA uses moving cross-covariance matrices in its calculation, then it is crucial to choose a suitable window size to enhance the results. Recall, the choice of the window size is case-dependent and can be determined based on the stationarity of the data under study. The issue of choosing a suitable window size for the moving cross-covariance matrix was discussed in Section 3.1 in Chapter 3, and therefore it will be omitted here.

Based on the previous discussion, to reduce the dimension using GTS-PCA and find the hidden segmentation of an observed multivariate time series \mathbf{z}_t , the following steps can be used:

1. Replace \mathbf{z}_t in equation (5.1) by $[\hat{\mathbf{M}}\hat{\Gamma}_z]^{-1/2} \mathbf{z}_t$.

2. Calculate $\hat{\mathbf{M}}\hat{\mathbf{W}}_z$ defined in (5.12).
3. Calculate $\hat{\mathbf{U}}_z$, the orthogonal matrix that consists of the eigenvectors of $\hat{\mathbf{M}}\hat{\mathbf{W}}_z$.
4. Calculate $\hat{\mathbf{U}}_z' \mathbf{z}_t$, which is equivalent to $\hat{\mathbf{A}}' \mathbf{z}_t$ subject to columns' rearrangement.
5. Find $\hat{\mathbf{x}}_t = \hat{\mathbf{A}}' \mathbf{z}_t$ by rearranging the columns of $\hat{\mathbf{U}}_z' \mathbf{z}_t$ such that $\hat{\mathbf{x}}_t$ is segmented into n uncorrelated subseries.

5.2 Columns' Rearrangement

The main goal of this section is to find the correct columns' rearrangement of $\hat{\mathbf{U}}_z$, which returns $\hat{\mathbf{A}}$; See step 4 and 5 of the GTS-PCA procedure. This can be done by correctly grouping the correlated components of the transformed series $\hat{\mathbf{U}}_z' \mathbf{z}_t$ in order to find its equivalent $\hat{\mathbf{A}}' \mathbf{z}_t$. For ease of notation, denote $\hat{\mathbf{f}}_t = \hat{\mathbf{U}}_z' \mathbf{z}_t$. The following strategy can be used to reach our goal. Start with each component of $\hat{\mathbf{f}}_t$ as a separate group. Then merge any two correlated groups into one group. Repeat the last step until no further groups are correlated.

While a pairwise multiple hypothesis testing can be used to find correlated components, other methods in literature are based on ratio criteria instead. These ratio tests are favourable when the dimension of the series is large because they use less calculation to find correlated components compared with the other methods. Chang et al. (2018) developed the maximum cross-correlation method that can capture the dynamic dependence of stationary series. However, this method is not suitable for non-stationary series because it uses the classical correlation function in its calculations. To this end, we are going to use the moving cross-correlation function that was introduced in Section 3.1 in Chapter 3 to find correlated components. Additionally, a new ratio based method will be developed to detect the correlated components of a large dimensional time series.

5.2.1 The Moving Cross-Correlation Function

Recall, the moving cross-correlation function defined in (3.16) in Chapter 3 is used to examine the dynamic dependence between the components (i.e. variables) of both stationary and non-stationary time series. Here, this function will be mainly used to investigate the relationships between the m components of the transformed series $\hat{\mathbf{f}}_t$ in order to correctly group the correlated components and find the hidden segmentations. Based on definition (3.16), the lag l moving cross-correlation matrix of the series \mathbf{z}_t can be calculated as follows:

$$\mathbf{M}\rho_z(l) = \frac{1}{T - 2 \max(l, w)} \sum_{i=\max(l, w)+1}^{T-\max(l, w)} \mathbf{S}_i^{-1} \Gamma_{z,i}(l) \mathbf{S}_{i-l}^{-1} \quad (5.15)$$

where $\Gamma_{z,i}(l)$ is defined in (5.2) and \mathbf{S}_i is the diagonal matrix of the standard deviations of \mathbf{z}_i . The $(j, j)^{th}$ element of \mathbf{S}_i is the square root of the $(j, j)^{th}$ element of $\Gamma_{z,i}(0)$ defined over \mathbf{z}_i . Denote the $(j, k)^{th}$ element of $\mathbf{M}\rho_z(l)$ by $\mathbf{M}\rho_{z,jk}(l)$, which is the lag l moving correlation between the components $z_{j,t}$ and $z_{k,t-l}$.

Notice that the moving cross-correlation matrix has the following property

$$\mathbf{M}\rho_z(l) = [\mathbf{M}\rho_z(-l)]'. \quad (5.16)$$

Therefore, in order to calculate and plot the moving cross-correlation between the components of the series \mathbf{z}_t in different time lags, it is enough to calculate $\mathbf{M}\rho_z(l)$ for positive lags (i.e. $l \geq 0$), then use the property in (5.16) to calculate $\mathbf{M}\rho_z(l)$ for negative lags (i.e. $l < 0$). The function in (5.15) will be used to evaluate the relationship between the components produced by GTS-PCA. For further information and examples about the usage of the moving cross-correlation function and its plots, the reader is referred to Section 3.1 in Chapter 3.

For the case where the time series has a large number of variables, the moving cross-correlation plots will not be practical as there will be $m(m - 1)$ individual plots to be examined for a series with dimension m . For example, if a series has a dimension $m = 10$, then the number of individual plots will be 90. Hence, the use

of multiple hypothesis testing or a ratio-based method is recommended. Therefore, an extension of the maximum cross-correlation method of Chang et al. (2018) will be proposed. The new method will be called the maximum moving cross-correlation method.

5.2.2 The Maximum Moving Cross-Correlation Method

The maximum moving cross-correlation method is a ratio-based test that will be used to test for significant correlations between the components (i.e. variables) of both stationary and non-stationary series. In general, for an observed series \mathbf{z}_t , we consider its j^{th} and k^{th} variables to be correlated if $\hat{\mathbf{M}}\rho_{z,jk}(l)$, the $(j, k)^{\text{th}}$ element of $\hat{\mathbf{M}}\rho_z(l)$, is significantly different from zero over different time lags. In other words, the two components $\mathbf{z}_{j,t}$ and $\mathbf{z}_{k,t}$ are correlated if we reject the hypothesis

$$H_o : \mathbf{M}\rho_{z,jk}(l) = 0 \quad \text{for } l = 0, \pm 1, \pm 2, \dots, \pm p \quad (5.17)$$

against

$$H_a : \mathbf{M}\rho_{z,jk}(l) \neq 0 \quad \text{for some } -p \leq l \leq p$$

where p is a positive integer.

As mentioned before, the maximum cross-correlation method is practical when the dimension m is large as it uses only a few steps to find the correlated components. The maximum moving cross-correlation statistics between components $\mathbf{z}_{j,t}$ and $\mathbf{z}_{k,t}$ is calculated as

$$\hat{\mathbf{L}}_d(j, k) = \max_{-p \leq l \leq p} |\hat{\mathbf{M}}\rho_{z,jk}(l)|, \quad 1 \leq j < k \leq m. \quad (5.18)$$

A total of $d = m(m - 1)/2$ values for the maximum moving cross-correlation statistics are needed to be calculated and then reordered in decreasing order as $\hat{\mathbf{L}}_1 \geq \hat{\mathbf{L}}_2 \geq \dots \geq \hat{\mathbf{L}}_d$.

We use the following ratio criterion to reject the hypothesis in (5.17) based on the statistics calculated in (5.18). Consider only the correlation corresponding to

the largest \hat{r} maximum statistics in (5.18) to be significantly correlated, where \hat{r} is defined as

$$\hat{r} = \arg \max_{1 \leq i \leq hd} \hat{\mathbf{L}}_i / \hat{\mathbf{L}}_{i+1} \quad (5.19)$$

where h is a pre-specified number and $0 < h < 1$. A value of $h = 0.75$ will be used as recommended by Chang et al. (2018) to avoid extreme cases, where the dominator in (5.19) is extremely close to zero (i.e. $\hat{\mathbf{L}}_{i+1} \simeq 0$). Therefore, we report that the two components corresponding to $\hat{\mathbf{L}}_j$ are significantly correlated, where $1 \leq j \leq r$. Notice that, using $h = 0.75$ is one drawback of this method. A small amount of information might be lost because of using h , especially for extreme cases where strong correlations exist among all tested components. For these extreme cases, it is recommended to validate the results of the maximum moving cross-covariance method with those of the moving cross-correlation plots.

Therefore, the maximum moving cross-correlation method can be applied to find the n correlated components of $\hat{\mathbf{f}}_t$ and hence correctly rearrange its columns to find the hidden segmentation of $\hat{\mathbf{x}}_t$ as summarized below:

1. Calculate $\hat{\mathbf{L}}_d(j, k)$ for the $d = \frac{m(m-1)}{2}$ pairs of the components of $\hat{\mathbf{f}}_t$.
2. Rearrange the values of the calculated d statistics in step 1 from largest to smallest (i.e. $\hat{\mathbf{L}}_1 \geq \hat{\mathbf{L}}_2 \geq \dots \geq \hat{\mathbf{L}}_d$).
3. Find the value of \hat{r} using the criterion in (5.19).
4. Consider the pair corresponding to the statistics $\hat{\mathbf{L}}_j$ to be significantly correlated (i.e. reject the corresponding null hypothesis), where $1 \leq j \leq \hat{r}$.
5. Rearrange the columns of $\hat{\mathbf{f}}_t$ by grouping correlated components according to step 4. The final result of this step is the vector with the hidden segmentation, $\hat{\mathbf{x}}_t$.

These steps can be considered as sub-steps of step 5 of the GTS-PCA procedure.

Example 5.2.1 This example is a short simulation study to test the ability of the maximum moving cross-correlation method to regroup the variables of a non-stationary time series based on their dynamic relationships. The simulated data

with eight variables and 1200 observations in Example 3.1.2 are used. Recall that this series consists of 3 subseries of 4, 3 and 1 variable such that each subseries was generated using different non-stationary model.

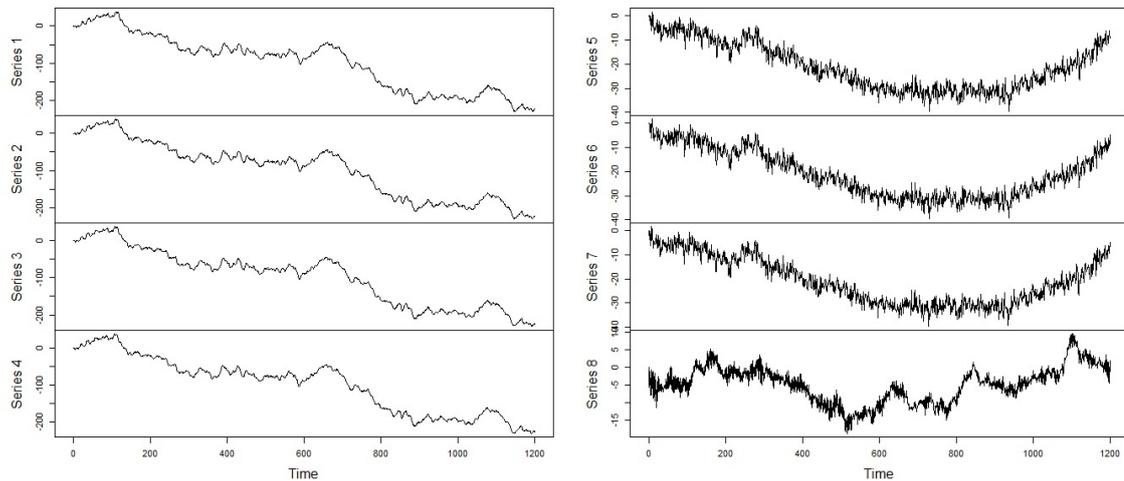


Figure 5.1: A time series plot of the simulated series of eight variables in Example 5.2.1.

Results Method	Correlated Variables
Maximum Moving Cross-Correlation (window=101)	{1, 2, 3, 4}, {5, 6, 7} and {8}
Moving Cross-Correlation (window=101)	{1, 2, 3, 4}, {5, 6, 7} and {8}

Table 5.1: Correlated variables of the simulated data of eight variables in Example 5.2.1 based on different methods.

A time series plots of the simulated data is shown below in Figure 5.1. A window with size 101 is used in the calculation of the moving cross-correlation function. The outcomes of the maximum moving cross-correlation method are reported and validated with those of moving cross-correlation method from Example 3.1.2; See Table 5.1. Both methods were able to capture the dynamic relationship among different simulated variables where 3 uncorrelated subseries of 4, 3 and 1 variable are suggested.

5.3 Theoretical Properties

For the GTS-PCA's model in equation (5.1), we shall show that the transformation matrix $\hat{\mathbf{A}}$ is a consistence estimator of \mathbf{A} . To this end, we are going to adapt the distance measurement defined in (3.20) in Chapter 3. Recall, for some constants $c_1 < c_2$, let \mathbf{B}_1 and \mathbf{B}_2 be any $c_2 \times (c_2 - c_1)$ matrices satisfying the condition $\mathbf{B}'_i \mathbf{B}_i = \mathbf{I}_{(c_2 - c_1) \times (c_2 - c_1)}$, where $i \in \{1, 2\}$, then the distance between the \mathbf{B}_1 and \mathbf{B}_2 is

$$D(\mathbf{B}_1, \mathbf{B}_2) = \sqrt{c_2 - c_1 - \text{tr}(\mathbf{B}_1 \mathbf{B}'_1 \mathbf{B}_2 \mathbf{B}'_2)}.$$

In particular, let $W = 2w + 1$ be the window size, then we shall approach the consistency by showing that $D(\mathcal{M}(\hat{\mathbf{A}}), \mathcal{M}(\mathbf{A})) \rightarrow 0$ as $W \rightarrow \infty$, where $\mathcal{M}(\mathbf{A})$ is the linear space spanned by \mathbf{A} 's columns and $D(\mathcal{M}(\hat{\mathbf{A}}), \mathcal{M}(\mathbf{A}))$ is the distance between the spaces $\mathcal{M}(\hat{\mathbf{A}})$ and $\mathcal{M}(\mathbf{A})$.

The convergence of $\hat{\mathbf{A}}$ can be implied mainly by the convergence of $\hat{\mathbf{M}}\Gamma_z(l)$, which is used to calculate $\hat{\mathbf{M}}\mathbf{W}_z$ to finally formulate $\hat{\mathbf{U}}_z$. Recall that $\hat{\mathbf{U}}_z$ is equivalent to $\hat{\mathbf{A}}$ after appropriate arrangement of its columns. Hence, we shall assume that $\hat{\mathbf{U}}_z$ has a correct columns rearrangement and therefore $\hat{\mathbf{U}}_z = \hat{\mathbf{A}}$. This assumption is justified as the order of the columns will not affect the convergence speed. In this section, we are going to show the consistency of our estimator when m is fixed and when the growth rate of m depends on W . Also, we are going to use the mixing coefficient defined in (3.21) in Chapter 3 as a measurement of dependence:

$$\theta_l = \sup_c \sup_{A \in \mathcal{F}_{-\infty}^c, B \in \mathcal{F}_{c+l}^\infty} |P(A \cap B) - P(A)P(B)|.$$

For some constants c_3 and c_4 , $\mathcal{F}_{c_3}^{c_4}$ is the σ -field generated by \mathbf{z}_t for $c_3 \leq t \leq c_4$.

We are going to assume that the transformed n blocks of $\mathbf{M}\mathbf{W}_x$ do not share the same eigenvalues, otherwise these blocks should be merged as one block. In other words, denote by $\sigma(\mathbf{M}\mathbf{W}_x)$ the set of all eigenvalues of $\mathbf{M}\mathbf{W}_x$, then the minimum distance between any two eigenvalues from the different blocks $\lambda_j, \lambda_k \in \sigma(\mathbf{M}\mathbf{W}_x)$

$$d_W = \min_{1 \leq j < k \leq n} |\lambda_j - \lambda_k| \quad (5.20)$$

is positive.

5.3.1 Asymptotic Results (m is fixed)

For ease of notation, assume $\max(l, w) = w$. Next we are going to state some regularity assumptions.

Assumption 5.1. Assume

$$\sup_i \max_{1 \leq j \leq m} E|z_{j,i} - E(z_{j,i})|^{2q}$$

to be upper bounded by a positive constant M for some constant $q > 2$, where $z_{j,i}$ is the j^{th} component of \mathbf{z}_i .

Assumption 5.2. Assume that the mixing coefficients θ_l defined in (3.21) satisfies $\sum_{l=1}^{\infty} \theta_l^{1-2/q} < \infty$ for q defined in Assumption 5.1.

Assumption 5.3. Assume that different blocks of $\mathbf{M}\mathbf{W}_x$ do not share the same eigenvalues. That is $d_W > 0$, where d_W is defined in (5.20).

Theorem 5.1. Under assumptions 5.1-5.3 and for a fixed dimension m , then as $W \rightarrow \infty$, there exists $\hat{\mathbf{A}} = (\hat{\mathbf{A}}_1, \hat{\mathbf{A}}_2, \dots, \hat{\mathbf{A}}_n)$ such that

$$\max_{1 \leq j \leq n} D(\mathcal{M}(\hat{\mathbf{A}}_j), \mathcal{M}(\mathbf{A}_j)) = O_p(1/\sqrt{W}).$$

In order to prove Theorem 5.1, the following lemmas are needed. The following lemma is based on Lemma 3.1 from Chapter 3.

