DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

AVERAGE INTEGRATED CAUSATION ENTROPY IN ANALYSIS OF CHAOTIC DYNAMICAL NETWORKS IN COMPLETE AND CLUSTER SYNCHRONIZATION

by
Özge CANLI USTA

September, 2022
İZMİR

AVERAGE INTEGRATED CAUSATION ENTROPY IN ANALYSIS OF CHAOTIC DYNAMICAL NETWORKS IN COMPLETE AND CLUSTER SYNCHRONIZATION

A Thesis Submitted to the
Graduate School of Natural And Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Department of Electrical and Electronics Engineering

by
Özge CANLI USTA

September, 2022
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**AVERAGE INTEGRATED CAUSATION ENTROPY IN ANALYSIS OF CHAOTIC DYNAMICAL NETWORKS IN COMPLETE AND CLUSTER SYNCHRONIZATION**" completed by **Özge CANLI USTA** under supervision of **ASSOC. PROF. DR. SERKAN GÜNEL** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

.....

Assoc. Prof. Dr. Serkan GÜNEL

Supervisor

.....	.....
Prof. Dr. Mustafa A. ALTINKAYA	Prof. Dr. Olcay AKAY
Thesis Committee Member	Thesis Committee Member
.....	.....
Prof. Dr. Müştak E. YALÇIN	Prof. Dr. Serdar İPLİKÇİ
Examining Committee Member	Examining Committee Member

Prof. Dr. Okan FISTIKOĞLU
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor Assoc. Prof. Dr. Serkan GÜNEL for providing me to think in different ways, listen patiently, impose ethical principles, and motivate me tirelessly. It was a great privilege and honor to work and study under his guidance.

Secondly, I would like to thank Prof. Dr. Olcay AKAY and Prof. Dr. Mustafa A. ALTINKAYA for their valuable contributions and comments throughout the thesis progress meetings.

Moreover, my appreciation would go to all of my friends Dr. Ali Emre KAVUR, Dr. Münire Sibel ÇETİN, Dr. Duygu ERDEM AKGÜN, İpek LÖK, Gizem KALENDER, Ekrem YAVUZ and Dr. Nail AKÇURA for their help and the beautiful days we spent together during my Ph.D. Finally, I would like to thank my parents, my brother, my sister and my husband for their love, encouragement and support of my decisions.

This thesis was supported by the Council of Higher Education under *Dokuz Eylül University Academia Education Program (Öğretim Üyesi Yetiştirme Programı - ÖYP)* and Dokuz Eylül University Scientific Research Projects (DEUBAP) under the Grant no. 2021.KB.FEN.007.

Özge CANLI USTA

AVERAGE INTEGRATED CAUSATION ENTROPY IN ANALYSIS OF CHAOTIC DYNAMICAL NETWORKS IN COMPLETE AND CLUSTER SYNCHRONIZATION

ABSTRACT

The chaotic dynamical systems with characteristically irregular and unpredictable behavior can form synchronized complex networks when allowed to interact. The main aim of this thesis is to analyze the synchronized chaotic networks in the view of information theory and to discover the network connectivity information using the information-theoretic measures.

The entropy-based measures estimated from observations are a valuable way to quantify information flow between the subsystems since they can provide a model-free approach for identifying network structures. This thesis initially addresses estimating the entropy-based measures to represent the information flow accurately. A comparative study has been conducted with the existing measures and the estimators. A new k-nearest neighbor causation entropy estimator based on the existing causation entropy measure and other similar estimators has been proposed to represent information flow effectively.

This thesis mainly focuses on inferring the unknown network properties using limited observations in directed chaotic synchronized networks even if the underlying dynamics and connections of the network are unknown. The problem is compelling since the observations in synchronized systems are identical. On the other hand, the systems can temporarily be perturbed to destroy synchronization. The average integrated causation entropy measure, proposed herein, evaluated in the partially reconstructed state space of the network, can be used to determine network connectivity as the systems tend to resynchronize after perturbations. The average integrated causation entropy properties have been investigated analytically, and a novel algorithm has been developed to reveal the network coupling matrix. It has been found that the proposed measure and related algorithm based on the difference

of average integrated causation entropy reveals the network structure in random networks in complete synchronization.

We have also shown that the proposed measure can be used to detect clusters in chaotic cluster synchronization networks. A novel algorithm has been suggested based on the k-means clustering of the proposed measure with only single series of observation. The results indicate that the proposed procedure can distinguish the systems regarding their memberships in the clusters formed.

Keywords: Synchronization, cluster synchronization, information-theory, causation entropy, integrated causation entropy, entropy estimation, network reconstruction

TAM VE KÜME EŞZAMANLILIĞI GÖSTEREN KAOTİK DİNAMİK AĞLARIN ORTALAMA TOPLAMSAL NEDENSELLİK ENTROPİSİ İLE ANALİZİ

ÖZ

Karakteristik olarak tahmin edilemeyen davranışlara sahip kaotik dinamik sistemlerin eşzamanlı karmaşık ağlar oluşturabildiği bilinmektedir. Bu tezin ana amacı, eşzamanlı kaotik ağları bilgi teorisi ölçülerini kullanarak analiz etmek ve ağların bağlantılılık bilgisini keşfetmektir.

Gözlemlerden kestirilen entropiye dayalı ölçüler, ağ yapılarının tanılanması için modelden bağımsız bir yaklaşım sağlayabildiklerinden, alt sistemler arasındaki bilgi akışını nicelendirebilmek için değerli bir yol sunar. Bu tez ilk olarak, bilgi akışını doğru bir şekilde nicelendirebilmek için entropiye dayalı ölçülerin kestirilmesi problemini ele almaktadır. Var olan ölçüler ve kestirimciler ile kıyaslamalı bir çalışma yapılmıştır. Ağ içindeki sistemler arasındaki bilgi akışını etkin bir şekilde temsil etmek için var olan nedensellik entropisi ölçüsüne dayalı yeni bir k-en yakın komşu nedensellik entropi kestirimcisi önerilmiştir.

Esas olarak bu tez, ağın temel dinamikleri ve bağlantılılığı bilinmese bile, yönlü kaotik eşzamanlı ağlarda sınırlı gözlemler kullanarak bilinmeyen ağ özelliklerinin çıkarılmasına odaklanmaktadır. Eşzamanlı sistemlerdeki gözlemler aynı olduğu için problem zorlayıcıdır. Öte yandan, eşzamanlılığı geçici olarak bozmak için sistemler perturbe edilebilir. Burada önerilen ortalama toplamsal nedensellik entropi ölçüsü, sistemler yeniden eşzamanlı olma eğilimindeyken kısmen geriçatılan durum uzayında hesaplanarak, ağ bağlantılarını belirlemek için kullanılabilir. Ortalama toplamsal nedensellik entropisinin özellikleri analitik olarak incelenmiş ve ağ bağlantılılık matrisini ortaya çıkarmak için yeni bir algoritma geliştirilmiştir. Önerilen ortalama toplamsal nedensellik entropisi ölçüsü ve ilgili algoritmanın, rasgele bağlantılar içeren tam eşzamanlı ağlardaki bağlantılılık yapısını ortaya çıkardığı gösterilmiştir.