Lemma 5.1. *Under the assumptions 5.1 and 5.2, and by assuming the dimension m is fixed, then $\|\hat{\Gamma}_{z,i}(l) - \Gamma_{z,i}(l)\|_2 = O_p(1/\sqrt{W})$ for all $l \leq l_1$ as $W \rightarrow \infty$, where $\Gamma_{z,i}(l)$ and $\hat{\Gamma}_{z,i}(l)$ are defined in (5.2) and (5.3), respectively.*

Proof of Lemma 5.1. Under the assumptions 5.1 and 5.2, then by applying Lemma 3.1 on each window we have

$$\max_{w+1 \leq i \leq T-w} \|\hat{\Gamma}_{z,i}(l) - \Gamma_{z,i}(l)\|_2 = O_p(1/\sqrt{W})$$

for all $l \leq l_1$. This also implied that

$$\|\hat{\Gamma}_{z,i}(l) - \Gamma_{z,i}(l)\|_2 = O_p(1/\sqrt{W}).$$

Lemma 5.2. *Under assumptions 5.1-5.2, then*

$$\|\hat{\mathbf{M}}\Gamma_z(l) - \mathbf{M}\Gamma_z(l)\|_2 = O_p(1/\sqrt{W})$$

for all $l \leq l_1$ as $W \rightarrow \infty$, where $\mathbf{M}\Gamma_z(l)$ and $\hat{\mathbf{M}}\Gamma_z(l)$ are defined in (5.4) and (5.5), respectively.

Proof of Lemma 5.2. From the definitions of $\mathbf{M}\Gamma_z(l)$ and $\hat{\mathbf{M}}\Gamma_z(l)$, and by assuming 5.1 and 5.2, hold and m is fixed, then

$$\begin{aligned} \|\hat{\mathbf{M}}\Gamma_z(l) - \mathbf{M}\Gamma_z(l)\|_2 &= \left\| \frac{1}{T-2w} \sum_{i=w+1}^{T-w} \hat{\Gamma}_{z,i}(l) - \frac{1}{T-2w} \sum_{i=w+1}^{T-w} \Gamma_{z,i}(l) \right\|_2 \\ &\leq \max_{w+1 \leq i \leq T-w} \left\| \frac{T-2w}{T-2w} \hat{\Gamma}_{z,i}(l) - \frac{T-2w}{T-2w} \Gamma_{z,i}(l) \right\|_2 \\ &= O_p(1/\sqrt{W}). \end{aligned}$$

The last inequality holds because from Lemma 5.1 as

$$\max_{w+1 \leq i \leq T-w} \|\hat{\Gamma}_{z,i}(l) - \Gamma_{z,i}(l)\|_2 = O_p(1/\sqrt{W}).$$

Lemma 5.3. *Under assumptions 5.1 and 5.2, then*

$$\|\hat{\mathbf{M}}\mathbf{W}_z - \mathbf{M}\mathbf{W}_z\|_2 = O_p(1/\sqrt{W})$$

as $W \rightarrow \infty$.

Proof of Lemma 5.3. From Lemma 5.2 above, we know that

$$\|\hat{\mathbf{M}}\Gamma_z(l) - \mathbf{M}\Gamma_z(l)\|_2 = O_p(1/\sqrt{W})$$

for all $l \leq l_1$, and by using the definition of $\mathbf{M}\mathbf{W}_z$ in (5.10), then the result follows immediately.

Now, we are ready to prove Theorem 5.1.

Proof of Theorem 5.1. Let the assumptions 5.1-5.3 hold and the dimension m is fixed. By applying the results of Theorem 8.1.10 of Golub and Van Loan (1996), which are stated in Lemma 1 and Remark 1 in Chang et al. (2018), then

$$\max_{1 \leq j \leq n} D(\mathcal{M}(\hat{\mathbf{A}}_j), \mathcal{M}(\mathbf{A}_j)) = O_p(\|\hat{\mathbf{M}}\mathbf{W}_z - \mathbf{M}\mathbf{W}_z\|_2)$$

Then, by using the results of Lemma 5.3 above, we have

$$\max_{1 \leq j \leq n} D(\mathcal{M}(\hat{\mathbf{A}}_j), \mathcal{M}(\mathbf{A}_j)) = O_p(1/\sqrt{W})$$

as $W \rightarrow \infty$.

5.3.2 Asymptotic Results ($m = o(W^c)$)

As shown earlier, when the dimension m is fixed, the sample moving cross-covariance matrix $\hat{\mathbf{M}}\Gamma_z(l)$ defined in (5.5) is a consistent estimator with $W^{-1/2}$ convergence rate. However, for the case where m is large, and its growth rate depends on W , then $\hat{\mathbf{M}}\Gamma_z(l)$ is no longer consistent. This is because each of the sample cross-covariance matrices used on the local windows is not consistent; See Chen et al. (2013). One solution, in this case, is to use a threshold estimator; See Bickel and Levina (2008). For GTS-PCA, the threshold estimator can be applied to the sample cross-covariance matrix $\hat{\Gamma}_{z,i}(l)$ in order to obtain a consistent estimator for $\Gamma_{z,i}(l)$. The obtained new matrices then can be used in the calculation of the sample moving cross-covariance matrix.

In specific, write the sample lag l cross-covariance matrix over window \mathbf{z}_i defined in (5.3) as $\hat{\Gamma}_{z,i}(l) = (\hat{s}_{j,k}^{(i)}(l))_{1 \leq j,k \leq m}$. Write the sample lag l moving cross-covariance matrix defined in (5.5) as $\hat{\mathbf{M}}\Gamma_z(l) = (\hat{\sigma}_{j,k}(l))_{1 \leq j,k \leq m}$. Then, the threshold estimator of the sample cross-covariance matrix of window \mathbf{z}_i is defined as

$$T_u(\hat{\Gamma}_{z,i}(l)) = (\hat{s}_{j,k}^{(i)}(l) \mathbb{I}\{|\hat{s}_{j,k}^{(i)}(l)| \geq u\})_{1 \leq j,k \leq m}. \quad (5.21)$$

When the threshold level u is chosen to be $u = cm^{2/\gamma} W^{-1/2}$, and under some regularity conditions for a specified constant γ and a positive constant c , then $\max_{j,k} |\hat{s}_{j,k}^{(i)}(l) - s_{j,k}^{(i)}(l)| = O_p(m^{2/\gamma} W^{-1/2})$; See Chang et al. (2018). Notice that $\hat{s}_{j,k}^{(i)}(l)$ now are the elements of the threshold estimator defined in (5.21).

The sample lag l moving cross-covariance matrix of \mathbf{z}_t defined in (5.5) can be redefined as

$$\hat{\mathbf{M}}\Gamma_z^{\text{thre}}(l) = \frac{1}{T - 2 \max(l, w)} \sum_{i=\max(l,w)+1}^{T-\max(l,w)} T_u(\hat{\Gamma}_{z,i}(l)). \quad (5.22)$$

$\hat{\mathbf{M}}\Gamma_z^{\text{thre}}(l)$ in this case will be a consistent estimator of $\mathbf{M}\Gamma_z(l)$; See Lemma 5.5. Also, redefine the sample quadratic order moving cross-covariance matrix of \mathbf{z}_t in (5.10) to be

$$\hat{\mathbf{M}}\mathbf{W}_z^{\text{thre}} = \mathbf{I}_m + \sum_{l=1}^{l_1} \hat{\mathbf{M}}\Gamma_z^{\text{thre}}(l) \cdot [\hat{\mathbf{M}}\Gamma_z^{\text{thre}}(l)]'. \quad (5.23)$$

Therefore, the calculation procedure of GTS-PCA in the case, where $m = o(W^c)$, will remain the same with replacing $\hat{\mathbf{M}}\mathbf{W}_z$ by $\hat{\mathbf{M}}\mathbf{W}_z^{\text{thre}}$, as defined in (5.23). Under stronger versions of assumptions 5.1 and 5.2, the consistency of $\hat{\mathbf{M}}\mathbf{W}_z^{\text{thre}}$ will be shown in this section.

Define the quadratic order cross-covariance matrix over window \mathbf{z}_i to be

$$\mathbf{W}\Gamma_{z,i} = \mathbf{I}_m + \sum_{l=1}^{l_1} \Gamma_{z,i}(l) \cdot [\Gamma_{z,i}(l)]' \quad (5.24)$$

which can be estimated by

$$\hat{\mathbf{W}}\Gamma_{z,i}^{\text{thre}} = \mathbf{I}_m + \sum_{l=1}^{l_1} T_u(\hat{\Gamma}_{z,i}(l)) \cdot [T_u(\hat{\Gamma}_{z,i}(l))]' \quad (5.25)$$

Also, let $\mathbf{A}^{(i)}$ be the matrix that consists of the eigenvectors of $\mathbf{W}\Gamma_{z,i}$ defined in (5.24), where $\mathbf{A}^{(i)} = (a_{j,k}^{(i)})_{1 \leq j,k \leq m}$. In the stationary TS-PCA, $\mathbf{A}^{(i)}$ is used to transform an m -dimensional series into uncorrelated r subseries, where the window \mathbf{z}_i consists of all data in the series (i.e. only one window is used as $W = T$). Denote by

$$B = \max_{1 \leq k \leq r} (m_k) \quad (5.26)$$

the maximum dimension among the r different subseries transformed by $\mathbf{A}^{(i)}$ after columns' rearrangement.

In what follows, we will show the consistency of $\hat{\mathbf{A}}$ based on the threshold estimators, as defined in (5.21), (5.22) and (5.23). In particular, we shall state the following assumptions.

Assumption 5.4. Assume for each window \mathbf{z}_i that

$$\max_{1 \leq k \leq m} \sum_{j=1}^m |a_{j,k}^{(i)}|^\delta \leq c_5 \quad (5.27)$$

and

$$\max_{1 \leq j \leq m} \sum_{k=1}^m |a_{j,k}^{(i)}|^\delta \leq c_6$$

for an arbitrary constant $\delta \in [0, 1)$ and $\{c_5, c_6\} > 0$, where c_5 and c_6 are allowed to diverge away with m .

Assumption 5.4 controls the sparsity of the matrix $\mathbf{A}^{(i)}$ through δ , c_5 and c_6 when the dimension m is large. $\hat{\mathbf{A}}^{(i)}$ converges to $\mathbf{A}^{(i)}$ faster when its sparsity level increases; See Chang et al. (2018).

Assumption 5.5. Assume that as $g \rightarrow \infty$,

$$\sup_i \max_{1 \leq j \leq m} P(|z_{j,i} - E(z_{j,i})| > g) = O(g^{-2(\gamma+\varepsilon)})$$

for constants $\gamma > 2$ and $\varepsilon > 0$.

Assumption 5.5 is a stronger version of Assumption 5.1, where the tail probabilities of \mathbf{z}_i is required to decay faster than those in Assumption 5.1.

Assumption 5.6. Assume that as $l \rightarrow \infty$,

$$\theta_l = O(l^{-\gamma(\gamma+\varepsilon)/2\varepsilon})$$

where θ_l is the mixing coefficients defined in (3.21), and the constants γ and ε are provided in Assumption 5.5.

Assumption 5.6 requires the mixing coefficients to decay faster than those in Assumption 5.2. Before stating the last assumption, consider the following notations. Let

$$f = Bc_5c_6 \tag{5.28}$$

where c_5 , c_6 are given in Assumption 5.4 and B is given in (5.26). Let

$$d_k = \min_{1 \leq j \leq n, \lambda_j \neq \lambda_k} |\lambda_j - \lambda_k| \tag{5.29}$$

for $1 \leq k \leq n$, and λ_j and λ_k are the eigenvalues of the j^{th} and k^{th} blocks of \mathbf{MW}_x , respectively. Let

$$h = \max_{1 \leq l \leq l_1} \|\mathbf{M}\Gamma_x(l)\|_2 \tag{5.30}$$

Assumption 5.7. Assume that $d_k > 0$ for $1 \leq k \leq n$, where d_k is defined in (5.29).

Theorem 5.2. Let the assumptions 5.4-5.7 hold. Then the following result holds for $m = o(W^{\gamma/4})$. There exists $\hat{\mathbf{A}} = (\hat{\mathbf{A}}_1, \hat{\mathbf{A}}_2, \dots, \hat{\mathbf{A}}_n)$ such that

$$\max_{1 \leq k \leq n} d_k D(\mathcal{M}(\hat{\mathbf{A}}_k), \mathcal{M}(\mathbf{A}_k)) = O_p(h(m^{4/\gamma} W^{-1})^{(1-\delta)/2} f + (m^{4/\gamma} W^{-1})^{(1-\delta)} f^2)$$

as $W \rightarrow \infty$.

We are going to prove Theorem 5.2 based on a few lemmas. Recall, the estimators in (5.21), (5.22) and (5.23) are used instead of the ones in (5.3), (5.5) and (5.10), respectively, where the threshold level is set to $u = cm^{2/\gamma} W^{-1/2}$. The following lemma of Chang et al. (2018) is useful.

Lemma 5.4. For an m -dimensional series \mathbf{y}_t , let $\gamma > 2$ and $\varepsilon > 0$ be arbitrary constants. Assume that

$$\sup_t \max_{1 \leq j \leq m} P(|y_{j,t} - \mu_j| > g) = O(g^{-2(\gamma+\varepsilon)})$$

as $g \rightarrow \infty$. Also, assume that the mixing coefficients

$$\theta_l = \sup_j \sup_{A \in \mathcal{F}_{-\infty}^j, B \in \mathcal{F}_{j+l}^\infty} (|P(A \cap B) - P(A)P(B)|)$$

is such that $\theta_l = O(l^{-\gamma(\gamma+\varepsilon)/(2\varepsilon)})$ as $l \rightarrow \infty$. Let Assumption 5.5 holds. Then, given that $m = o(T^{\gamma/4})$ and $u = cm^{2/\gamma} T^{-1/2}$, we have

$$\|T_u(\hat{\Sigma}_y(l)) - \Sigma_y(l)\|_2 = O_p((m^{4/\gamma} T^{-1})^{(1-\delta)/2} f)$$

for all $l \leq l_1$, where $\Sigma_y(l)$ is the cross-covariance matrix and T is the sample size.

Therefore we have the following lemma based on the results of Lemma 5.4 above.

Lemma 5.5 Let the assumptions 5.4-5.6 hold. Assume that $m = o(W^{\gamma/4})$. Then we have

$$\|\hat{\mathbf{M}}\Gamma_z^{\text{thre}}(l) - \mathbf{M}\Gamma_z(l)\|_2 = O_p((m^{4/\gamma} W^{-1})^{(1-\delta)/2} f)$$

for all $l \leq l_1$, where f is defined in (5.28).

Proof of Lemma 5.5 Under the assumptions 5.4-5.6, if we apply Lemma 5.4 on each window, where $m = o(W^{\gamma/4})$, then we have

$$\|T_u(\hat{\Gamma}_{z,i}(l)) - \Gamma_{z,i}(l)\|_2 = O_p((m^{4/\gamma}W^{-1})^{(1-\delta)/2}f).$$

By following similar argument to the proof of Lemma 5.2, then the proof of this lemma is complete.

Lemma 5.6. *Let the assumptions 5.4-5.6 hold. Assume further that $m = o(W^{\gamma/4})$. Then it holds that*

$$\|\hat{\mathbf{M}}\mathbf{W}_z^{\text{thre}} - \mathbf{M}\mathbf{W}_z\|_2 = O_p(h(m^{4/\gamma}W^{-1})^{(1-\delta)/2}f + (m^{4/\gamma}W^{-1})^{(1-\delta)}f^2)$$

where f and h are defined in (5.28) and (5.30), respectively.

Proof of Lemma 5.6. Since

$$\begin{aligned} \|\hat{\mathbf{M}}\mathbf{W}_z^{\text{thre}} - \mathbf{M}\mathbf{W}_z\|_2 &= \|\hat{\mathbf{M}}\mathbf{\Gamma}_z^{\text{thre}}(l) [\hat{\mathbf{M}}\mathbf{\Gamma}_z^{\text{thre}}(l)]' - \mathbf{M}\mathbf{\Gamma}_z(l) \mathbf{M}\mathbf{\Gamma}_z'(l)\|_2 \\ &\leq 2\|\hat{\mathbf{M}}\mathbf{\Gamma}_z^{\text{thre}}(l) - \mathbf{M}\mathbf{\Gamma}_z(l)\|_2 \|\mathbf{M}\mathbf{\Gamma}_z(l)\|_2 + \|\hat{\mathbf{M}}\mathbf{\Gamma}_z^{\text{thre}}(l) - \mathbf{M}\mathbf{\Gamma}_z(l)\|_2^2 \\ &\leq 2\|\hat{\mathbf{M}}\mathbf{\Gamma}_z^{\text{thre}}(l) - \mathbf{M}\mathbf{\Gamma}_z(l)\|_2 \|\mathbf{M}\mathbf{\Gamma}_x(l)\|_2 + \|\hat{\mathbf{M}}\mathbf{\Gamma}_z^{\text{thre}}(l) - \mathbf{M}\mathbf{\Gamma}_z(l)\|_2^2. \end{aligned}$$

Recall that $h = \max_{1 \leq l \leq l_1} \|\mathbf{M}\mathbf{\Gamma}_x(l)\|_2$, then by applying the results from Lemma 5.5, we complete the proof as

$$\|\hat{\mathbf{M}}\mathbf{W}_z^{\text{thre}} - \mathbf{M}\mathbf{W}_z\|_2 = O_p(h(m^{4/\gamma}W^{-1})^{(1-\delta)/2}f + (m^{4/\gamma}W^{-1})^{(1-\delta)}f^2).$$

Now, we are ready to prove Theorem 5.2.

Proof of Theorem 5.2. Under the assumptions 5.4-5.7. Assume further that $m = o(W^{\gamma/4})$. Then by applying the results of Theorem 8.1.10 of Golub and Van Loan (1996), we have

$$\max_{1 \leq k \leq n} d_k D(\mathcal{M}(\hat{\mathbf{A}}_k), \mathcal{M}(\mathbf{A}_k)) = O_p(\|\hat{\mathbf{M}}\mathbf{W}_z^{\text{thre}} - \mathbf{M}\mathbf{W}_z\|_2).$$

By Lemma 5.6, then we have

$$\max_{1 \leq k \leq n} d_k D(\mathcal{M}(\hat{\mathbf{A}}_k), \mathcal{M}(\mathbf{A}_k)) = O_p(h(m^{4/\gamma}W^{-1})^{(1-\delta)/2}f + (m^{4/\gamma}W^{-1})^{(1-\delta)}f^2).$$

5.3.3 Asymptotic Results ($\log m = o(W^c)$)

In this section, we are going to study the asymptotic properties of the transformation matrix $\hat{\mathbf{A}}$ in the case where the growth rate of m has an exponential relationship with W . Recall, the sample cross-covariance matrices that are calculated over the local windows will not be consistent when the dimension m is large. Therefore, we shall apply a threshold estimator to $\Gamma_z(l)$ as we are considering series with extremely high dimensions in this section. In particular, the threshold estimator $T_u(\hat{\Gamma}_{z,i}(l))$ defined in (5.21) will be used, but with a different threshold level than the previously used. Chang et al. (2018) showed that when $\log m = o(W^c)$, under some regularity assumptions, the threshold estimator of the cross-covariance matrix with a threshold level $u = c(W^{-1} \log m)^{1/2}$ is consistent. This result can be applied to $T_u(\hat{\Gamma}_{z,i}(l))$ over the local windows to calculate $\hat{\mathbf{M}}\Gamma_z^{\text{thre}}(l)$ as follows.

Assumption 5.8. Assume for any $g > 0$ and $\|\mathbf{y}\|_2 = 1$ that

$$\sup_i P(|\mathbf{y}'(\mathbf{z}_i - E(\mathbf{z}_i))| > g) \leq c_7 \exp(-c_8 g^\eta)$$

where the constants c_7, c_8 are positive and $\eta \in (0, 2]$.

This assumption implies that the tail probabilities of linear combinations of \mathbf{z}_i decays exponentially.

Assumption 5.9. Assume that

$$\theta_l \leq \exp(-c_9 l^\xi)$$

for all $l \geq 0$, where $\xi \in (0, 1]$ and c_9 is a positive constant.

This assumption requires the mixing coefficient to decay exponentially.

Theorem 5.3. Suppose assumptions 5.4, 5.7, 5.8 and 5.9 hold. Let $\log m = o(W^{(b/(2-b))})$, where $b = 1/(2\eta^{-1} + \xi^{-1})$. Then there exists $\hat{\mathbf{A}} = (\hat{\mathbf{A}}_1, \hat{\mathbf{A}}_2, \dots, \hat{\mathbf{A}}_n)$ such that

$$\max_{1 \leq k \leq n} d_k D(\mathcal{M}(\hat{\mathbf{A}}_k), \mathcal{M}(\mathbf{A}_k)) = O_p(fh(W^{-1} \log m)^{(1-\delta)/2} + f^2(W^{-1} \log m)^{(1-\delta)})$$

as $W \rightarrow \infty$, where f and h are constants specified in (5.28) and (5.30), respectively.

The following lemmas are needed to prove Theorem 5.3. Recall that we are using the estimators in (5.21), (5.22) and (5.23) but with the threshold level $u = c(W^{-1} \log m)^{1/2}$.

Lemma 5.7. *Let assumptions 5.4, 5.8 and 5.9 hold. Let $\log m = o(W^{b/(2-b)})$ where $b = 1/(2\eta^{-1} + \xi^{-1})$. Then*

$$\|\hat{\mathbf{M}}\Gamma_z^{\text{thre}}(l) - \mathbf{M}\Gamma_z(l)\|_2 = O_p(f(W^{-1} \log m)^{(1-\delta)/2}).$$

Proof of Lemma 5.7. Notice that $\log m = o(W^{b/(2-b)})$ and $u = c(W^{-1} \log m)^{1/2}$. Under the assumptions 5.4, 5.8 and 5.9, if we apply Lemma 5.4 on each window, then it follows

$$\|T_u(\hat{\Gamma}_{z,i}(l)) - \Gamma_{z,i}(l)\|_2 = O_p(f(W^{-1} \log m)^{(1-\delta)/2}).$$

The rest of the proof can be obtained by following similar argument to the proof of Lemma 5.2.

Lemma 5.8. *Let assumptions 5.4, 5.8 and 5.9 hold. Then for $\log m = o(W^{b/(2-b)})$, it follows*

$$\|\hat{\mathbf{M}}\mathbf{W}_z^{\text{thre}} - \mathbf{M}\mathbf{W}_z\|_2 = O_p(fh(W^{-1} \log m)^{(1-\delta)/2} + f^2(W^{-1} \log m)^{(1-\delta)})$$

where $b = 1/(2\eta^{-1} + \xi^{-1})$, f and h are constants as provided in (5.28) and (5.30), respectively.

Proof of Lemma 5.8. Using the results from Lemma 5.7, the proof of this lemma can be obtained by following similar steps to the proof of Lemma 5.6.

Proof of Theorem 5.3. Under assumptions 5.4, 5.7, 5.8 and 5.9. Let $\log m = o(W^{b/(2-b)})$ where $b = 1/(2\eta^{-1} + \xi^{-1})$. Theorem 8.1.10 of Golub and Van Loan (1996) yields

$$\max_{1 \leq k \leq n} d_k D(\mathcal{M}(\hat{\mathbf{A}}_k), \mathcal{M}(\mathbf{A}_k)) = O_p(\|\hat{\mathbf{M}}\mathbf{W}_z^{\text{thre}} - \mathbf{M}\mathbf{W}_z\|_2).$$

Then, by using Lemma 5.8, we complete the proof as

$$\max_{1 \leq k \leq n} d_k D(\mathcal{M}(\hat{\mathbf{A}}_k), \mathcal{M}(\mathbf{A}_k)) = O_p(fh(W^{-1} \log m)^{(1-\delta)/2} + f^2(W^{-1} \log m)^{(1-\delta)}).$$

5.4 Simulations and Real Data Examples

In the following studies, the numerical properties of GTS-PCA will be investigated. Our method will be tested on both simulated and real data sets with various dimensions and sample sizes. All analyses are carried out using R software. The necessary functions to produce and assess the results of GTS-PCA can be found under the following R packages: `GTSPCA` of Alshammri (2020a) and `MpermutMax` of Alshammri (2020e).

5.4.1 Simulations

The simulations below are generated using the following steps. First, we generate the subseries of \mathbf{x}_t separately using the `arma.sim` command. Then, the elements of the transformation matrix \mathbf{A} are generated randomly from $unif(-5, 5)$ using the `runif` command. Post multiplying \mathbf{x}_t by \mathbf{A} will generate \mathbf{z}_t as in (5.1). GTS-PCA with different window sizes $W = \{101, 201, 501\}$ and TS-PCA (i.e. GTS-PCA with $W = T$) will be applied to simulated data. Then, the percentage of correct segmentation will measure the accuracy of the results, where “correct segmentation” refer to the case where the final transformed series is correctly rearranged into uncorrelated subseries as simulated in \mathbf{x}_t at the first step of the simulation process. Following Remark 1 of Chang et al. (2018), the number of lags l_1 to include in the calculation of (5.12) could be small as long as the first l_1 contain sufficient amount of information on the block diagonal structure of $\mathbf{M}\mathbf{W}_x$. Therefore, for demonstration purposes, a choice of five-time lags (i.e. $l_1 = 5$) will be used in the analyses unless stated otherwise.

Example 5.4.1 The first simulation study includes a non-stationary time series \mathbf{z}_t with a dimension of $m = 5$. Three different models are used to generate the subgroups (i.e. subseries) of this series, which consists of 2, 2 and 1 independent variables.

The subgroups of \mathbf{x}_t are simulated as follows. Let a_t , b_t and c_t be independent

standard normal white noises, which are the innovation terms of the following three models, respectively, then:

$$\begin{cases} x_{j,t} = u_{t+j-1}, & j = 1 \text{ and } 2 \\ x_{j,t} = v_{t+j-3}, & j = 3 \text{ and } 4 \\ x_{j,t} = w_t, & j = 5 \end{cases} \quad (5.31)$$

where $(u_t, v_t, w_t)'$ satisfies

$$\begin{cases} \nabla u_t = 0.75 \nabla u_{t-1} + a_t + 1.5a_{t-1} + 0.5a_{t-2} + 0.2a_{t-3} \\ \nabla v_t = -0.5 \nabla v_{t-1} - 0.4 \nabla v_{t-2} + b_t - 1.2b_{t-1} - 2.75b_{t-2} + 0.5b_{t-3} \\ w_t = -0.9w_{t-1} - 0.45w_{t-2} + c_t - 1.1c_{t-1} - 0.8c_{t-2} - 0.6c_{t-3} - 1.3c_{t-4} \end{cases} \quad (5.32)$$

Then, we post multiply \mathbf{x}_t by \mathbf{A} whose elements are generated randomly from $unif(-5, 5)$ to obtain \mathbf{z}_t , as mentioned earlier. GTS-PCA with different window sizes is applied to the simulated series.

Method	GTS-PCA with W=101						
	T=300	T=400	T=500	T=600	T=1000	T=1500	T=2000
Percentage of Correct Segmentation	0.33	0.51	0.54	0.64	0.86	0.94	0.96
Method	GTS-PCA with W=201						
	T=300	T=400	T=500	T=600	T=1000	T=1500	T=2000
Percentage of Correct Segmentation	0.15	0.18	0.29	0.38	0.55	0.72	0.75
Method	GTS-PCA with W=501						
	T=300	T=400	T=500	T=600	T=1000	T=1500	T=2000
Percentage of Correct Segmentation	NA	NA	NA	0.07	0.15	0.24	0.28
Method	TS-PCA						
	T=300	T=400	T=500	T=600	T=1000	T=1500	T=2000
Percentage of Correct Segmentation	0.44	0.47	0.52	0.49	0.47	0.50	0.43

Table 5.2: Percentages of correct segmentation of 500 replications of the simulated series in Example 5.4.1.

The above simulation was replicated 500 times, and the percentage of correct segmentations are reported in Table 5.2. It can be seen that the performance of GTS-PCA was enhanced by increasing T . GTS-PCA with small window sizes (e.g. $W = 101$ and $W = 201$) produced more accurate results compared with those with larger windows (e.g. $W = 501$). This is mainly because of the non-stationarity of the data. TS-PCA, on the other hand, produced the correct segmentations about 47% of the time for different sizes of T . Hence, GTS-PCA with a small window size is more suitable to capture the hidden segmentation in this example.

GTS-PCA with $W = 101$ is of particular interest, as it outperformed those with larger window sizes. It produced reasonable results for moderate sample sizes (e.g. $T \geq 600$) in this example. For instance, it produced the correct segmentations 86% of the time when $T = 1000$, and 96% of the time when $T = 2000$.

Consider one randomly selected replica of the above simulations of \mathbf{z}_t with $T = 1000$. A time series plot of \mathbf{z}_t is shown in Figure 5.2. All five variables exhibited similar non-stationary behaviours. Sample moving cross-correlation plots with $W = 101$ between the variables of \mathbf{z}_t before transformation are shown in Figure 5.3, where all variables are correlated. Figure 5.4 provides moving cross-correlation plots with $W = 101$ between the components of \mathbf{z}_t after transformation by GTS-PCA with $W = 101$. Notice that the transformed series consisted of three uncorrelated subseries, namely $\{1, 3\}$, $\{2, 5\}$ and $\{4\}$, which revealed the hidden correct segmentation of \mathbf{x}_t with 2, 2 and 1 variable.

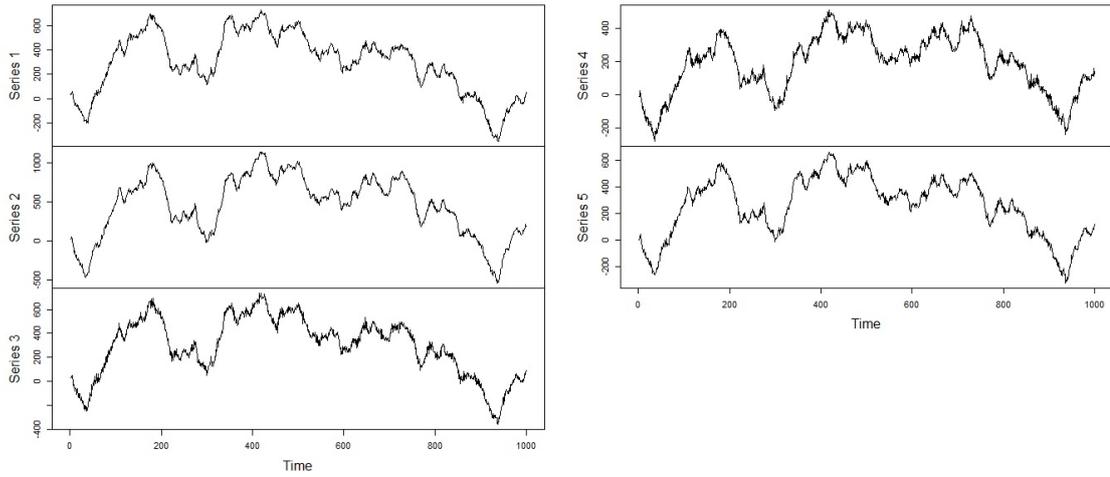


Figure 5.2: A time series plot of the randomly selected replica in Example 5.4.1.

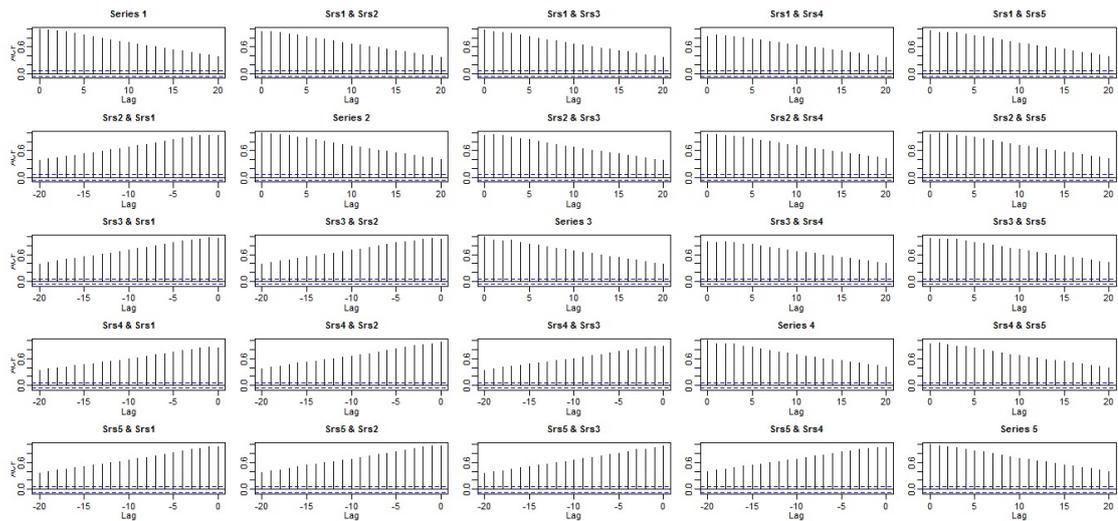


Figure 5.3: Sample moving cross-correlation plots with $W = 101$ between the components of the randomly selected replica in Example 5.4.1 before transformation.

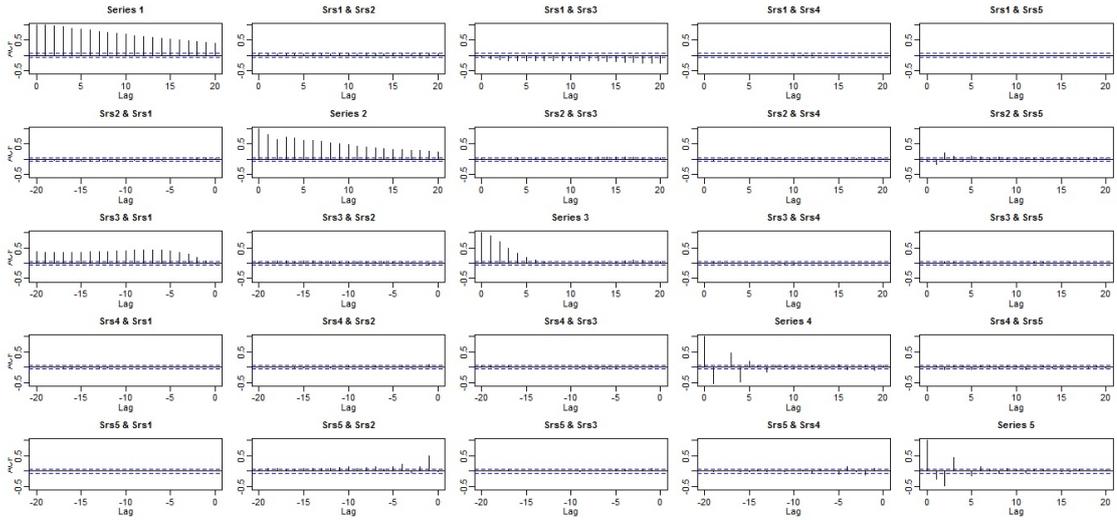


Figure 5.4: Sample moving cross-correlation plots with $W = 101$ between the components of the randomly selected replica in Example 5.4.1 after GTS-PCA with $W = 101$ is applied.