Önerilen ölçünün kaotik küme eşzamanlı ağlarda kümelerin tespiti için kullanılabileceği de gösterilmiştir. Önerilen yöntem ve k-ortalama kümelenmesini de kullanılarak bu tür ağlardaki küme aidiyetlerini tek bir gözlem serisi üzerinden belirleyen yeni bir algoritma önerilmiştir. Sonuçlar, önerilen yaklaşımın küme eşzamanlı sistem içindeki grupları ayırt edebildiğini göstermektedir.

Anahtar kelimeler: Eşzamanlılık, öbek eşzamanlılığı, bilgi kuramı, nedensellik entropisi, toplamsal nedensellik entropisi, entropinin kestirimi, ağ geriçatma

CONTENTS

Page

Ph.D. THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZ vi
LIST OF FIGURES xiv
LIST OF TABLES xv
LIST OF SYMBOLS xvii
ABBREVIATIONS xviii

CHAPTER ONE – INTRODUCTION 1

1.1 Thesis Statement and Contributions 5
1.2 Outline 7

CHAPTER TWO – MATHEMATICAL BACKGROUND 9

2.1 Graph Theoretic Definitions 9
2.2 Dynamical Networks 11
2.3 State Space Reconstruction 17
2.4 Synchronization 21
2.5 Entropy 23

CHAPTER THREE – SYNCHRONIZATION OF DYNAMICAL SYSTEMS .. 28

3.1 Preliminaries 30
3.2 Design of Synchronous Networks 30
3.2.1 Global synchronization with master stability function 31
3.2.2 A method to realize cluster synchronization 32
3.2.3 Cluster synchronization network via the distributed controller 35

CHAPTER FOUR – INFORMATION THEORY 40

4.1 Information-Theoretic Tools 40
4.2 Information-Theoretic Tools in case of Synchronization 48

CHAPTER FIVE – ESTIMATION OF ENTROPY BASED MEASURES 50

5.1 Entropy Estimation Techniques 52
5.1.1 Histogram estimator 54
5.1.2 Kernel density estimator 55
5.1.3 k-nearest neighbor entropy estimator 56
5.1.4 kpN entropy estimator 60
5.2 Simulations and Results 63
5.2.1 Coupled autoregressive processes 64
5.2.2 Bistable system 79
5.2.3 Temperature electricity consumption dataset 86

CHAPTER SIX – UNFOLDING THE STRUCTURE OF CHAOTIC DYNAMICAL NETWORKS VIA CAUSATION ENTROPY 90

6.1 Network Reconstruction via Causation Entropy 92
6.2 Network Reconstruction Algorithm 101
6.3 Numerical Results 104

CHAPTER SEVEN – DETECTION OF CLUSTERS OF CHAOTIC DYNAMICAL NETWORKS VIA CAUSATION ENTROPY 113

7.1 The Problem Statement 114
7.2 Illustrative Examples 118
7.3 Detection of Cluster Synchronization when the Clusters Change 131

CHAPTER EIGHT – CONCLUSION 140

8.1 Future Work 143

REFERENCES 144

LIST OF FIGURES

Page

Figure 2.1 Lorenz attractor for $\sigma = 16$, $b = 4$ and $r = 45.92$. 14

Figure 2.2 Chua's circuit and its nonlinear characteristics 15

Figure 2.3 Chua Attractor..... 16

Figure 2.4 The state space reconstruction of the chaotic attractor 20

Figure 3.1 The complete synchronization network. 32

Figure 3.2 The global synchronization error in Figure 3.1 for various ϵ's 33

Figure 3.3 The cluster synchronization network. 34

Figure 3.4 $e_{i,j}(t)$, i, j = 1, 2, 3, 4 for the systems in the cluster synchronization network..... 36

Figure 3.5 The cluster synchronization network via the distributed controller..... 38

Figure 3.6 Synchronization error between d and k system and global synchronization error without distributed control 38

Figure 3.7 Synchronization error between d and k system and global synchronization error with distributed control..... 39

Figure 5.1 The number of points determination in kNN estimator 61

Figure 5.2 The NB of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using histogram estimator 66

Figure 5.3 The NSE of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using histogram estimator ... 67

Figure 5.4 The NB of $\widehat{TE}_{X_1 \to X_2}$, $\widehat{TE}_{X_1 \to X_3}$, $\widehat{TE}_{X_2 \to X_3}$ $\widehat{TE}_{X_2 \to X_1}$ using histogram estimator 68

Figure 5.5 The NSE of $\widehat{TE}_{X_1 \to X_2}$, $\widehat{TE}_{X_1 \to X_3}$, $\widehat{TE}_{X_2 \to X_3}$ $\widehat{TE}_{X_2 \to X_1}$ using histogram estimator 68

Figure 5.6 The NB of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using Gaussian KDE 69

Figure 5.7 The NSE of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using Gaussian KDE 69

Figure 5.8 The NB of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using Epanechnikov KDE 70

Figure 5.9 The NSE of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using Epanechnikov KDE ... 70

Figure 5.10 The NB of $\widehat{TE}_{X_1 \to X_2}$, $\widehat{TE}_{X_1 \to X_3}$, $\widehat{TE}_{X_2 \to X_3}$ $\widehat{TE}_{X_2 \to X_1}$ using Gaussian KDE71

Figure 5.11 The NSE of $\widehat{TE}_{X_1 \to X_2}$, $\widehat{TE}_{X_1 \to X_3}$, $\widehat{TE}_{X_2 \to X_3}$ $\widehat{TE}_{X_2 \to X_1}$ using Gaussian KDE71

Figure 5.12 The NB of $\widehat{TE}_{X_1 \to X_2}$, $\widehat{TE}_{X_1 \to X_3}$, $\widehat{TE}_{X_2 \to X_3}$ $\widehat{TE}_{X_2 \to X_1}$ using Epanechnikov KDE 72

Figure 5.13 The NSE of $\widehat{TE}_{X_1 \to X_2}$, $\widehat{TE}_{X_1 \to X_3}$, $\widehat{TE}_{X_2 \to X_3}$ $\widehat{TE}_{X_2 \to X_1}$ using Epanechnikov KDE 72

Figure 5.14 The NB of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using kNN estimator 73

Figure 5.15 The NSE of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using kNN estimator 73

Figure 5.16 The NB of $\widehat{TE}_{X_1 \to X_2}$, $\widehat{TE}_{X_1 \to X_3}$, $\widehat{TE}_{X_2 \to X_3}$, $\widehat{TE}_{X_2 \to X_1}$ using kNN estimator74

Figure 5.17 The NSE of $\widehat{TE}_{X_1 \to X_2}$, $\widehat{TE}_{X_1 \to X_3}$, $\widehat{TE}_{X_2 \to X_3}$, $\widehat{TE}_{X_2 \to X_1}$ using kNN estimator 74

Figure 5.18 The NB of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using kpN estimator when N = 4096 75

Figure 5.19 The NB of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using kpN estimator when N = 8192 75

Figure 5.20 The NSE of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using kpN estimator when N = 4096 76

Figure 5.21 The NSE of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using kpN estimator when N = 8192 76