Example 5.4.2 The simulated series in this example consists of nine variables that are generated from three different models such that each model generates a subseries of three correlated variables.

The three subseries of \mathbf{x}_t are simulated as follows. Let a_t , b_t and c_t be independent standard normal white noises, which are the innovation terms of the following three models, respectively, then:

$$\begin{cases} x_{j,t} = u_{t+j-1}, & j = 1, 2 \text{ and } 3 \\ x_{j,t} = v_{t+j-4}, & j = 4, 5 \text{ and } 6 \\ x_{j,t} = w_{t+j-7}, & j = 7, 8 \text{ and } 9 \end{cases} \quad (5.33)$$

where $(u_t, v_t, w_t)'$ satisfies

$$\begin{cases} \nabla u_t = 0.4 \nabla u_{t-1} + 0.4 \nabla u_{t-2} + 0.1 \nabla u_{t-3} + a_t \\ \nabla v_t = -0.6 \nabla v_{t-1} - 0.4 \nabla v_{t-2} + 0.5 \nabla v_{t-3} + b_t - 0.9b_{t-1} - 1.4b_{t-2} - 0.4b_{t-3} \\ \nabla w_t = 0.8 \nabla w_{t-1} - 0.4 \nabla w_{t-2} + c_t + 0.8c_{t-1} + 0.9c_{t-2} + 1.7c_{t-3} \end{cases} \quad (5.34)$$

The series \mathbf{z}_t then is generated as in the previous example. Based on 500 replications, Table 5.3 shows the percentage of correct segmentation of GTS-PCA with different sizes of W and TS-PCA. It can be seen that as we increased the sample size, GTS-PCA with $W = 101$ produced more accurate results compared to those with larger window sizes (e.g. $W = 201$ and $W = 501$) and TS-PCA. For example, for \mathbf{z}_t with $T = 2000$, GTS-PCA with $W = 101$ produced the correct segmentation 85% of the time compared with 51%, and 7% when using $W = 201$ and 501, respectively. By increasing the sample size of \mathbf{z}_t to $T = 2500$, the accuracy of GTS-PCA with $W = 101$ increased from 85% to 91%. On the other hand, TS-PCA produced the correct segmentations 2% to 4% of the time for different sample sizes. Therefore, GTS-PCA with $W = 101$ can be used to reduce the dimension of \mathbf{z}_t in this example. The above results show the improvement of the accuracy of capturing the hidden segmentation of non-stationary series by using GTS-PCA compared with TS-PCA.

Method	GTS-PCA with W=101						
	T=600	T=900	T=1000	T=1500	T=2000	T=2500	T=3000
Percentage of Correct Segmentation	0.16	0.31	0.35	0.72	0.85	0.91	0.97
Method	GTS-PCA with W=201						
	T=600	T=900	T=1000	T=1500	T=2000	T=2500	T=3000
Percentage of Correct Segmentation	0.03	0.07	0.13	0.33	0.51	0.67	0.75
Method	GTS-PCA with W=501						
	T=600	T=900	T=1000	T=1500	T=2000	T=2500	T=3000
Percentage of Correct Segmentation	0.00	0.00	0.01	0.05	0.07	0.10	0.13
Method	TS-PCA						
	T=600	T=900	T=1000	T=1500	T=2000	T=2500	T=3000
Percentage of Correct Segmentation	0.04	0.02	0.04	0.02	0.03	0.02	0.03

Table 5.3: Percentages of correct segmentation of 500 replications of the simulated series in Example 5.4.2.

Consider one randomly selected replica of the 500 replicas where \mathbf{z}_t has a sample

of size 1500. A time series plot of the data is shown in Figure 5.5. Sample moving cross-correlation plots with $W = 101$ are provided to show the relationship between the variables before transformation; See figures 5.6, 5.7, 5.8 and 5.9. A strong relationship existed across almost all variables. Overall, if we want to regroup the variables based on their relationships, then one group consists of all nine variables is suggested.

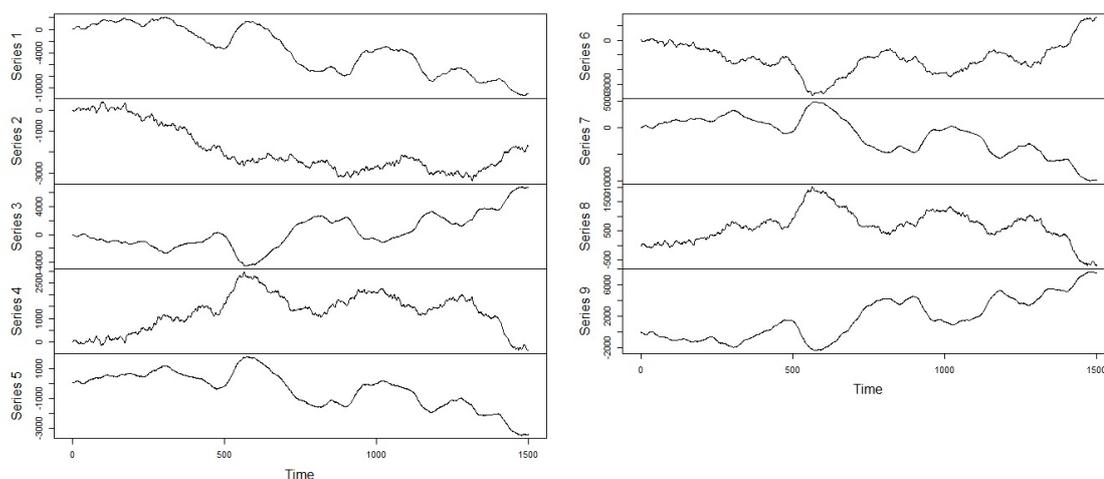


Figure 5.5: A time series plot of the randomly selected replica in Example 5.4.2.

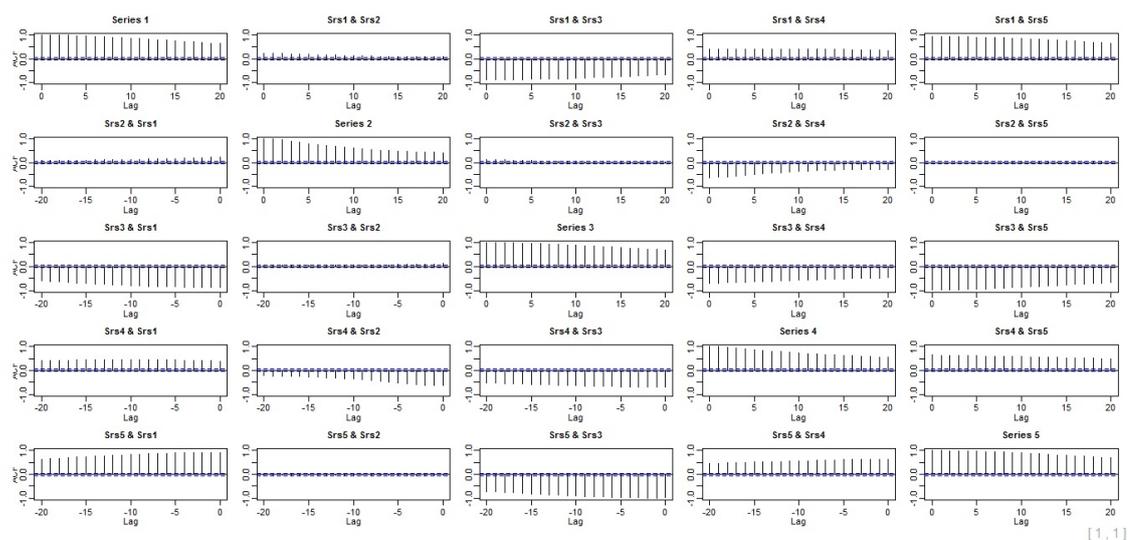


Figure 5.6: Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 before transformation - part 1.

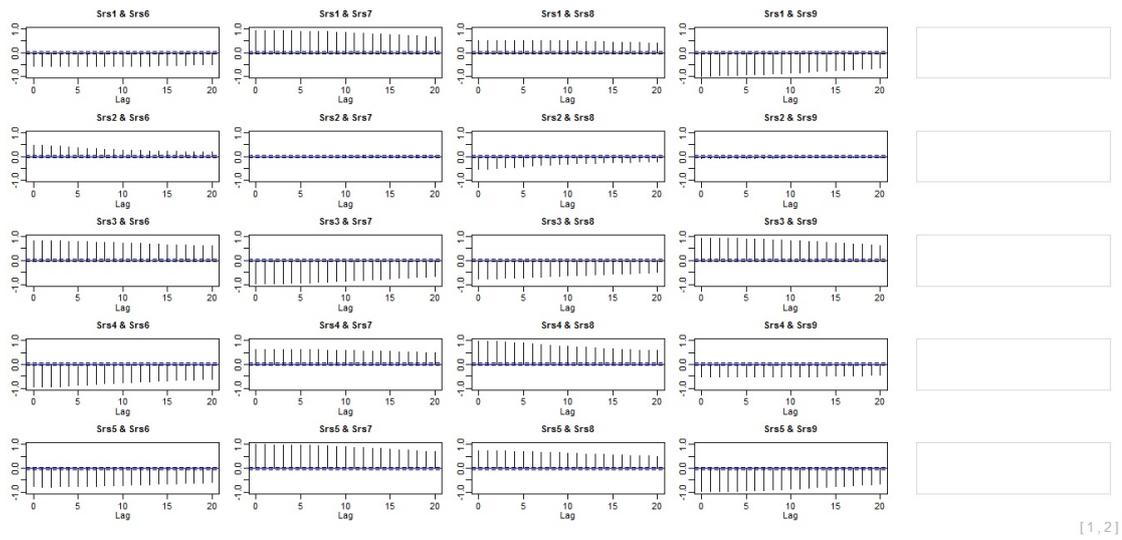


Figure 5.7: Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 before transformation - part 2.

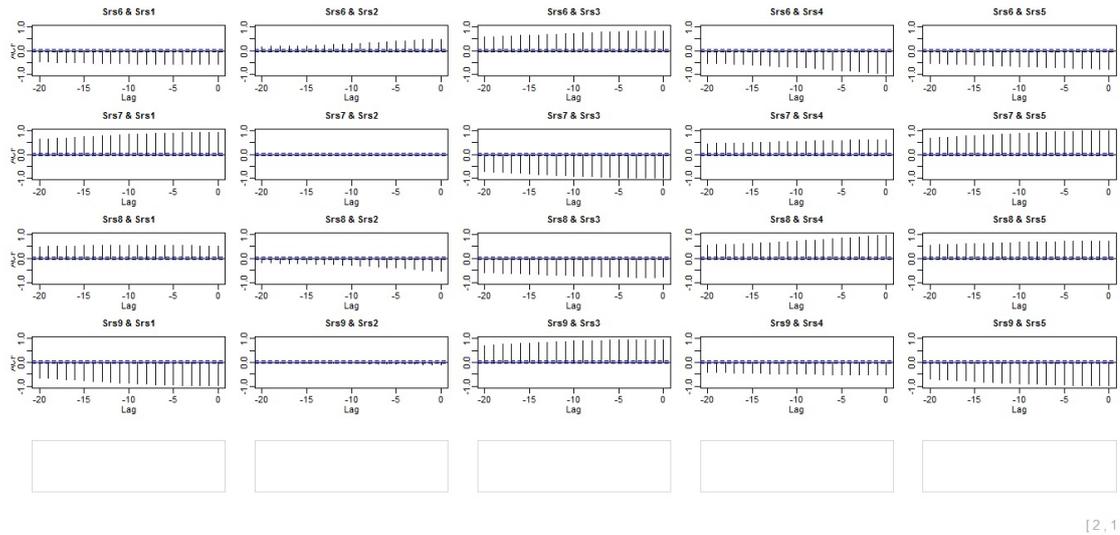
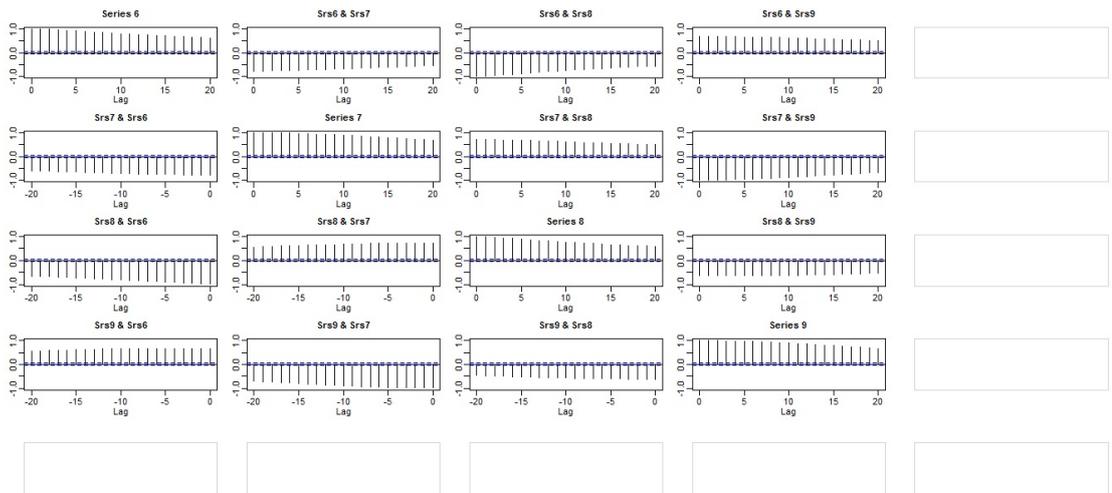


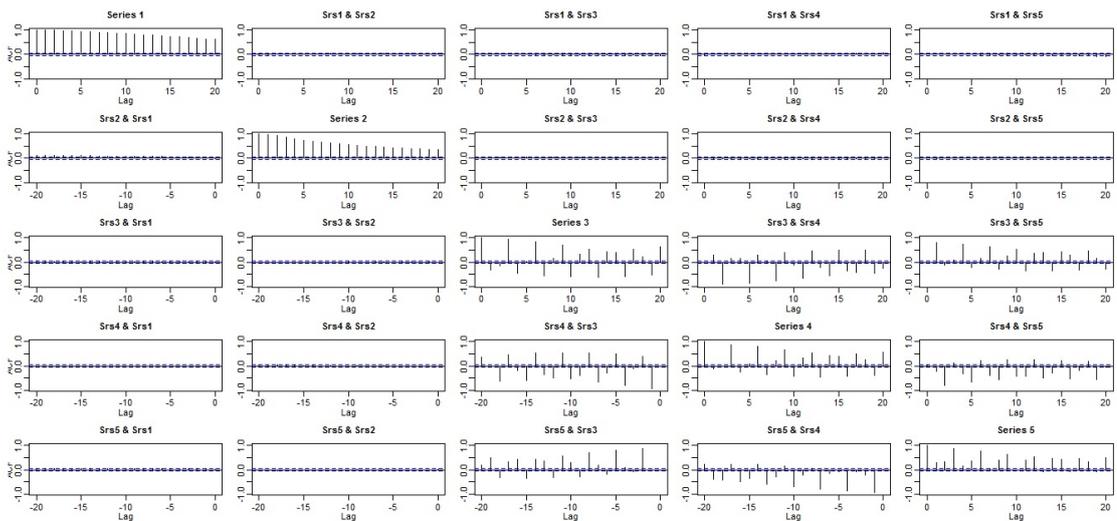
Figure 5.8: Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 before transformation - part 3.



[2, 2]

Figure 5.9: Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 before transformation - part 4.

After GTS-PCA with $W = 101$ was applied, the maximum moving cross-correlation method and sample moving cross-correlation plots with $W = 101$ suggested regrouping the transformed nine series into three subgroups, where each subgroup consisted of the following variables, respectively: $\{1, 6, 9\}$, $\{2, 7, 8\}$ and $\{3, 4, 5\}$; See figures 5.10, 5.11, 5.12 and 5.13. Therefore, we conclude that GTS-PCA with $W = 101$ revealed the hidden correct segmentation of \mathbf{x}_t with 3, 3 and 3 variable.



[1, 1]

Figure 5.10: Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 after GTS-PCA with $W = 101$ is applied - part 1.

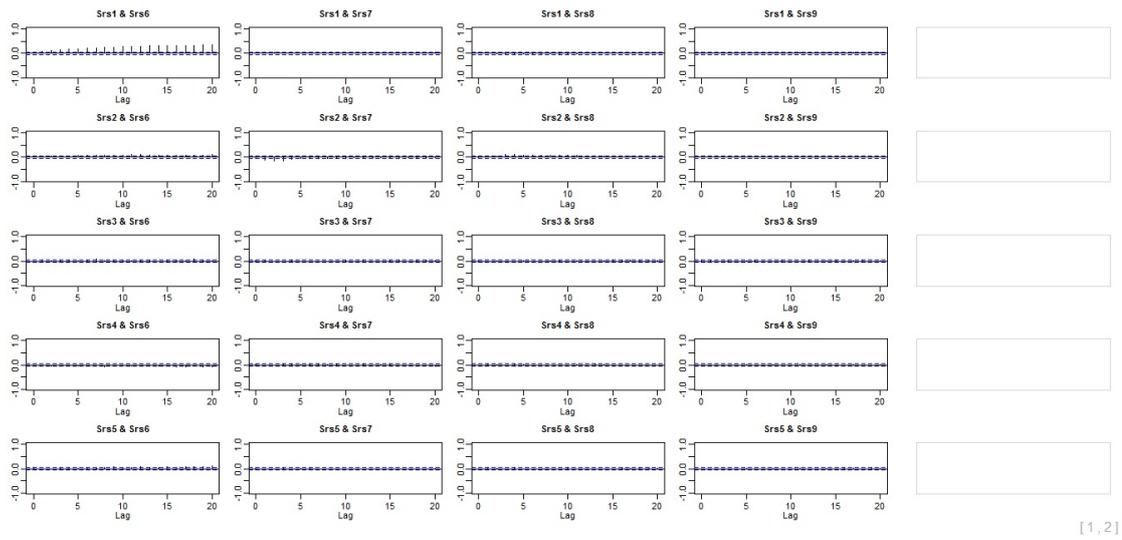


Figure 5.11: Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 after GTS-PCA with $W = 101$ is applied - part 2.

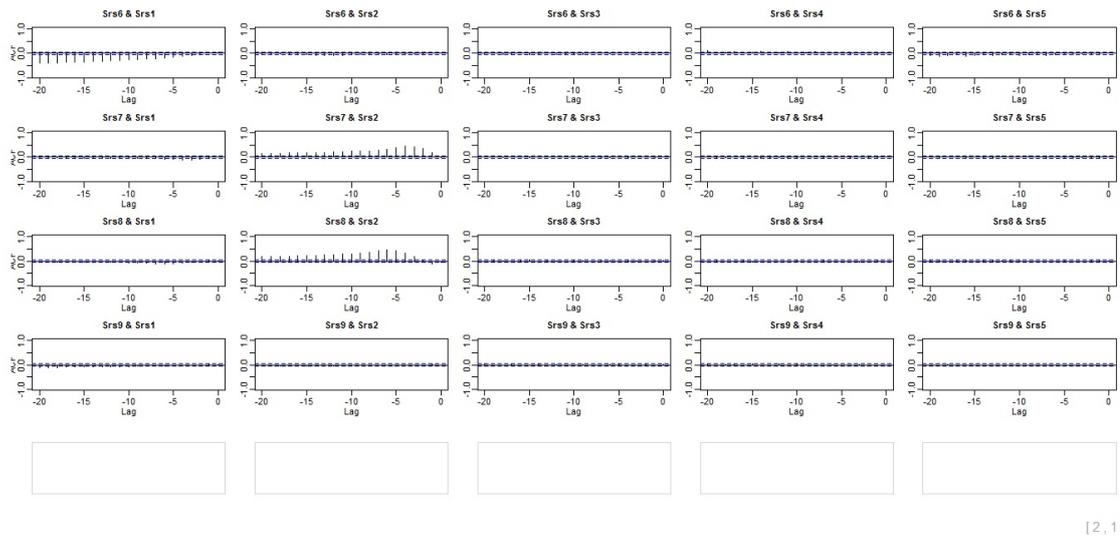
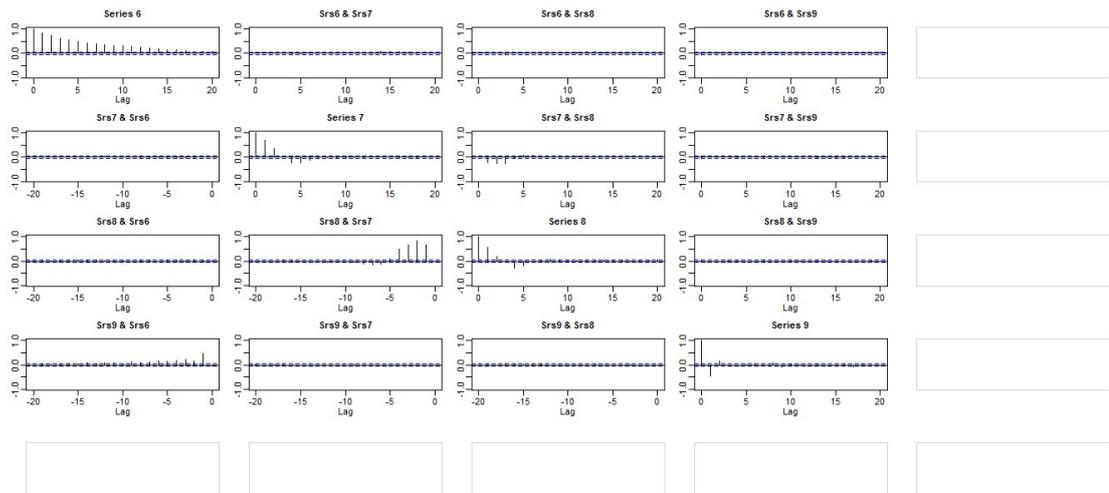


Figure 5.12: Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 after GTS-PCA with $W = 101$ is applied - part 3.



[2, 2]

Figure 5.13: Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.2 after GTS-PCA with $W = 101$ is applied - part 4.

Example 5.4.3 In this example, a non-stationary time series with a dimension of $m = 20$ is simulated such that it consists of five uncorrelated subseries with 6, 5, 4, 3 and 2 correlated variables, respectively.

The five subseries of \mathbf{x}_t are simulated as follows. Let a_t, b_t, c_t, d_t and e_t be independent standard normal white noises, which are the innovation terms of the following five models, respectively, then:

$$\left\{ \begin{array}{ll} x_{j,t} = u_{t+j-1}, & j = 1, 2, 3, 4, 5 \text{ and } 6 \\ x_{j,t} = v_{t+j-7}, & j = 7, 8, 9, 10 \text{ and } 11 \\ x_{j,t} = w_{t+j-12}, & j = 12, 13, 14 \text{ and } 15 \\ x_{j,t} = f_{t+j-16}, & j = 16, 17 \text{ and } 18 \\ x_{j,t} = g_{t+j-19}, & j = 19 \text{ and } 20 \end{array} \right. \quad (5.35)$$

where $(u_t, v_t, w_t, f_t, g_t)'$ satisfies

$$\begin{cases} \nabla u_t = 0.8 \nabla u_{t-1} + a_t + 3a_{t-1} \\ v_t = -0.9v_{t-1} - 0.4v_{t-2} + b_t - b_{t-1} - 0.8b_{t-2} - 0.6b_{t-3} - 1.4b_{t-4} \\ \nabla w_t = -0.55 \nabla w_{t-1} + c_t - 2.1c_{t-1} \\ \nabla f_t = -0.5 \nabla f_{t-1} - 0.4 \nabla f_{t-2} + d_t - 1.2d_{t-1} - 2.75d_{t-2} + 0.5d_{t-3} \\ \nabla g_t = 0.4 \nabla g_{t-1} - 0.3 \nabla g_{t-2} + e_t + 2.9e_{t-1} - 0.8e_{t-2} - 1.8e_{t-3} - 1.9e_{t-4} \end{cases} \quad (5.36)$$

Method	GTS-PCA with W=101						
	T=1500	T=2000	T=2500	T=3000	T=3500	T=4000	T=4500
Percentage of Correct Segmentation	0.13	0.35	0.59	0.78	0.84	0.92	0.95
Method	GTS-PCA with W=201						
	T=1500	T=2000	T=2500	T=3000	T=3500	T=4000	T=4500
Percentage of Correct Segmentation	0.11	0.21	0.28	0.30	0.39	0.42	0.43
Method	GTS-PCA with W=501						
	T=1500	T=2000	T=2500	T=3000	T=3500	T=4000	T=4500
Percentage of Correct Segmentation	0.01	0.10	0.12	0.13	0.15	0.16	0.17
Method	TS-PCA						
	T=1500	T=2000	T=2500	T=3000	T=3500	T=4000	T=4500
Percentage of Correct Segmentation	0.00	0.02	0.01	0.01	0.01	0.02	0.03

Table 5.4: Percentages of correct segmentation of 500 replications of the simulated series in Example 5.4.3.

The series \mathbf{z}_t then is generated as in the previous example. Based on 500 replications of the above simulation, Table 5.4 summarizes the obtained percentages of correct segmentations. The accuracy of GTS-PCA improved as the sample size is increased, especially for small window sizes, where the GTS-PCA captured more information out of the non-stationary series. For example, GTS-PCA with $W = 101$ produced the correct segmentations 78% of the time when $T = 3000$, and 95% of

the time when $T = 4500$. Furthermore, the performance of GTS-PCA decreased as we increased W . For instance, when $T = 3500$, GTS-PCA with windows sizes 101, 201 and 501 produced the correct segmentations 84%, 39% and 15% of the time, respectively. On the other hand, TS-PCA (i.e. GTS-PCA with $W = T$) had a poor performance on the simulated non-stationary series with percentages of correct segmentations less than 3 % in all cases. Therefore, GTS-PCA with $W = 101$ is suggested to reduce the dimension of the simulated series in this example.

Based on one randomly selected replica with $T = 3000$, a time series plot of \mathbf{z}_t is shown in figures 5.14 and 5.15. First and last 50 sample moving cross-correlation plots with $W = 101$ between the components of \mathbf{z}_t before transformation are shown in figures 5.16, 5.17, 5.18 and 5.19. The plots suggested that strong relationships existed among the components of \mathbf{z}_t . After applying GTS-PCA with $W = 101$ on \mathbf{z}_t , the maximum moving cross-correlation method with $W = 101$ suggested to rearrange the transformed series into five uncorrelated subgroups that consist of the following variables, respectively: $\{1, 2, 7, 11, 15, 20\}$, $\{3, 6, 9, 13, 18\}$, $\{4, 8, 14, 19\}$, $\{5, 12, 17\}$ and $\{10, 16\}$, which match the hidden correct segmentations of \mathbf{x}_t .

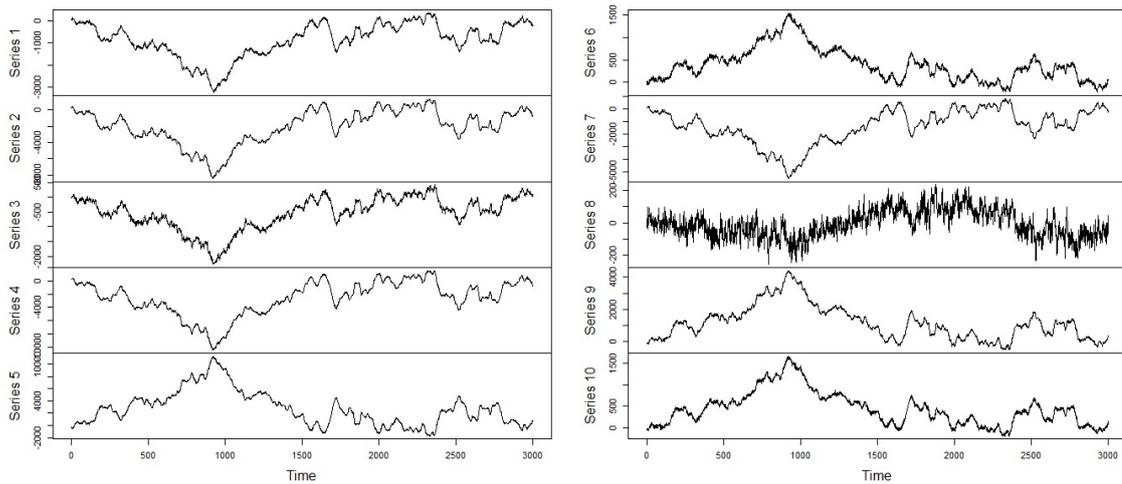


Figure 5.14: A time series plot of the variables 1 - 10 of the randomly selected replica in Example 5.4.3.

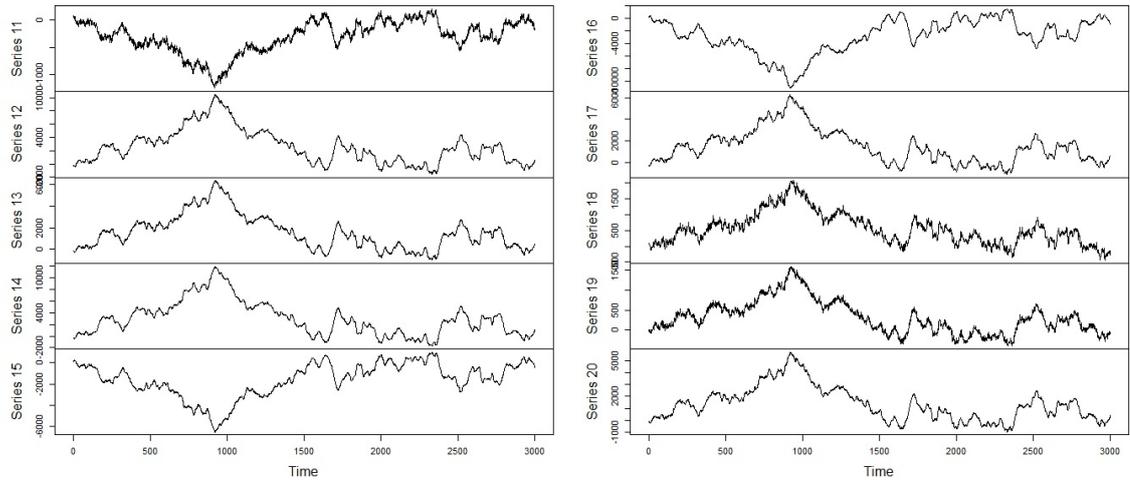


Figure 5.15: A time series plot of the variables 11 - 20 of the randomly selected replica in Example 5.4.3.

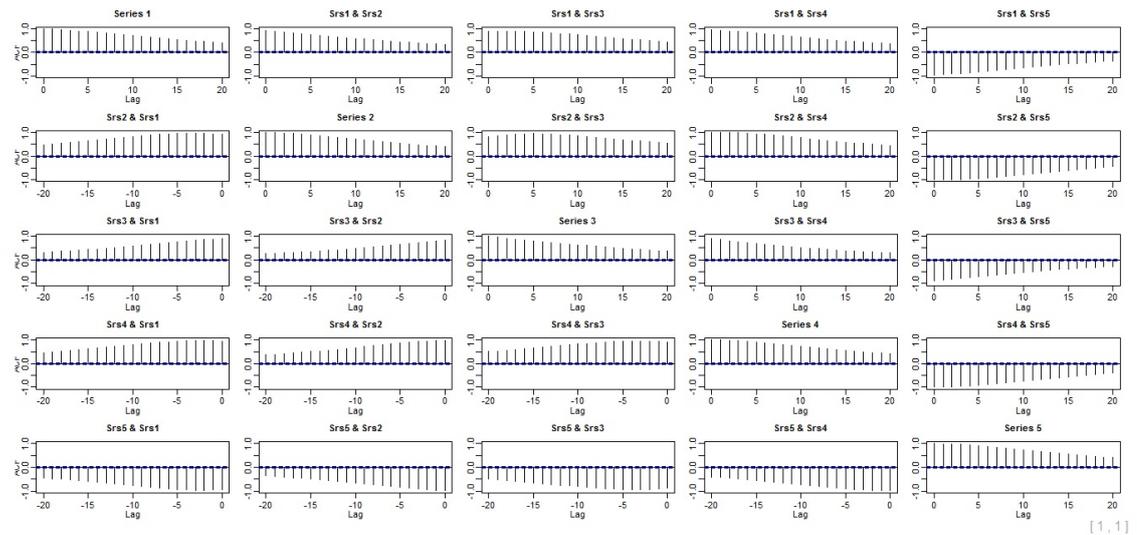


Figure 5.16: First 50 sample moving cross-correlation plots $W = 101$ of the randomly selected replica in Example 5.4.3 before transformation - part 1.

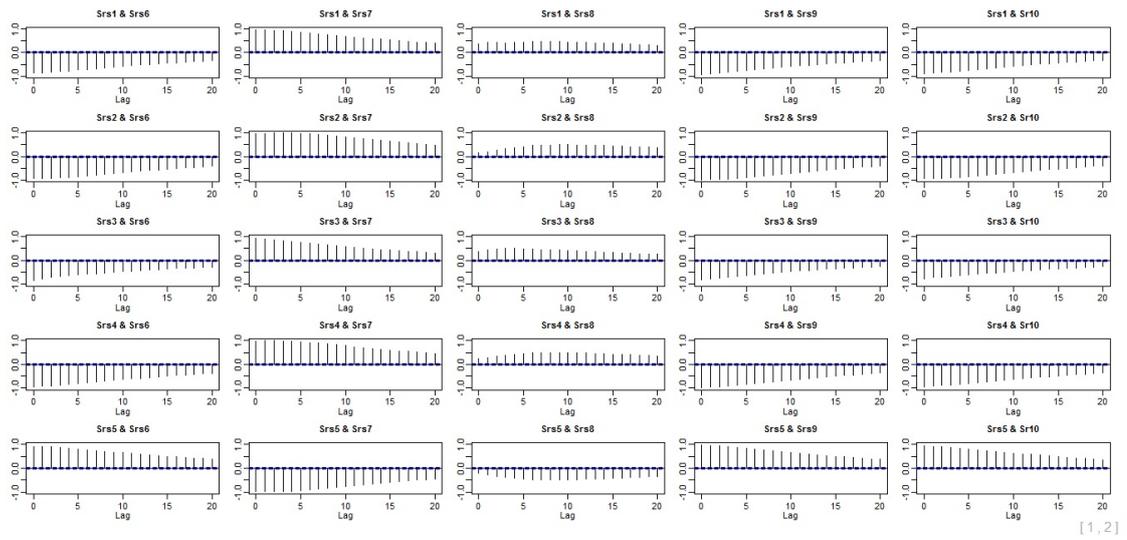


Figure 5.17: First 50 sample moving cross-correlation plots $W = 101$ of the randomly selected replica in Example 5.4.3 before transformation - part 2.