Figure 5.22 The NB of $\widehat{TE}_{X_1 \to X_2}$, $\widehat{TE}_{X_1 \to X_3}$, $\widehat{TE}_{X_2 \to X_3}$ $\widehat{TE}_{X_2 \to X_1}$ using kpN estimator when N = 4096 77

Figure 5.23 The NB of $\widehat{TE}_{X_1 \to X_2}$, $\widehat{TE}_{X_1 \to X_3}$, $\widehat{TE}_{X_2 \to X_3}$ $\widehat{TE}_{X_2 \to X_1}$ using kpN estimator when N = 8192 77

Figure 5.24 The NSE of $\widehat{TE}_{X_1 \to X_2}$, $\widehat{TE}_{X_1 \to X_3}$, $\widehat{TE}_{X_2 \to X_3}$ $\widehat{TE}_{X_2 \to X_1}$ using kpN estimator when N = 4096 78

Figure 5.25 The NSE of $\widehat{TE}_{X_1 \to X_2}$, $\widehat{TE}_{X_1 \to X_3}$, $\widehat{TE}_{X_2 \to X_3}$ $\widehat{TE}_{X_2 \to X_1}$ using kpN estimator when N = 8192 78

Figure 5.26 The NB and the NSE of the $\hat{h}(X)$ of bistable system using histogram estimator 82

Figure 5.27 The NB and the NSE of the $\hat{h}(X)$ of bistable system using Gaussian KDE 83

Figure 5.28 The NB and NSE of $\hat{h}(X)$ of the bistable system using Epanechnikov KDE 83

Figure 5.29 The NB and NSE of $\hat{h}(X)$ of the bistable system using kNN estimator.. 84

Figure 5.30 The NB and NSE of $\hat{h}(X)$ of the bistable system using kpN entropy estimator 84

Figure 5.31 The expected value and the NMSE error of the entropy estimations 85

Figure 5.32 A real dataset 87

Figure 6.1 The possible coupling schemes between the nodes i, j, and k..... 97

Figure 6.2 Update network paths function is called in Algorithm 1. 105

Figure 6.3 Check_symmetry_rmv, check_source_sink and check_symmetry functions are called in update network paths function. 106

Figure 6.4 Small networks to validate the proposed network reconstruction algorithm 107

Figure 6.5 The cube centered at i, j, k represents the value of $AICE_{i \to j \mid k}$ in volume and color. The largest $AICE_{i \to j \mid k}$ is normalized to 1 unit^3..... 108

Figure 6.6 Illustrative representation of the proposed algorithm for the network in Figure 6.4f 110

Figure 6.7 Randomly selected Erdös-Rényi networks of N = 12 with the probability p = 0.1, p = 0.2, and p = 0.3, respectively. 112

Figure 6.8 The ratio between # of iterations and maximum # of possible iterations of the algorithm with respect to link probability. 112

Figure 7.1 Illustrative cluster synchronization networks..... 119

Figure 7.2 An illustrative cluster synchronization network, where each node represents a Chua system. There are 17 clusters in (a) The connection matrix of (a) has been illustrated in (b). 121

Figure 7.3 The AICE from a_2^{th} system to other systems by observing any system ($AICE_{a_2 \to j|k}$ $\forall$ j, k) for the network in Figure 7.1a. They have been illustrated in a color bar when we are perturbing the node $\mathcal{I} = \{a_2\}$.... 122

Figure 7.4 The AICE from a_3^{th} system to other systems by observing any system ($AICE_{a_3 \to j|k}$ $\forall$ j, k) for the network in Figure 7.1a. They have been illustrated in a color bar when we are perturbing the node $\mathcal{I} = \{a_3\}$.... 123

Figure 7.5 The AICE from a_5^{th} system to other systems by observing any system ($AICE_{a_5 \to j|k}$ $\forall$ j, k) for the network in Figure 7.1a.They have been illustrated in a color bar when we are perturbing the node $\mathcal{I} = \{a_5\}$.... 124

Figure 7.6 The AICE from a_8^{th} system to other systems by observing any system ($AICE_{a_8 \to j|k}$ $\forall$ j, k) for the network in Figure 7.1a. They have been illustrated in a color bar when we are perturbing the node $\mathcal{I} = \{a_8\}$.... 125

Figure 7.7 The AICEs have been represented after the algorithm has been performed when we are perturbing the nodes in a_2, a_3, a_5 and a_8, respectively for the network as shown in Figure 7.1a. 126

Figure 7.8 The AICEs have been represented after the algorithm has been performed when we are perturbing the nodes in a_2, a_4, a_6 and a_7, respectively for the network as shown in Figure 7.1b. 127

Figure 7.9 The AICEs of the network as shown in Figure 7.2a..... 129

Figure 7.10 The coupled chaotic networks 133

Figure 7.11 The errors between the outputs and the CEs of the systems versus time for varying coupling parameter when $i = b$..... 134

Figure 7.12 The errors between the outputs and the CEs of the systems versus time for varying coupling parameter when $i = c$..... 135

Figure 7.13 The errors between the outputs and the CEs of the systems versus time for varying coupling parameter when $i = d$ 136

Figure 7.14 The errors between the outputs and the CEs of the systems versus time for varying coupling parameter when $i = a$ 137

Figure 7.15 The errors between the outputs and the CEs of the systems versus time for varying coupling parameter when $i = f$ 138

LIST OF TABLES

Page

Table 5.1 The significance level $p_{\mathcal{H}_0}$ for the null hypothesis $\mathcal{H}_0$ with $N = 1024$, $N = 2048$ and $N = 4096$ 88

Table 6.1 A summary of the difference of $AICEs$ for the networks in Figure 6.1 102

Table 6.2 Simulation parameters for the networks Figure 6.4 107

Table 7.1 Simulation parameters for the network 7.1b..... 120

Table 7.2 The results of k-means classification algorithm for $AICEs$ as shown in Figure 7.8 128

LIST OF SYMBOLS

α, β	: The symbols of scalars
$\mathcal{L}$	: The Laplacian matrix
$\delta(\cdot)$	: Diract delta function
$\det(\cdot)$	: Determinant
ϵ	: The coupling strength
$\forall$	: For all
κ_i	: The inner coupling matrix
$\mathbb{R}$	: The set of Real numbers
$\mathbb{R}^d$	: d dimensional Real space
$\mathbb{R}^+$	: The set of positive reals
$\mathbb{Z}$	: The set of integers
$\mathbb{Z}^+$	: The set of positive integers
$\mathbf{A}$	: The adjacency matrix or the coupling matrix
$\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$	: The symbols of sets
$\mathcal{E}$	: The edge sets of a graph
$\mathcal{G}$	: A Graph (network)
$\mathcal{V}$	: The vertex set of a graph
$\psi(\cdot)$	: Digamma function
τ	: Argument of the first minimum of the time-delayed MI
$\tilde{p}_x(\cdot)$	: Estimated density
d_e	: Embedding dimension

$E[\cdot]$	: Expectation operator
$H(\cdot)$	: Entropy
$h(\cdot)$	: Differential entropy
$I(\cdot;\cdot)$	: Mutual information
$J_{\mathbf{x}}(\mathbf{x}')$	: Jacobi determinant of $\mathbf{x}$ with respect to $\mathbf{x}'$
$p_x(\cdot)$	: The probability function
$p_{st}(\cdot)$	: The stationary probability density
[a, b]	: Interval on real line

ABBREVIATIONS

pdf	: probability density function
DCM	: Diffusive Coupling Matrix
MSF	: Master Stability Function
MI	: Mutual Information
TE	: Transfer Entropy
CE	: Causation Entropy
OCE	: Optimal Causation Entropy
AICE	: Average Integrated Causation Entropy
kNN	: k-Nearest Neighbor
NB	: Normalized Bias
MSE	: Mean Square Error
NSE	: Normalized Standard Error
NMSE	: Normalized Mean Standard Error
KDE	: Kernel Density Estimator

CHAPTER ONE
INTRODUCTION

The complex networks formed by interactions of identical or similar dynamical systems allow us to model various natural and artificial phenomena. It is not surprising that modeling and analyzing such complex networks in physical, biological, engineering, and social sciences is a subject of ongoing interest. When interacting systems are chaotic, they exhibit unpredictable behavior due to their sensitive dependence on initial conditions and parameters, although they are deterministic systems. The chaotic systems exhibit complex dynamics characterized mainly by their strange attractors. Coupling chaotic systems within a network create even more complex behavior to be tamed.

On the other hand, synchronization is a well-known phenomenon observed in networks of coupled dynamical systems. The term *synchronization* originates from two Greek words, 'chronos' and 'syn' (Pikovsky et al., 2002). These words signify time and same, respectively. To clarify, it is used for things that occur at the same time. The synchronization can happen in nature frequently, e.g., fireflies are flashing their lights synchronously, violinists play their violins simultaneously in an orchestra, or it happens when everybody is clapping at the same time in the audience. The common property of these examples is that they all produce rhythms. Synchronization is related to oscillating objects. Therefore, it is formally defined as *'an adjustment of rhythms of oscillating objects due to their weak interaction'* (Pikovsky et al., 2002).

Loosely speaking, the systems in the network replicate each other's dynamical behavior when synchronized. It was known by the end of the 17th century that loosely coupled periodic systems can synchronize (Huygens, 1673). Yet, it was somewhat surprising, after three centuries later, that chaotic systems could synchronize, too. Interestingly, this could only be understood in late 20th century (Pecora & Carroll, 1990, 1991; Rulkov et al., 1995; Lorenz & Haman, 1996; Fujisaka & Yamada, 1983), only several decades later than the introduction of the chaotic systems (Blekhman, 1964).

Depending on how dynamical behavior is replicated throughout the network, several types of synchronization have been proposed in the literature; *complete* (Pecora & Carroll, 1990), *cluster* or *partial* (Hasler et al., 1998), *lag* (Rosenblum et al., 1997) and *phase* (Rosenblum et al., 1996) synchronization.

When all systems in the network follow the same trajectory asymptotically irrespective of initial conditions of the individual systems, the network is in *complete* synchronization. In *phase* synchronization, there is a ratio between the (generalized) phases of the systems in the network. If a system in the network is synchronized with the other systems in amplitude and phase with a time delay, the situation is defined as *lag* synchronization. If there is more than one cluster (groups) in the network, the cluster members converge to the same trajectory asymptotically. However, if the members of the different clusters do not synchronize with each other; we have *cluster* synchronization. The conditions under which dynamical networks will be synchronous are currently the subject of intense scientific interest. In this thesis, we mainly deal with complete and cluster synchronization.

The synchronization of chaotic systems has been extensively studied in different areas of science (Eroglu et al., 2017). One of the applications of chaotic synchronization is the design of secure communication systems, which are based on transmitting information by using chaotic signals (Hasler, 1995). Moreover, Gambuzza et al. have designed a cluster synchronization network of chaotic Chua systems, ultimately allowing us to model biological networks. The synchronization of chaotic systems can be used to understand large power distribution networks, social networks, and the complex networks of cellular and metabolic structures (Strogatz, 2001).

The synchronization conditions under which a network exhibits a particular type of synchronization have been investigated in the literature extensively. Pecora and Carroll have proposed a master stability function to obtain the stability of the synchronous state of a given network previously (Pecora & Carroll, 1998). Belykh et al. have suggested the connection graph-based stability method that calculates upper

bounds for complete synchronization in coupled chaotic networks (Belykh et al., 2005). Furthermore, the design of synchronous networks guaranteed to be synchronized can be achieved utilizing a controller (Chen et al., 2007; Wu et al., 2008; Liu & Li, 2017). In the controller-based methods, the complete or cluster synchronization is achieved by adding inputs to the selected systems. It has been recently shown that the symmetry in the coupling structure of the network is an essential property to achieve the desired synchronization scheme (Pecora et al., 2014; Gambuzza et al., 2020). Pecora et al. have investigated the determination of cluster synchronization patterns where hidden symmetries exist in the network. The authors have used computational group theory methods to expose symmetries, which cannot be detected at first sight in a large-scale network. They have detected the clusters that can appear in the network by using the transformation matrix that diagonalizes the Laplacian matrix of the network. Determining the transformation matrix is not a straightforward task, and the procedure needs the use of tools of the computational group theory (Pecora et al., 2014).

The stability of the clusters has been analyzed by deriving the variational equations of the systems around the synchronous state. Therefore, the method requires the Laplacian matrix of the network, the transformation matrix, and the variational equations of the systems. To sum up, the design of synchronizing chaotic networks requires the determination of the state equations, the coupling strengths, and the coupling connections of the network.

The design and analysis problems cast aside, in many cases of interest, the underlying dynamical systems in the network are unknown. Although we can gather the observations from the systems' outputs, the observations can only be collected from specific nodes. One of the fundamental problems is to infer useful information about the network structure using observations gathered from only a subset of systems in a network.

The information theory can be used as a model-free approach for determining the network structure using the observed data. The idea of quantifying information flow

between the dynamical systems using the state variables has been around for quite some time (Paluš et al., 2001; Bollt, 2012). The information measures of small-scale networks that consist of coupled logistic maps in complete synchronization have been studied by Sun and Bollt. The authors have examined whether three logistic maps in the complete synchronization are coupled directly or not by estimating the transfer entropies (Sun & Bollt, 2014). However, analysis of dynamical networks in complete or cluster synchronization via information measures poses some important, open, and challenging questions.

The extensive reviews on network reconstruction have been presented in (Timme & Casadiego, 2014; Wang et al., 2016; Wu & Jiang, 2021). The network reconstruction can be done by implementing various ways, from controller-based methods to machine learning methods and from statistical perspective to information-theoretic measures (Liu et al., 2022; Panaggio et al., 2019; Leguia et al., 2019; Kuzmanovski et al., 2018; Novelli & Lizier, 2021). However, the network reconstruction in a coupled synchronized network is very rare in the literature (Chen et al., 2009).

Up to our knowledge, the cluster synchronization of chaotic networks has not been studied through information-theoretic approaches and the information flow between the cluster nodes in a complex network has not been discussed in the literature. Investigation of synchronization from the perspective of information theory leads us to new questions:

i) "How to quantify the relation between synchronization and information transfer ?",

ii) "Is there any information transfer between synchronized groups ?",

iii) "How to estimate the information transfer between coupling dynamical systems ?",

iv) "Can we reconstruct the network via information-theoretic tools? ",

v) "Can we detect synchronization using the information measures ?",

vi) "What will be the information flow when the coupling strength changes in the network ?", ...