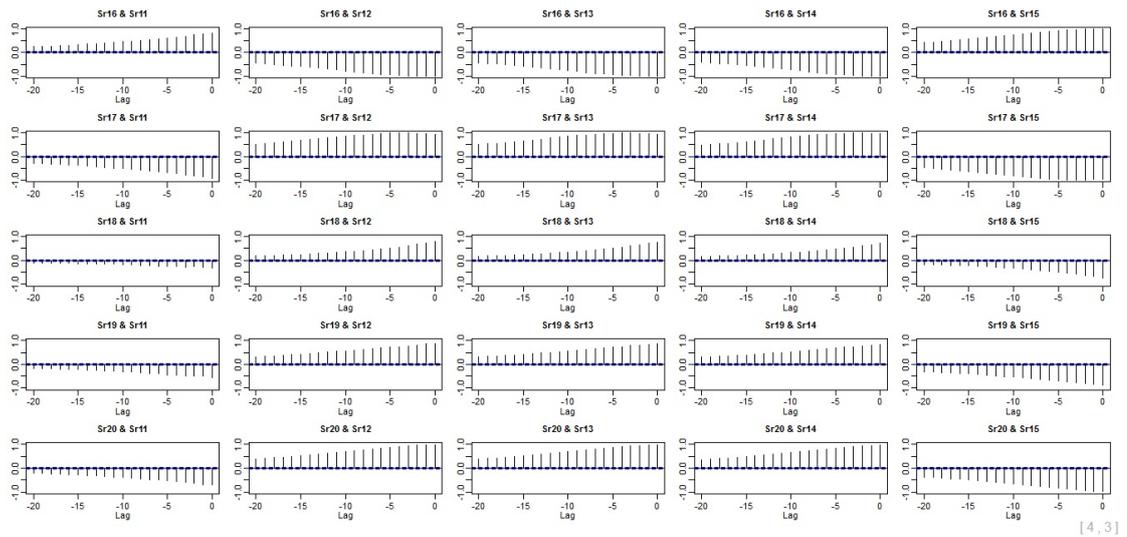


Figure 5.18: Last 50 sample moving cross-correlation plots $W = 101$ of the randomly selected replica in Example 5.4.3 before transformation - part 1.

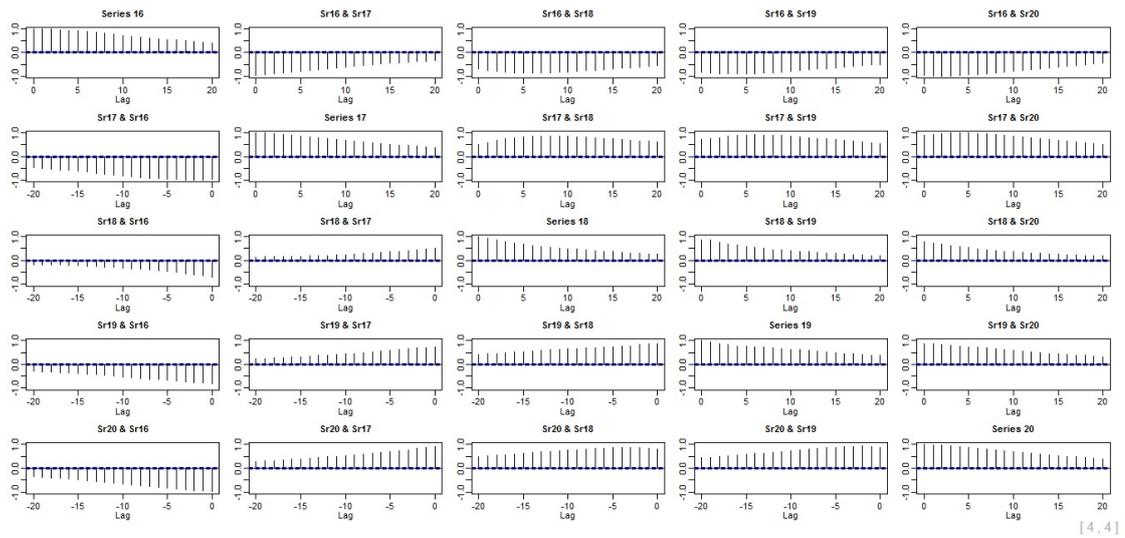


Figure 5.19: Last 50 sample moving cross-correlation plots $W = 101$ of the randomly selected replica in Example 5.4.3 before transformation - part 2.

In conclusion, the dynamic relationships between the 20 non-stationary series were successfully captured by GTS-PCA with $W = 101$ and the dimension was successfully reduced to five uncorrelated subseries of 6, 5, 4, 3 and 2 variables. Notice that, since we increased the dimension of the simulated series to $m = 20$ in this example, larger sample sizes were needed to obtain accurate results comparing with those in the previous examples.

Example 5.4.4 In this example, GTS-PCA is tested on a stationary data set, and then its results are validated with those of TS-PCA. Ideally, when GTS-PCA is applied to stationary data, then one window with a size $W = T$ would be suggested. This is equivalent to use TS-PCA. However, one would ask about the behaviour of GTS-PCA when it uses a window size that is smaller than the ideal size. In this simulation study, we are going to answer this question. We will consider almost the worst-case scenario, where GTS-PCA with a small window size (e.g. $W = 101$) is applied to a stationary data.

The data set used in this example is the one used by Chang et al. (2018) with $m = 6$ and $T = 1500$. The simulation codes are available in the *PCA4TS* library in R software. This data set consists of three subgroups of dimensions 3, 2 and 1.

A time series plot of a randomly selected series is available in Figure 5.20. Sample moving cross-correlation plots to show the relationships between the variables before transformation are provided in Figure 5.21. A moderate correlation existed across all six variables. A comparison between the results of GTS-PCA with $W = 101$ and TS-PCA are summarized in table 5.5. Sample moving cross-correlation plots of the data after transformation are shown in Figure 5.22. GTS-PCA and TS-PCA produced the same results for this data set, as they were both able to successfully find the correct segmentations.

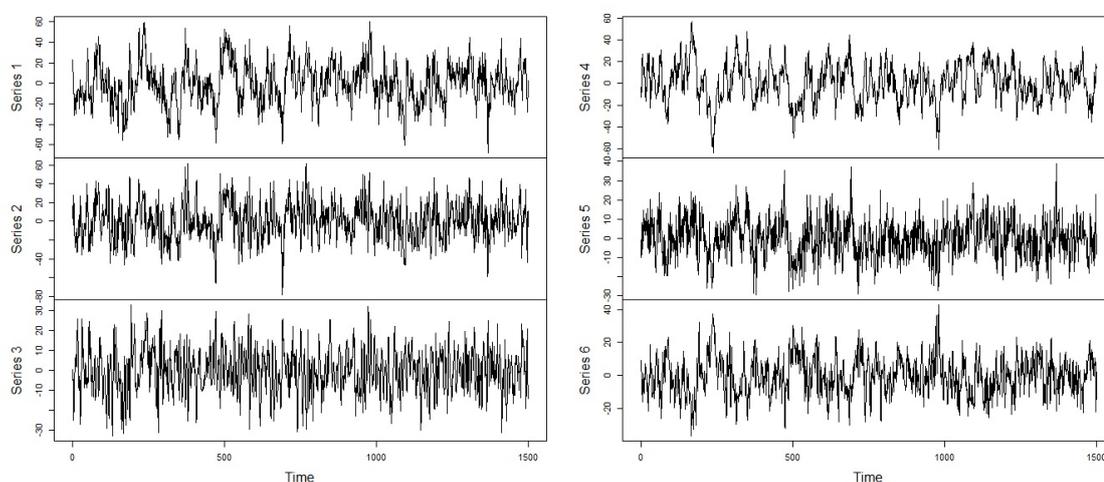


Figure 5.20: A time series plot of the randomly selected replica in Example 5.4.4.

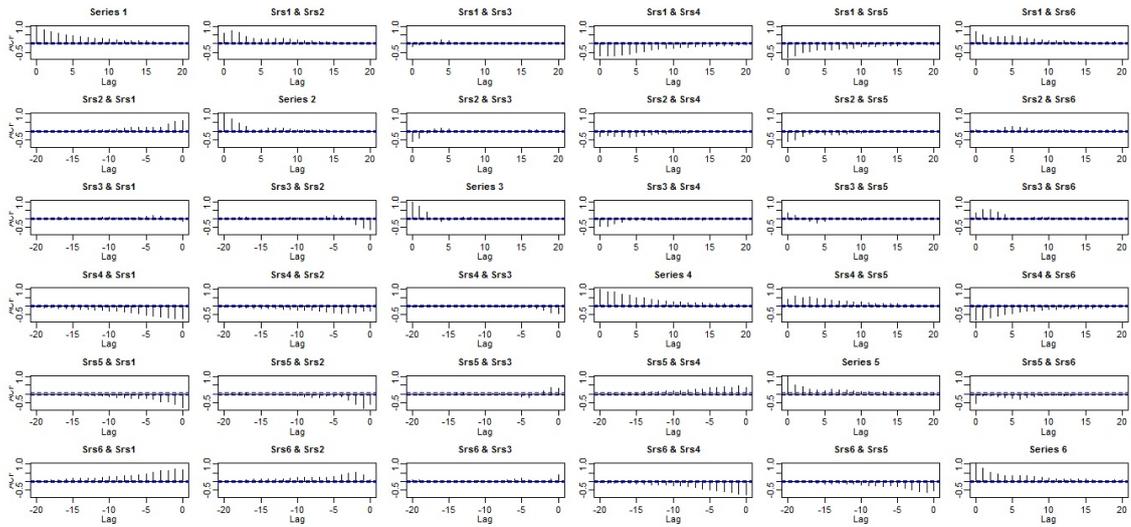


Figure 5.21: Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.4 before transformation.

Stationary Series $m=6, T=1500$	Number of subseries	Variables in subseries
GTS-PCA (window =101)	3 subseries of sizes 3, 2, 1	{1, 3, 6}, {2, 4}, {5}
TS-PCA	3 subseries of sizes 3, 2, 1	{1, 3, 6}, {2, 4}, {5}

Table 5.5: Number of subseries after applying GTS-PCA and TS-PCA to the randomly selected replica in Example 5.4.4.

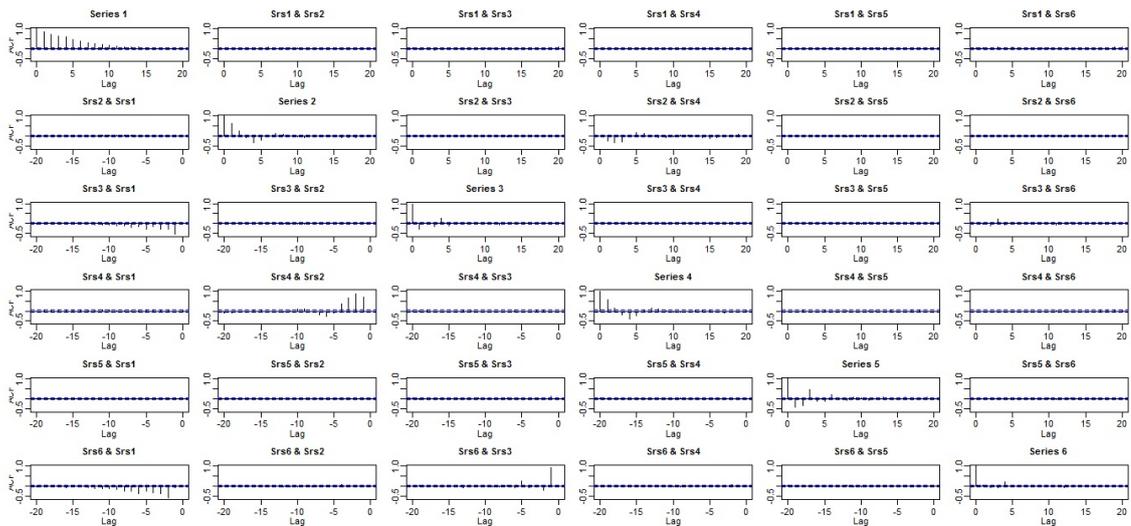


Figure 5.22: Sample moving cross-correlation plots with $W = 101$ of the randomly selected replica in Example 5.4.4 after GTS-PCA is applied.

Moreover, by replicating the above simulation 500 times, GTS-PCA with $W = 101$ was able to extract the correct segmentations 97% of the time. This is the same percentage that was reported by Chang et al. (2018) when TS-PCA was used. This indicates that both methods have identical performance on stationary series, even if GTS-PCA uses a window that is much smaller than the ideal size.

By the end of the above simulation studies, we conclude that GTS-PCA was able to reduce the dimension of a multivariate time series into uncorrelated subseries with lower dimensions. Our simulations showed that GTS-PCA performed well when applied to series with large sample sizes.

5.4.2 Real Data Examples

Example 5.4.5 In this example, we apply GTS-PCA on a non-stationary series that consists of the daily stock prices of six US companies in US Dollar from Nov 07, 2013 to Dec 18, 2017. The total number of observations is 1036 days. The six companies are CVS health (CVS), Chevron (CVX), Express Scripts Holding (ESRX), Intel (INTC), Lowe's (LOW) and Prudential Financial (PRU). The data was obtained from *Yahoo!* Finance.

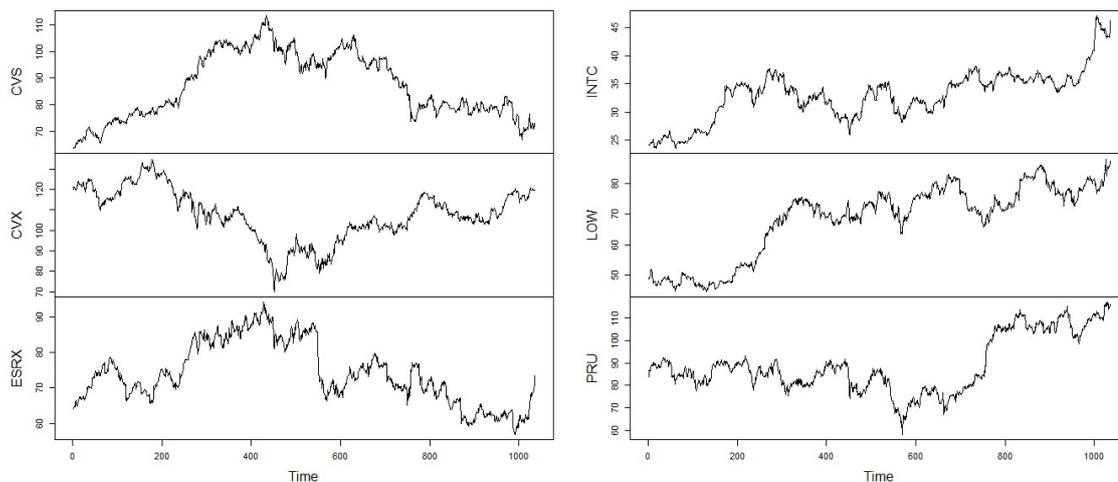


Figure 5.23: A time series plot of the daily stock prices of the six US companies in Example 5.4.5.

The six companies showed non-stationary behaviours over time, according to the time series plot in Figure 5.23. GTS-PCA with $W = 101$ and $l_1 = 5$ is used to reduce the dimension of the series. The sample moving cross-correlation plots with $W = 101$ between the six companies before applying GTS-PCA is provided in Figure 5.24. A moderated dynamic relationship existed between the six companies. For example, the company INTC had a moderate relationship with LOW. The companies INTC and PRU had a strong relationship with CVX for negative lagged values. Moreover, the company ESRX was weakly correlated with INTC.

After GTS-PCA was applied, the moving cross-correlation with $W = 101$ suggested dividing the data into four subseries with three univariate subseries and one multivariate subseries that consisted of the companies CVS, ESRX and LOW; See Figure 5.25.

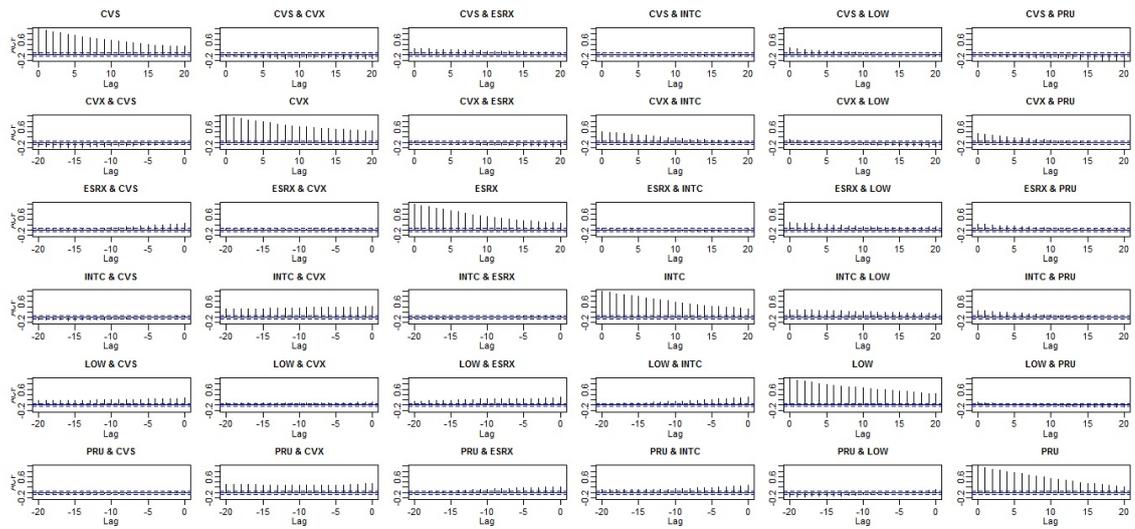


Figure 5.24: Sample moving cross-correlation plots of the daily stock prices of the six US companies in Example 5.4.5 before GTS-PCA is applied.

Therefore, we conclude that GTS-PCA was able to reduce the dimension of \mathbf{z}_t from six to four uncorrelated series of dimensions 3, 1, 1 and 1. These four sub-series can be analysed separately as they are uncorrelated statically and dynamically.

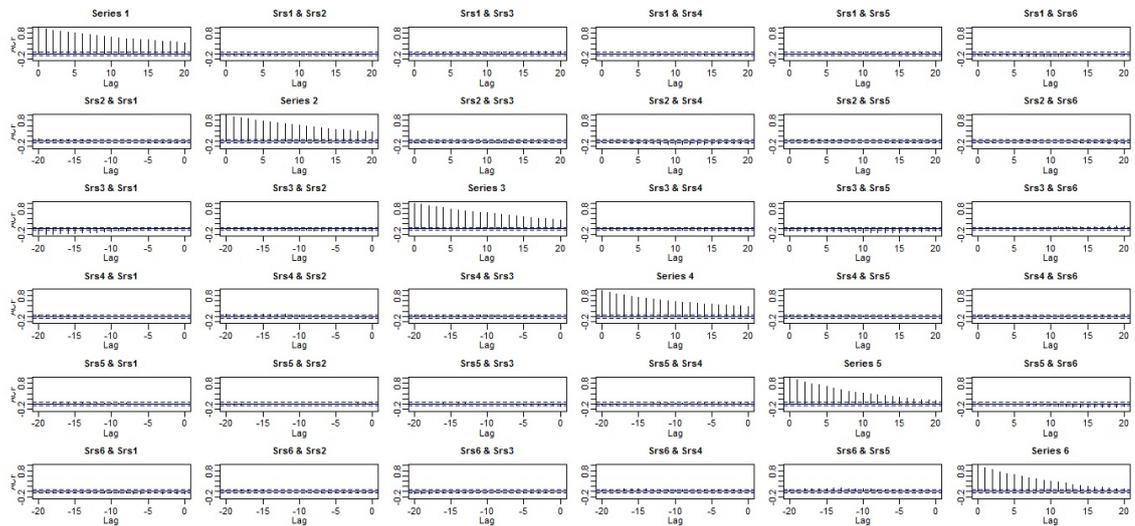


Figure 5.25: Sample moving cross-correlation plots of the daily stock prices of the six US companies in Example 5.4.5 after the GTS-PCA is applied.

Example 5.4.6 In the following example, we apply the GTS-PCA on a non-stationary

data set with 22 variables. The data set in this example consists of the daily stock prices of 22 US companies in US Dollar from Nov 07, 2013 to Dec 18, 2017 by *Yahoo! Finance*. The total number of observations is 1036 days. The companies' names and labels are listed in Table 5.6. A time series plot of the daily stock prices of the 22

No.	Label	Name	No.	Label	Name
1	CVS	CVS Health	12	KR	Kroger-The Kroger
2	CVX	Chevron	13	LOW	Lowe's
3	ESRX	Express Scripts Holding	14	MCK	McKesson
4	FB	Facebook	15	MET	MetLife Insurance
5	F	Ford Motor	16	MSFT	Microsoft
6	GE	General Electric	17	PEP	PepsiCo
7	GM	General Motor	18	PG	The Procter & Gamble
8	HD	The Home Depot	19	PRU	Prudential Financial
9	IBM	IBM	20	PSX	Philips 66
10	INTC	Intel	21	TGT	Target
11	JNJ	Johnson & Johnson	22	TWTR	Twitter

Table 5.6: Names and labels of the 22 US companies in Example 5.4.6.

companies is shown in figures 5.26, 5.27 and 5.28. The non-stationary behaviour of the variables can be clearly seen from the time series plot.

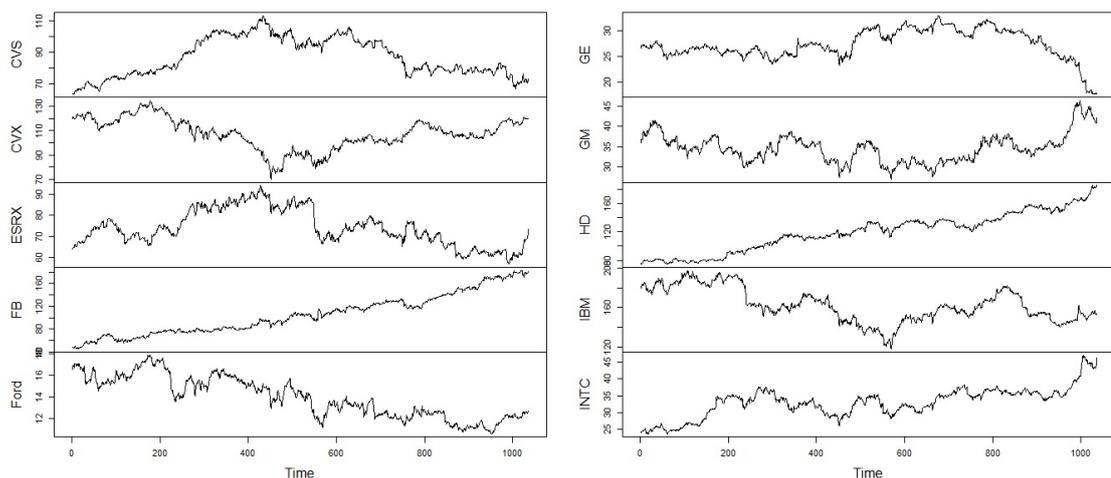


Figure 5.26: A time series plot of the daily stock prices of the 22 US companies in Example 5.4.6 - Part 1.

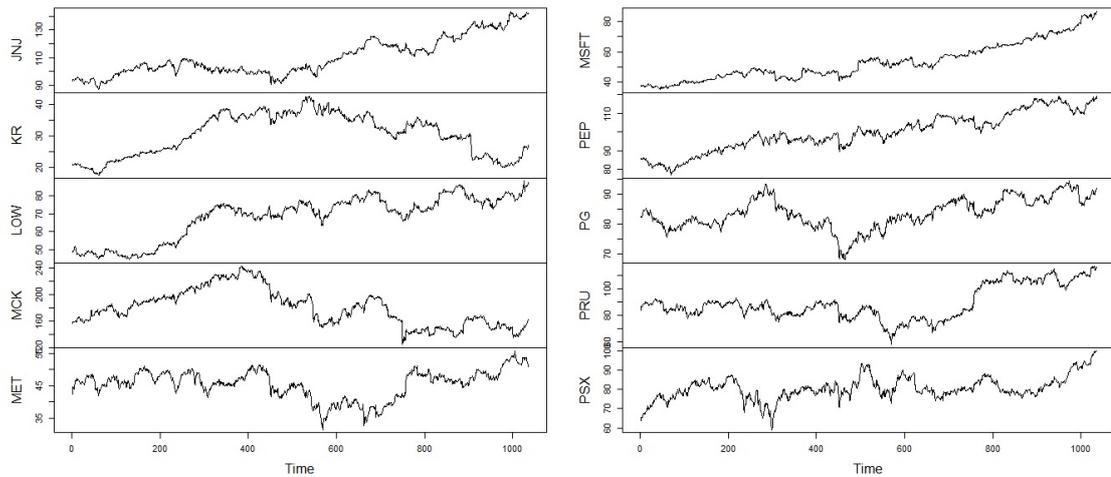


Figure 5.27: Time series plot of the daily stock prices of the 22 US companies in Example 5.4.6 - Part 2.

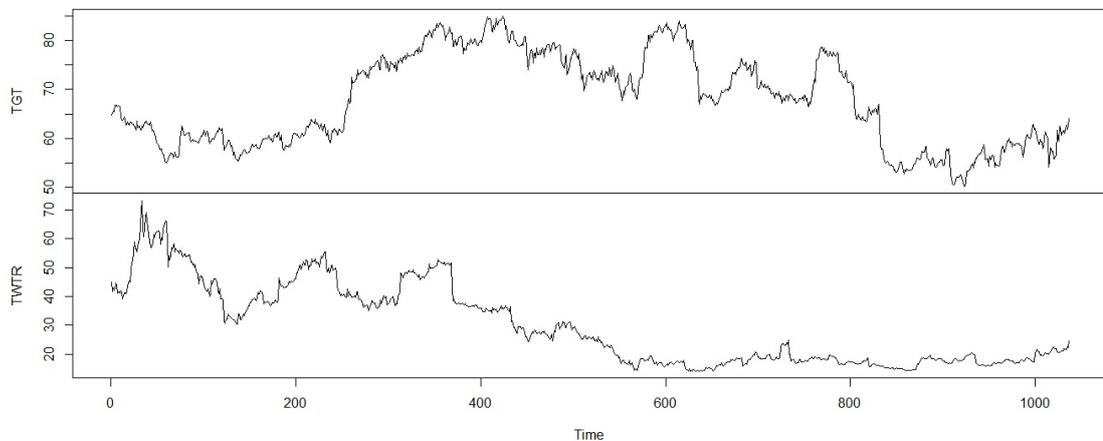


Figure 5.28: Time series plot of the daily stock prices of the 22 US companies in Example 5.4.6 - Part 3.

Figures 5.29, 5.30, 5.31 and 5.32 show only the first and last 50 sample moving cross-correlation plots with $W = 101$ of the 22 companies before applying GTS-PCA, where a moderated relationship existed between the 22 companies.

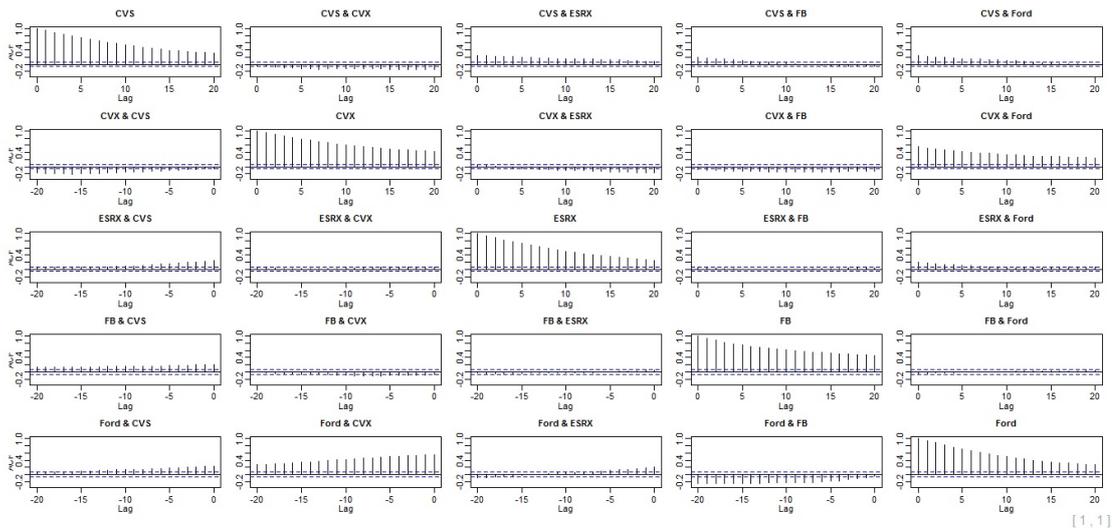


Figure 5.29: First 50 sample moving cross-correlation plots of the daily stock prices of the 22 US companies in Example 5.4.6 before GTS-PCA is applied - Part 1.

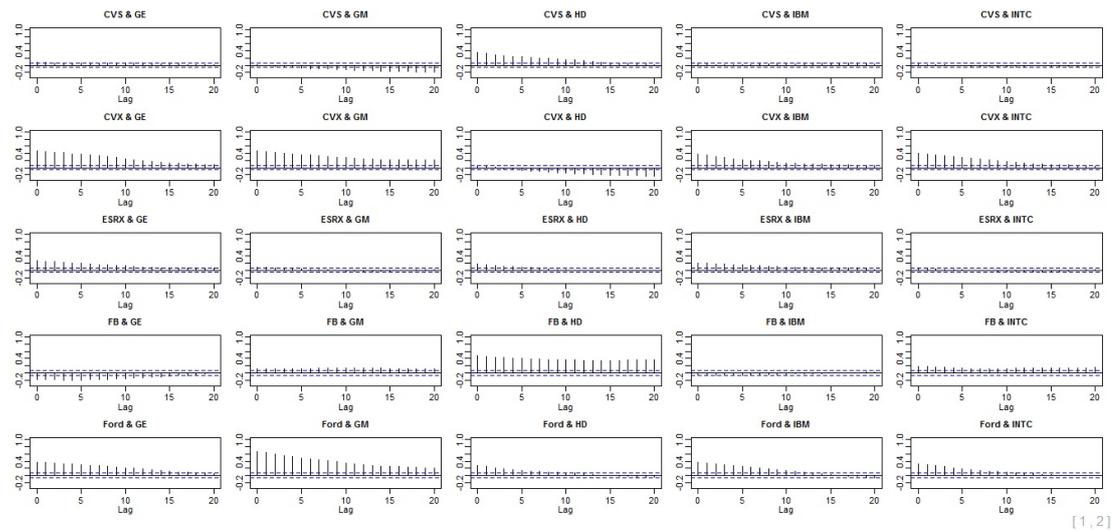
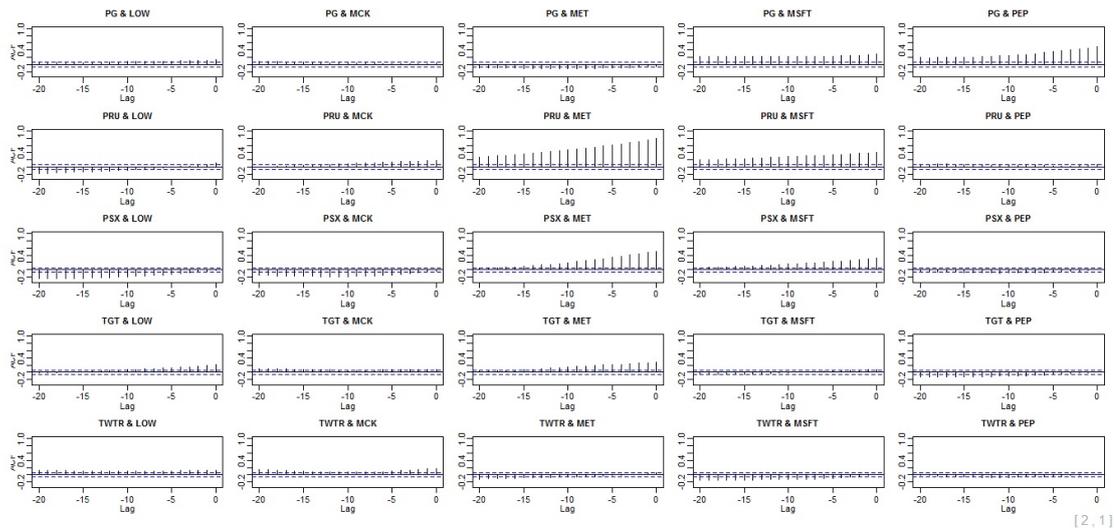
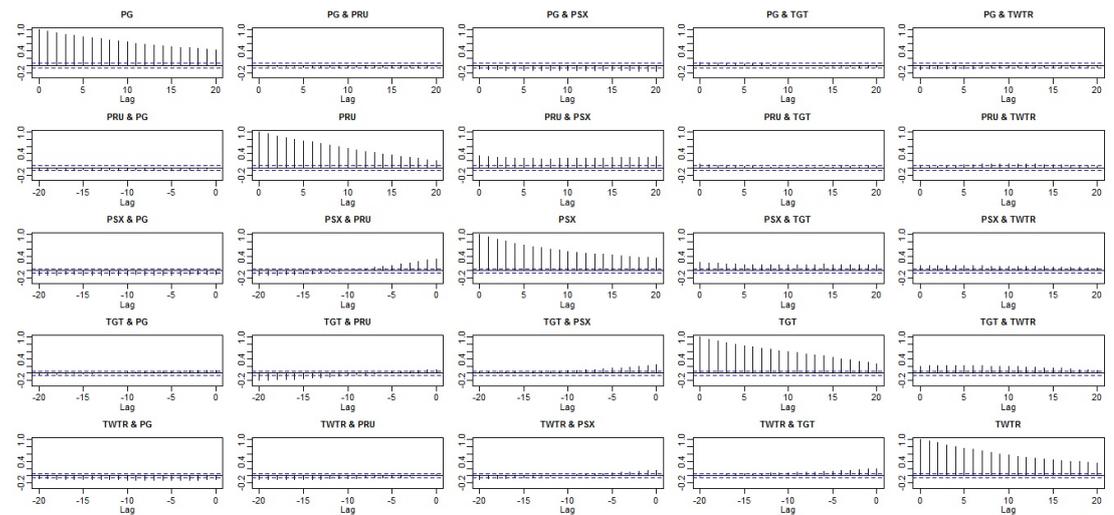


Figure 5.30: First 50 sample moving cross-correlation plots of the daily stock prices of the 22 US companies in Example 5.4.6 before GTS-PCA is applied - Part 2.