In this thesis, we have mainly dealt with the problem of revealing the unknown properties of synchronized networks utilizing limited observations, assuming that the underlying dynamics and connections of the network are unknown. However, synchronization can be an obstacle to inferring desired properties of the network. The problem becomes especially hard because the observations from synchronizing nodes are eventually identical and do not reveal much about the network properties.

1.1 Thesis Statement and Contributions

In this thesis, we have focused on solving two major problems; the unfolding of the structure of the directed chaotic networks using information theory and the detection of cluster synchronization in chaotic networks via information-theoretic measures.

First, the reconstruction of the complex network structure from observed data has been examined. The main problem is to infer or reconstruct the unknown network coupling scheme using observations from only a subset of individual systems in a network. In other words, we have focused on the hidden network structure as in *a black box* and revealed only from the outputs of the individual dynamics.

We assume that the networks are unidirectional, chaotic, and synchronized. Although the number of systems in a network is known, we do not observe all the systems' outputs. We only observe a single output from individual dynamics. We assess the states of the individual systems and can add input to the systems at specific times. There is no a priori knowledge about the underlying dynamics, the coupling strength, and coupling connections.

Under the assumptions mentioned above, we have solved the network reconstruction problem by perturbing the systems and estimating information-theoretic measures. The perturbations disturb the synchronization and

excite the network by injecting new information into the system. The connection information is revealed as the injected information flows through the network en route to the re-synchronization. The state space can be reconstructed by observing only the system outputs using Takens' embedding theorem to estimate information theoretic measures. As a promising information-theoretic measure, we have estimated the causation entropies between the subsystems using k-nearest neighbor entropy estimator (Kozachenko & Leonenko, 1987; Kraskov et al., 2004). For this purpose, we have suggested a k-nearest neighbor entropy estimator for estimating the causation entropy effectively. Then, we have estimated average integrated causation entropies (i.e., firstly proposed in this thesis) between the subsystems using the proposed estimator.

The central hypothesis is that the directly connected nodes contribute to the average integrated causation entropies less than the non-directed nodes. The difference in average integrated causation entropies between the reconstructed outputs can be used to reveal the paths and, therefore, can be used to reconstruct the connection matrix. We have analytically shown that the difference of this measure for the possible coupling schemes between the three related systems can be distinguished. Accordingly, we have proposed novel theorems based on the average integrated causation entropy difference. We developed an algorithm based on the properties of the difference in average integrated causation entropy based on these theorems. The proposed algorithm has been verified with small-size and randomly coupled networks comprising Chua systems.

The second aim is to analyze information transfer among the same or different cluster groups in case of cluster synchronization. As a first step, our problem is to determine clusters in a chaotic network by estimating information measures.

We assume that each system in the network can be observed via a single output. Besides, the individual dynamics of the nodes and the number of clusters are unknown in the network. Additionally, to observe the formation of clusters, we can perturb the systems in the network individually at specific times.

As an interesting result, estimating information measures via sequential observation vectors is insufficient to determine cluster synchronization directly. The state space reconstructed by observing only the system outputs by means of Takens' embedding theorem in delay coordinates is much more revealing to detect clusters via estimated information measures (Takens, 1981).

We have started by estimating the causation entropies in the reconstructed state space to determine the clusters in an unknown network. Then, we have defined and estimated the integrated causation entropy (ICE) and its expected value to detect the existence of cluster synchronization in the network. The results show that the average integrated causation entropies (AICEs) can be used to determine the number of clusters and classify whether the individual systems belong to a particular cluster or not effectively.

We have demonstrated that we could obtain clusters in the coupled chaotic networks of Chua's systems by estimating the AICEs. Then, we have clustered obtained AICE values using the basic k-means classification algorithm.

1.2 Outline

The thesis is organized as follows. In Chapter 2, we have represented the necessary mathematical tools, the fundamental theorems and concepts of the state-space reconstruction, the synchronization, and the entropy. We have summarized the related synchronization concepts and the synchronization techniques of dynamical systems in Chapter 3. We have introduced the information-theoretic tools existing in the literature in Chapter 4. Therein, the theoretical values of the information-theoretic measures in case of synchronization have been derived briefly.

Chapter 5 focuses on the commonly-used entropy estimation techniques that exist in the literature. The performance of the entropy measures has been investigated on linear and nonlinear systems as well as the real dataset to obtain estimators' optimal parameters to quantify information flow accurately.

In Chapter 6, we have introduced our main result, the novel approach to unfolding the structure of chaotic dynamical networks using causation entropy. We have presented a method for reconstructing unidirectional chaotic synchronized networks by injecting random information via impulsive perturbations into the individual systems. The proposed method is based on the average integrated causation entropy. Besides, an algorithm has been introduced that reveals the network structure by using average integrated causation entropy between the systems. The performance of the proposed algorithm has been illustrated in synchronized networks.

Chapter 7 introduces the detection of clusters via information measures in continuous coupled chaotic networks. We have shown that the causation entropies estimated from observations can be used to determine the number of synchronous clusters and membership information of the network elements in a chaotic network. Therefore, the average integrated causation measure can detect the clusters. We propose a novel algorithm to distinguish the clusters in the network. The proposed procedure has been validated on various chaotic networks.

Chapter 8 concludes the thesis; the main contributions have been summarized, and future works have been discussed briefly.

CHAPTER TWO
MATHEMATICAL BACKGROUND

Herein, we have introduced the necessary definitions and theorems used in the sequel. We have begun with the necessary mathematical definitions of graph theory and follow up with the definitions of the systems, complex networks, and synchronization required to follow the upcoming chapters. Also, the state space reconstruction techniques and Shannon's entropy definition have been reviewed.

2.1 Graph Theoretic Definitions

Graphs are objects that are used to represent networks naturally. A *graph* is a mathematical object that has connections (links) between various objects (Latora et al., 2017). The existence and the absence of connections can be represented between the various objects in terms of graphs. Thus, we need to define a set of elements, the graph *nodes*, and a set of *links* between the nodes.

Definition 2.1.1 (Undirected Graph, (Latora et al., 2017))**.** A graph $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$, consists of two sets $\mathcal{V} \neq \emptyset$ and $\mathcal{E}$. Here, the elements of $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ are defined as *vertices* or nodes and are distinct. The elements of $\mathcal{E}$, consist of the pairs of elements of $\mathcal{V}$ ($\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$), are defined as edges or links. The link between the node v_i and v_j is denoted with the tuple $(i, j) \in \mathcal{E}$. In a graph, particularly an undirected graph, if there is a link between i and j, the graph also has the vertex of (j, i).

Definition 2.1.2 (Directed Graph, (Latora et al., 2017))**.** A *directed* graph also $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$, consists of two sets $\mathcal{V} \neq \emptyset$ and $\mathcal{E}$. The vertices are $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$. The elements of $\mathcal{E}$, consist of the pairs of elements of $\mathcal{V}$ ($\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$), are defined as links. The link between the node v_i and v_j is denoted with the tuple $(i, j) \in \mathcal{E}$. In a directed graph, if (i, j) symbolizes the link from i to j, the link (j, i) may not necessarily exist.

The *connectivity* information of a graph can be defined in terms of the adjacency matrix defined below.

Definition 2.1.3 (Adjacency Matrix)**.** The adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ of a network whose elements a_{ij} consist of ones or zeros such that

$$a_{ij} = \begin{cases} 1, & \text{iff } (j,\ i) \in \mathcal{E} \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

The nodes can be connected through several links. The following definition of degree will prove useful in presenting several results in the upcoming discussions.

Definition 2.1.4 (Node degree)**.** Let $\mathbf{A} \in \mathbb{R}^{N\times N}$ be the adjacency matrix associated with $\mathcal{G}$ and the in–degree and out–degree of v_i be denoted and k_i^{in}, k_i^{out}. Clearly,

$$k_i^{in} = \sum_{j=1, j\neq i}^{N} a_{ij}, \qquad k_i^{out} = \sum_{j=1, j\neq i}^{N} a_{ji}. \tag{2.2}$$

The total degree of v_i is defined as $k_i = k_i^{in} + k_i^{out}$.

Related to these concepts, another mathematical structure that relays the connectivity information is the *Laplacian matrix*, which plays an important role in the design of synchronous networks.

Definition 2.1.5 (Laplacian Matrix)**.** The Laplacian matrix $\mathcal{L} \in \mathbb{R}^{N\times N}$ of a directed network whose elements $\mathcal{L}_{ij}$, is given by:

$$\mathcal{L}_{ij} = \begin{cases} -a_{ij}, & \text{if } i \neq j \\ k_i^{out}, & \text{if } i = j. \end{cases} \tag{2.3}$$

Briefly, the Laplacian matrix is then $\mathcal{L} = \text{diag}\left\{k_1^{out}, k_2^{out}, \ldots, k_N^{out}\right\} - \mathbf{A}$.

Next, we define *isomorphism* to compare graphs of identical size and order.

Definition 2.1.6 (Isomorphism (Latora et al., 2017))**.** Let $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ be two graphs with the same size and order. $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic if there

is a bijection $\Phi : \mathcal{G}_1 \to \mathcal{G}_2$, such that $(i, j) \in \mathcal{E}_1$ if and only if $(\Phi(i), \Phi(j)) \in \mathcal{E}_2$. Then, the bijection is called an isomorphism.

2.2 Dynamical Networks

In a general setting, the basic elements of a dynamical network consist of a stochastic system in terms of a stochastic differential equation.

Definition 2.2.1 (Stochastic System)**.**

$$d\mathbf{x}(t) = f(\mathbf{x}, t)dt + g(\mathbf{x}, t)d\mathbf{W}(t) \tag{2.4}$$

with initial condition $\mathbf{x}(0) = \mathbf{x}_0$. Here, $\mathbf{x}(t) = [x^1(t), x^2(t), \ldots, x^d(t)]^T \in \mathbb{R}^d$ is the state vector. $f(\mathbf{x}, t)$ is drift-term and $g(\mathbf{x}, t)$ is diffusion term of the stochastic system. $\mathbf{W}(t)$ is a vector of independent Wiener processes.

Each node v_i of a graph can be associated with an identical chaotic dynamical system that connects and interacts with the other systems via the links (edges) of $\mathcal{G}$. More precisely, the nodes of the state equations of the networks are given in the sequel.

Definition 2.2.2 (Continuous-time complex network)**.** N non-identical systems can be coupled to yield a system of stochastic differential equations of the form:

$$\begin{aligned} d\mathbf{x}_i(t) &= f_i(\mathbf{x}_i(t), \mathbf{u}_i, t)dt + g_i(\mathbf{x}_i(t), \mathbf{u}_i, t)d\mathbf{W}_i(t) + \sum_{j=1}^{N} \epsilon_{ij}(t)a_{ij}\kappa_i h_i(\mathbf{x}_i(t), \mathbf{x}_j(t), t)dt \\ \mathbf{x}_{i,t_0} &= \mathbf{x}_i(t_0) \\ \mathbf{y}_i(t) &= \phi_i(\mathbf{x}_i(t), \mathbf{u}_i, t, \eta) \quad i = 1, 2, \ldots, N \end{aligned} \tag{2.5}$$

where $\mathbf{x}_i(t) = [x_i^1(t), x_i^2(t), \ldots, x_i^{d_i}(t)]^T \in \mathbb{R}^{d_i}$ is the state vector of ith system, $\mathbf{u}_i \in \mathbb{R}^{r_i}$ is

the input of ith system, $f_i : \mathbb{R}^{d_i} \times \mathbb{R}^{r_i} \times \mathbb{R}^+ \rightarrow \mathbb{R}^{d_i}$ represents the local dynamic of ith system, $g_i : \mathbb{R}^{d_i} \times \mathbb{R}^{r_i} \times \mathbb{R}^+ \rightarrow \mathbb{R}^{d_i}$ describes coefficient of noise which depends on the value of state variables and input, $d\mathbf{W}_i(t) \in \Omega^{d_i}$ and $d\mathbf{W}_i(t) = \xi_i(t)dt$ denotes Wiener process, $\xi_i(t)$ denotes a noise process of ith system. $\epsilon_{ij}(t)$ is time-varying coupling strength from j^{th} system to i^{th} system, and it is represented in matrix form $\boldsymbol{\epsilon}(t) = [\epsilon_{ij}(t)]$ $\in \mathbb{R}^{N\times N}$. $\boldsymbol{\kappa}_i$ is inner coupling matrix, $h(\mathbf{x}_i(t), \mathbf{x}_j(t), t) : \mathbb{R}^{d_j} \times \mathbb{R}^+ \rightarrow \mathbb{R}^{d_i}$ is coupling function. $\mathbf{y}_i(t) = [y_{i1}(t), y_{i2}(t), \ldots, y_{im_i}(t)]^T \in \mathbb{R}^{m_i}$ is the output function of ith system and $\phi_i : \mathbb{R}^{d_i} \times \mathbb{R}^{r_i} \times \mathbb{R}^+ \times \Omega^{d_i} \rightarrow \mathbb{R}^{m_i}$ is observation function and $\eta \in \Omega^{d_i}$ is vector of unknown parameters. The observation function is unknown; the output can be dispersed, filtered, or distorted with noise.