[2, 1]

Figure 5.31: Last 50 sample moving cross-correlation plots of the daily stock prices of the 22 US companies in Example 5.4.6 before GTS-PCA is applied - Part 1.



[2, 2]

Figure 5.32: Last 50 sample moving cross-correlation plots of the daily stock prices of the 22 US companies in Example 5.4.6 before GTS-PCA is applied - Part 2.

After applying GTS-PCA with $W = 101$ and $l_1 = 5$ to the daily stock prices, the maximum moving cross-correlation with $W = 101$ suggested to segment the data into 15 subseries that consisted of 11 univariate subseries, three bivariate subseries and one multivariate time series with five variables, as shown in Table 5.7. Further analyses can be applied separately to these 15 subseries since they are uncorrelated

statically and dynamically.

Results Method	Correlated Variables after applying GTS-PCA with (W=101)
Maximum Moving Cross-Correlation (W=101)	{CVX, F, HD, KR, MET}, {ESRX, LOW}, {GE, MSFT} and {JNJ, MCK}

Table 5.7: Summary results of the transformed segmentation of the 22 US companies in Example 5.4.6 after applying GTS-PCA with $W = 101$.

In conclusion, we were able to reduce the dimension of the non-stationary daily stock prices of 22 US company to 15 uncorrelated subseries by using GTS-PCA with a window size of 101 and 5 lagged series.

5.5 Summary and Remarks

In the following, we are going to summarize our work in Chapter 5 and state some remarks. GTS-PCA is a PCA-based dimension reduction method that can reduce the dimension of both stationary and non-stationary time series to uncorrelated subseries, where further analyses can be done separately on these subseries. TS-PCA can be considered as a special case of GTS-PCA where $W = T$, and they produce the same segmentation on stationary data; See Example 5.4.4.

GTS-PCA transform data by applying eigenanalysis on $\hat{\mathbf{M}}\mathbf{W}_z$ defined in 5.12. The moving cross-correlation function that we introduced in Chapter 3 can be used to rearrange the components of the transformed series with moderate dimensions into subseries based on their dynamic relationship. The maximum moving cross-correlation is a new proposed tool that is based on the moving cross-correlation function and can be used to rearrange the components of transformed series with higher dimensions into subseries based on their dynamic relationship.

Choosing the window size for GTS-PCA depends on the stationarity of the data.

GTS-PCA with smaller window sizes is more suitable for data with strong non-stationarity, and the opposite is true. Based on simulated data, we noticed that GTS-PCA was able to produce accurate results even when it used window sizes that are smaller than required (i.e. using a small window of size $W = 101$ on stationary data, where we suppose to use $W = T$).

We studied the convergence of the estimator $\hat{\mathbf{A}}$ when the dimension m is fixed and when m depends on the window size W for large dimensional cases. We stated the needed assumptions to obtain the convergence rates for these cases and showed that $\hat{\mathbf{A}}$ is a consistent estimator of \mathbf{A} . For ultra higher dimensions, where m depends on W , a threshold estimator of the sample cross-covariance matrix over each window $\hat{\Gamma}_{z,i}(l)$ should be used to preserve the convergence of $\hat{\mathbf{A}}$.

Our simulations showed that GTS-PCA performs reasonably on time series with different dimensions and sample sizes. We also tested GTS-PCA on real-life data sets. A drawback of GTS-PCA is that the transformed uncorrelated subseries might not exist. This might occur in some cases where all the transformed series are correlated. A similar conclusion was reported for the stationary TS-PCA; See Chang et al. (2018).

Chapter 6

Conclusions and Further Work

6.1 Conclusions

The goal of this thesis is to develop dimension reduction methods that can be applied to both stationary and non-stationary multivariate time series. We proposed three different PCA-based methods to reach our goal. Each one of the proposed methods can be considered as an extension or a generalization of an already existing dimension reduction method from the stationary to the non-stationary case. We showed that by using the moving cross-covariance matrix of the data instead of the classical cross-covariance matrix, we were able to account for static, dynamic, stationary and non-stationary behaviours in a time series. An important feature of the moving cross-covariance matrix is that it is calculated over windows or intervals whose lengths are predetermined according to the stationarity of the series. This allows the information collected by the moving cross-covariance matrix to be updated at each time point as we move in time.

In Chapter 3, we introduced the moving dynamic principal component analysis (MDPCA). This method is a generalization of DPCA of Ku et al. (1995) to non-stationary time series. Dimension reduction using MDPCA can be reached by left multiplying the extended data vector \mathbf{y}_t by the transpose of the first k^{th} columns of $\hat{\mathbf{U}}$, where $\hat{\mathbf{U}}$ is the matrix that consists of the eigenvectors of the sample moving cross-covariance matrix of \mathbf{y}_t . We also developed the RCC criterion in order to

determine the optimal number of retained components, k . Additionally, we introduced the moving cross-correlation function that can be plotted to evaluate static and dynamic relationships between the components of non-stationary time series data. Thus, this function can also evaluate the relationships between the retained components (i.e. MDPCs). The asymptotic results of the estimator of \mathbf{U} were discussed. We showed under some regularity assumptions that $\hat{\mathbf{U}}$ is a consistent estimator of \mathbf{U} with $W^{-1/2}$ convergence rate. The performance of MDPCA was investigated by applying it to simulated and real data sets. MDPCA showed a steady performance when applied to non-stationary series with moderate dimension sizes. MDPCA can also be used to reduce the dimension of stationary series by adjusting the window size to $W = N$, where its results are equivalent to those of DPCA. This makes DPCA a special case of MDPCA where $W = N$. We also reported that the performance of MDPCA improved as the sample size increased, where consistent performance was obtained on data with reasonably small sample sizes.

The quadratic moving dynamic principal component analysis (QMDPCA) was introduced in Chapter 4. This is a PCA-based method that can transfer a stationary and non-stationary multivariate time series into uncorrelated lower-dimensional components (i.e. QMDPCs). The details and steps needed to transfer data using QMDPCA were discussed in the methodology section. Dimensional reduction using this method can be reached by left multiplying the extended data vector \mathbf{y}_t by the transpose of the first k^{th} columns of $\hat{\mathbf{U}}$ (i.e. eigenvectors that correspond to the first k^{th} eigenvalues of the sample quadratic order moving cross-covariance matrix of \mathbf{y}_t). The RCC criterion introduced in Chapter 3 can be used to determine the optimal number of retained QMDPCs, k . The relationships between the retained components can be investigated by plotting their sample moving cross-correlation matrix introduced in Chapter 3. The convergence of the estimator of \mathbf{U} was studied where, under some regularity assumptions, we showed that $\hat{\mathbf{U}}$ is a consistent estimator of \mathbf{U} with $W^{-1/2}$ convergence rate. The numerical properties of QMDPCA were shown on simulated and real data sets with different combinations of sample and dimension sizes. QMDPCA showed consistent performance on both stationary

and non-stationary series. Additionally, we reported that the accuracy of QMDPCA could be further enhanced by increasing the sample size.

In Chapter 5, we proposed the generalized principal components analysis for time series data (GTS-PCA). This method is a generalization of TS-PCA of Chang et al. (2018) to non-stationary time series data. GTS-PCA can achieve dimension reduction by transforming series into uncorrelated subseries with lower dimensions. These subseries can be analysed separately since they are not correlated. The transformed subseries \mathbf{x}_t can be obtained by using the model $\mathbf{x}_t = \hat{\mathbf{A}}' \mathbf{z}_t$, where $\hat{\mathbf{A}}$ is the matrix that consists of all eigenvectors of the sample lag l quadratic moving cross-covariance matrix of the original data vector \mathbf{z}_t , subject to columns' rearrangement. For series with small to moderate dimension sizes, the correct columns' rearrangement can be obtained by investigating the moving cross-correlation plots of the transformed series. For series with large dimensions, we suggest using the maximum moving cross-covariance method introduced in Section 5.2 in Chapter 5 to avoid investigating a large number of plots. Since GTS-PCA is developed to deal with high dimensional series, the theoretical properties of this method were discussed in three cases, where the dimension m is fixed, $m = o(W^c)$ and $\log m = o(W^c)$. We provided the convergence rates of the estimator $\hat{\mathbf{A}}$ in these cases. It is essential to mention that for the cases where m depends on W , a threshold estimator of the sample cross-covariance matrix over each window $\hat{\Gamma}_{z,i}(l)$ should be used to preserve the convergence of $\hat{\mathbf{A}}$. Furthermore, GTS-PCA was applied to simulated and real data sets of different sample and dimension sizes. We showed that GTS-PCA has the ability to reduce the dimension of both stationary and non-stationary series by adjusting its window size W . We reported that TS-PCA is a special case of GTS-PCA where $W = T$. We also reported that the ability of GTS-PCA to reveal the hidden segmentations enhanced as we increased the sample size T .

There are a few critical differences between GTS-PCA and QMDPCA. The first difference is the way they deal with lagged series. The former applies eigenanalysis on \mathbf{MW}_z , which is a sum of the quadratic order of the lagged moving cross-covariance

matrices of the original data vector \mathbf{z}_t up to some predetermined lag. The latter, on the other hand, requires to build an extended data vector \mathbf{y}_t that consists of lagged series of \mathbf{z}_t up to some predetermined lag l , then apply eigenanalysis on \mathbf{Q} , the quadratic moving cross-covariance matrix of \mathbf{y}_t with no lags included. The second difference is the total number of produced eigenvalues of the two methods. GTS-PCA produces a total of m eigenvalues, where QMDPCA produces a total of $m(l + 1)$ eigenvalues. Another difference is that the two methods seek dimension reduction differently. GTS-PCA seeks uncorrelated subseries such that the sum of their dimensions is equal to the dimension of the original series m . QMDPCA, on the other hand, seeks k (where $k < m$) uncorrelated components (i.e. QMDPCs) that consist of the most variation of the data and produce minimum reconstruction error.

As we show, the methods we proposed can be applied to stationary and non-stationary time series with various dimensions. As mentioned earlier, there are cases where GTS-PCA is not able to achieve dimension reduction when the component of the transformed data are correlated. In this case, the usage of MDPCA and QMDPCA is recommended. Additionally, applying MDPCA and QMDPCA on large dimensional series might produce less number of principal components to analyse compared with those produced by GTS-PCA. For example, for a series with ten variables, GTS-PCA could produce ten univariate principal components, in the ideal case, where one univariate principal component would be ideal for MDPCA or QMDPCA.

6.2 Further Work

In this section, we provide some ideas and open questions that can be solved in further research works.

- Is possible to objectively quantify the degree of non-stationarity of time series such that optimal window sizes are objectively determined to carry out MDPCA, QMDPCA and GTS-PCA?

-
- How to develop a theory to show the convergence rate of the estimator of \mathbf{U} used in MDPCA for series with ultra-high dimensions and what are the necessary assumptions needed in this case?
 - How to develop a theory and state the sufficient assumptions to approach the convergence of the estimator of \mathbf{U} used in QMDPCA for series with ultra-high dimensions?
 - Is it possible to develop a dimension reduction method for time series such that it combines between PCA and Co-integration?

Bibliography

- Alshammri, F. (2020a). *GTSPCA: Generalized Principal Component Analysis for Non-Stationary Vector Time Series*. R package version 0.1.0.
- Alshammri, F. (2020b). *macf: Moving Auto- and Cross-correlation Function*. R package version 0.1.0.
- Alshammri, F. (2020c). *mcov: Moving Cross-covariance Matrix*. R package version 0.1.0.
- Alshammri, F. (2020d). *MDPCA: Moving Dynamic Principal Component Analysis for Non-Stationary Multivariate Time Series*. R package version 0.1.0.
- Alshammri, F. (2020e). *MpermutMax: The Maximum Moving Cross-Correlation Method*. R package version 0.1.0.
- Alshammri, F. (2020f). *QMDPCA: Quadratic Moving Dynamic Principal Component Analysis for Non-Stationary Multivariate Time Series*. R package version 0.1.0.
- Alshammri, F. (2020g). *RCCM: Retained Component Criterion for the Moving Dynamic Principal Component Analysis*. R package version 0.1.0.
- Alshammri, F. (2020h). *RCCQ: Retained Component Criterion for the Quadratic Moving Dynamic Principal Component Analysis*. R package version 0.1.0.
- Bai, J. and Ng, S. (2002). Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221.

- Bickel, P. J. and Levina, E. (2008). Covariance regularization by thresholding. *The Annals of Statistics*, 36(6):2577–2604.
- Box, G. and Jenkins, G. (1976). *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco.
- Box, G. E. and Tiao, G. C. (1977). A canonical analysis of multiple time series. *Biometrika*, 64(2):355–365.
- Bradley, R. C. (1986). Basic Properties of Strong Mixing Conditions. In Eberlein, E. and Taqqu, M. S., editors, *Dependence in Probability and Statistics: A Survey of Recent Results*, volume 11 of Progress in Probability and Statistics, pages 165–192. Birkhäuser, Boston, MA.
- Brillinger, D. (1981). *Time Series: Data Analysis and Theory*, volume 36. Siam, Philadelphia.
- Chang, J., Guo, B., and Yao, Q. (2018). Principal component analysis for second-order stationary vector time series. *The Annals of Statistics*, 46(5):2094–2124.
- Chen, X., Xu, M., and Wu, W. B. (2013). Covariance and precision matrix estimation for high-dimensional time series. *The Annals of Statistics*, 41(6):2994–3021.
- Dahlhaus, R. and Rao, S. S. (2006). Statistical inference for time-varying arch processes. *The Annals of Statistics*, 34(3):1075–1114.
- Dickey, D. A. and Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366a):427–431.
- Forni, M., Hallin, M., Lippi, M., and Reichlin, L. (2005). The generalized dynamic factor model: one-sided estimation and forecasting. *Journal of the American Statistical Association*, 100(471):830–840.
- Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD.

- He, X. B. and Yang, Y. P. (2008). Variable mwpca for adaptive process monitoring. *Industrial and Engineering Chemistry Research*, 47(2):419–427.
- Henderson, H. V. and Velleman, P. F. (1981). Building multiple regression models interactively. *Biometrics*, 37(2):391–411.
- Hosking, J. R. (1980). The multivariate portmanteau statistic. *Journal of the American Statistical Association*, 75(371):602–608.
- Jentsch, C. and Subba Rao, S. (2015). A test for second order stationarity of a multivariate time series. *Journal of Econometrics*, 185(1):124–161.
- Johansen, S. (1995). *Likelihood-based Inference in Cointegrated Vector Autoregressive Models*. Advanced texts in econometrics. Oxford University Press, Oxford ; New York.
- Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer series in statistics. Springer, New York, 2nd edition.
- Ku, W., Storer, R. H., and Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30(1):179–196.
- Lam, C. and Yao, Q. (2012). Factor modeling for high-dimensional time series: inference for the number of factors. *The Annals of Statistics*, 40(2):694–726.
- Lennox, B., Montague, G., Hiden, H., Kornfeld, G., and Goulding, P. (2001). Process monitoring of an industrial fed-batch fermentation. *Biotechnology and Bioengineering*, 74(2):125–135.
- Pan, J. and Yao, Q. (2008). Modelling multiple time series via common factors. *Biometrika*, 95(2):365–379.
- Peña, D. and Box, G. E. (1987). Identifying a simplifying structure in time series. *Journal of the American Statistical Association*, 82(399):836–843.
- Peña, D. and Poncela, P. (2006). Nonstationary dynamic factor analysis. *Journal of Statistical Planning and Inference*, 136(4):1237–1257.

- Peña, D. and Yohai, V. J. (2016). Generalized dynamic principal components. *Journal of the American Statistical Association*, 111(515):1121–1131.
- Reinsel, G. (2003). *Elements of Multivariate Time Series Analysis*. Springer, New York.
- Stock, J. H. and Watson, M. W. (1988). Testing for common trends. *Journal of the American Statistical Association*, 83(404):1097–1107.
- Stock, J. H. and Watson, M. W. (2002). Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, 97(460):1167–1179.
- Tiao, G. C. and Tsay, R. S. (1989). Model specification in multivariate time series. *Journal of the Royal Statistical Society: Series B (Methodological)*, 51(2):157–213.
- Tsay, R. (2013). *Multivariate Time Series Analysis: With R and Financial Applications*. Wiley, Hoboken, NJ.
- Vogt, M. (2012). Nonparametric regression for locally stationary time series. *The Annals of Statistics*, 40(5):2601–2633.
- Zhang, R., Robinson, P., and Yao, Q. (2019). Identifying cointegration by eigenanalysis. *Journal of the American Statistical Association*, 114(526):916–927.