If noise does not affect individual node dynamic, $g_i(\cdot)$ is zero in (2.5) and it reduces to:

$$\begin{aligned}
\dot{\mathbf{x}}_i(t) &= f_i(\mathbf{x}_i(t), t) + \sum_{j=1}^{N} \epsilon_{ij}(t) a_{ij} \kappa_i h_i(\mathbf{x}_i(t), \mathbf{x}_j(t), t) \\
\mathbf{x}_{i,t_0} &= \mathbf{x}_i(t_0) \\
\mathbf{y}_i(t) &= \phi_i(\mathbf{x}_i(t), t, \eta) \quad i = 1, 2, \ldots, N.
\end{aligned} \tag{2.6}$$

In discrete time, we can define the network using the difference equations as follows:

Definition 2.2.3 (Discrete-time complex network)**.**

$$\begin{aligned}
\mathbf{x}_i[n+1] &= f_i(\mathbf{x}_i[n]) + \sum_{\substack{j=1 \\ i\neq j}}^{N} \epsilon_{ij} a_{ij} \boldsymbol{\kappa}_i h_i(\mathbf{x}_i[n], \mathbf{x}_j[n]) + \mathbf{u}_i[n], \\
\mathbf{x}_{i,0} &= \mathbf{x}_i[n_0] \\
\mathbf{y}_i[n] &= r_i(\mathbf{x}_i[n], \eta[n]) \quad i, j = 1, 2, \ldots, N.
\end{aligned} \tag{2.7}$$

Here, $\mathbf{x}_i[n] \in \mathbb{R}^{d_s}$ is the state of dynamical system ν_i at time index n, $f_i : \mathbb{R}^{d_s} \times \mathbb{R}^{d_i} \rightarrow \mathbb{R}^{d_s}$

represents the inner dynamics of each system and $h_i : \mathbb{R}^{d_s} \to \mathbb{R}^{d_s}$ is the coupling function, and $\mathbf{u}_i(t) \in \mathbb{R}^{d_s}$ is the input of ν_i. We assume that f_i and h_i are sufficiently differentiable to ensure the existence and uniqueness of the solutions. ϵ_{ij} is the coupling strength, $\boldsymbol{\kappa}_i$ is the inner coupling matrix, $\mathbf{y}_i[n] \in \mathbb{R}^m$ is the output of ν_i at time n and $r_i : \mathbb{R}^{d_s} \to \mathbb{R}^m$ represents the readout function, $\eta \in \Omega$ is vector of unknown parameters and Ω is the related sample space.

In the rest of the thesis, we will use the symbol $\mathcal{G}$ to refer (2.5), (2.6) or (2.7), loosely. We will assume that the outputs are sampled at a proper rate.

Although we observe the sampled outputs of the network (2.5), (2.6) or (2.7), it is not required to use each sequential observation, in fact, down sampling the outputs will prove more useful. Therefore, it is generally preferred to resample the outputs of the individual system using an optimal delay. This issue will be discussed later in Section 2.3.

We will additionally assume that the systems $\dot{\mathbf{x}}_i(t) = f_i(\mathbf{x}_i(t), t)$ in (2.6) is *chaotic* in the sense of Wiggins (Wiggins, 2003).

Definition 2.2.4 (Chaos)**.** One widely accepted definition describes *chaos* as "aperiodic long-term behavior in a deterministic system that exhibits sensitive dependence on initial conditions" (Strogatz, 2018). Mathematically, a dynamical system is chaotic in an invariant set $\mathcal{X}$, if

i) it is topologically transitive in $\mathcal{X}$,

ii) it is sensitive to initial conditions,

iii) the set of periodic orbits of it is dense in $\mathcal{X}$.

One of the first examples of such systems is the well-known system due to Lorenz, that models simplified atmospheric interactions, has shown to have aperiodic behavior, which is extremely sensitive to initial conditions (Lorenz, 1963). The state space equation of this system is given by (2.8):

Figure 2.1 Lorenz attractor for $\sigma = 16$, $b = 4$ and $r = 45.92$.

$$\begin{aligned} \dot{x_1} &= \sigma(x_2 - x_1) \\ \dot{x_2} &= -x_1 x_3 + r x_1 - x_2 \\ \dot{x_3} &= x_1 x_2 - b x_3 \end{aligned} \tag{2.8}$$

Here σ, b, r are real parameters. Lorenz's attractor is illustrated in Figure 2.1.

Another well-studied chaotic dynamical system, Chua's circuit, is widely used in many practical situations. The attractor is easily implemented using simple electronic components shown as Figure 2.2a and has many variations (Chua et al., 1986; Morgül, 1995; Cruz & Chua, 1992; Fortuna et al., 2009).

The state space representation of Chua's circuit is given as follows :

(a) Chua's Circuit

(b) $i - v$ relation of the nonlinear resistor i_{R_2}

Figure 2.2 Chua's circuit and its nonlinear characteristics

$$
\begin{aligned}
C_1 \frac{dV_{C_1}}{dt} &= G_1[V_{C_2} - V_{C_1}] - i_{R_2}(v_{R_2}) \\
C_2 \frac{dV_{C_2}}{dt} &= G_1[V_{C_1} - V_{C_2}] - i_L \\
L \frac{dV_{i_L}}{dt} &= +V_{C_2}
\end{aligned}
\tag{2.9}
$$

where $i_{R_2}(v_{R_2}) = G_a v_{R_2} + 0.5(G_a - G_b)((v_{R_2} + E) - (v_{R_2} - E))$ is the equation of nonlinear resistor and its $i - v$ relation is given in Figure 2.2b.

If we transform variables as in

$$
\begin{aligned}
x_1 &= V_{C_1}/E, \quad x_2 = V_{C_2}/E, \quad x_3 = i_L/EG_1, \quad \alpha = C_2/C_1 \\
m_0 &= G_b/G_1, \quad m_1 = G_b/G_1, \quad \tau = tG_1/C_2, \quad \beta = C_2/LG_1^2
\end{aligned}
\tag{2.10}
$$

the normalized equations of Chua's circuit become,

$$
\begin{aligned}
\dot{x_1} &= \alpha(x_2 - h(x_1)) \\
\dot{x_2} &= x_1 - x_2 + x_3 \\
\dot{x_3} &= -\beta x_2
\end{aligned}
\tag{2.11}
$$

where

Figure 2.3 Chua Attractor

$$h(x_1) = \begin{cases} m_1 x_1 + (m_0 - m_1) & x_1 > 1 \\ m_0 x_1 & |x_1| \leq 1 \\ m_1 x_1 + (m_1 - m_0) & x_1 < -1 \end{cases} \tag{2.12}$$

The Chua's circuit has an attractor known as *double scroll* for the parameters $\alpha = 9$, $\beta = 14.28$, $\gamma = 0$, $m_0 = -1/7$, $m_1 = 2/7$ and it is known to be chaotic (Chua et al., 1986). The associated attractor for the chosen parameters is plotted in Figure 2.3.

A possible implementation of the Chua circuit is also given by:

$$\begin{aligned} \dot{x_1} &= \alpha(x_2 - h(x_1)) \\ \dot{x_2} &= x_1 - x_2 + x_3 \\ \dot{x_3} &= -\beta x_2 - \gamma x_3 \end{aligned} \tag{2.13}$$

where $m_0 = -1.27$, $m_1 = -0.68$, $\alpha = 10$, $\beta = 15$, $\gamma = 0.0385$. It also satisfies all properties of chaotic systems for these parameters (Fortuna et al., 2009).

2.3 State Space Reconstruction

Consider a dynamical system $\dot{\mathbf{x}}_i(t) = f_i(\mathbf{x}_i(t), t)$ in (2.6) with an initial value $\mathbf{x}_{i,t_0} = \mathbf{x}_i(t_0)$. After transients die out, the system's trajectory converges to an attractor $\mathcal{A}$. Let $\mathcal{M}$ be a manifold and $\mathcal{M} \subset \mathcal{A}$. Assume that $\mathcal{A}$ is a bounded and invariant set. The system has dense trajectories if it is chaotic according to definition 2.2.4. The sequential observations can contain squeezed information about the future states of the dynamics. Thus, densely sampled outputs are redundant and do not yield much additional information. On the other hand, when we sample the observation too infrequently, the observations do not provide any information about the dynamics and become irrelevant. The problem can be resolved using the state space reconstruction techniques (Takens, 1981; Casdagli et al., 1991; Kantz & Schreiber, 2004; Uzal et al., 2011; Krämer et al., 2021). The main goal of the reconstruction is to present the state dynamics by reducing the correlation between the observations and preserving the important invariants of the dynamics.

Takens proved that even single output from the system could be used to reconstruct the state space in the embedding theorem (Takens, 1981).

Theorem 2.3.1 (Takens embedding theorem)**.** *Let $\mathcal{M}$ be a d-dimensional manifold. For pairs of (Φ, h), $\Phi : \mathcal{M} \to \mathcal{M}$ be a smooth diffeomorphism and $h : \mathcal{M} \to \mathbb{R}$ a smooth function, it is a generic property that the observation map $\Psi : \mathcal{M} \to \mathbb{R}^{T_d}$ is an immersion and becomes*

$$\Psi(\mathbf{x}(t)) = [h(\mathbf{x}(t)), h(\Phi(\mathbf{x}(t))), \ldots, h(\Phi^{T_{d-1}}(\mathbf{x}(t)))] \tag{2.14}$$

where $T_d = 2d_e + 1$ is the dimension of the reconstructed attractor and $d_e \in \mathbb{Z}^+$.

A delay coordinate map (time-delayed embedding) is based on Takens embedding theorem in 2.3.1. Let h be chosen as $y(t) = h(\mathbf{x}(t))$ a smooth measurement function. Φ is selected by τ delayed measurements, i.e., $\Phi^{T_k}(\mathbf{x}(t)) = \mathbf{x}(t + T_k) = \mathbf{x}(t + k\tau)$ where $k \in \mathbb{Z}^+$. Then, a delay coordinate map that reconstructs the state space is defined as:

$$\tilde{\mathbf{x}}(t) = [y(t), y(t+\tau), y(t+2\tau), \ldots, y(t+(d_e-1)\tau)] \tag{2.15}$$

where $\tau > 0$ and $d_e \in \mathbb{Z}^+ > 1$ represent the delay and the embedding dimension, respectively. τ and d_e is to be determined in such a way that while the topological properties of the attractor are preserved, the correlation between the sequential observations is minimized. One of the many ad hoc techniques exploits the mutual information between the sequential observations in which $\tau > 0$ is chosen to be the argument of the first minimum of the time-delayed mutual information described by:

$$I(y(t), y(t+\tau)) = \sum_{y(t),y(t+\tau)} p(y(t), y(t+\tau)) \log\left[\frac{p(y(t), y(t+\tau))}{p(y(t))p(y(t+\tau))}\right]. \tag{2.16}$$

Here, $y(t)$ is the observation vector at time step t, $y(t+\tau)$ is the delayed observation vector and $p(\cdot)$ is the probability distribution function.

The correlation between the observation vectors will be reduced if observation vectors are resampled by τ instead of sequentially sampling observations. Sequential observations do not provide any new information for the immediate future because of the continuous dynamics of the attractor. On the other hand, if the observations are resampled too infrequently, they do not represent any information about dynamics. Hence, there is no correlation between observation vectors. However, if the resampling duration between observation vectors is chosen as the first local minimum time-lag τ of mutual information, the amount of information inferred from observations increases. Furthermore, there is a remarkable speed-up when estimating the information measures at the reconstructed state space due to the down-sampling of the data.

d_e is obtained by minimizing the false nearest neighbors of the states in the reconstructed space (Abarbanel, 2012). The false nearest neighbor ratio between the observation vectors should be computed to estimate the embedding dimension. If the

dimension is less than the embedding dimension, the points that are not real neighbors to each other may appear very close. The observation points and the false neighbors differ from each other at the dimension that is equal to or greater than the embedding dimension. The point and neighbor candidates are defined at d-dimension as:

$$\begin{aligned} \mathbf{y}(t) &= [y(t), y(t+\tau), \ldots, y(t+(d-1)\tau)] \\ \mathbf{y}^{NN}(t) &= [y^{NN}(t), y^{NN}(t+\tau), \ldots, y^{NN}(t+(d-1)\tau)] \end{aligned} \tag{2.17}$$

The distance is computed between $\mathbf{y}(t)$ and $\mathbf{y}^{NN}(t)$ in d and $d+1$ dimensions as follows:

$$R_d(t)^2 = \sum_{m=1}^{d} [y(t+(m-1)\tau) - y^{NN}(t+(m-1)\tau)]^2. \tag{2.18}$$

$$\begin{aligned} R_{d+1}(t)^2 &= \sum_{m=1}^{d+1} [y(t+(m-1)\tau) - y^{NN}(t+(m-1)\tau)]^2 \\ &= R_d(t)^2 + |y(t+d\tau) - y^{NN}(t+d\tau)|^2. \end{aligned} \tag{2.19}$$

The relative distance is given by:

$$\sqrt{\frac{R_{d+1}(t)^2 - R_d(t)^2}{R_d(t)^2}} = \frac{|y(t+d\tau) - y^{NN}(t+d\tau)|}{R_d(t)} \tag{2.20}$$

or

$$\frac{|y(t+d\tau) - y^{NN}(t+d\tau)|}{R_A(t)} \tag{2.21}$$

where R_A is the diameter of the attractor.

If the relative distance is more significant than a threshold, then $\mathbf{y}^{NN}(t)$ is the false

neighbor of $\mathbf{y}(t)$ at d-dimension. The dimension where the percentage of false nearest neighbor drops to zero is the embedding dimension (Abarbanel, 2012).

Figure 2.4 The state space reconstruction of the chaotic attractor. (a) the trajectory of the unknown chaotic dynamical system. (b) The output of a chaotic dynamical system. (c) The trajectory in the reconstructed state space. (d) The blue points represent one of the sampled outputs of the reconstructed state space waveform while the gray line indicates the actual output.

Figure 2.4 illustrates the state space reconstruction technique. The outputs of a system are observed through the readout function as $\mathbf{y}(t) = r(\mathbf{x}(t), \eta)$. With large enough d_e, the existence of a smooth transformation $\tilde{\mathbf{x}}(t) = \mathbf{\Psi}(\mathbf{x}(t))$ is guaranteed, and the crucial topological properties of the attractor is preserved (Takens, 1981).

Let $\boldsymbol{\phi} : \mathbb{R}^{d_e} \rightarrow \mathbb{R}^{m_r}$ be the unknown read out function in the reconstructed space. The reconstructed output $\tilde{\mathbf{y}}_r(t) = \boldsymbol{\phi}(\tilde{\mathbf{x}}(t))$ can be formed by considering the possibilities of the noise, non-uniform delay coordinates, or a noise suppression algorithm. We choose one of outputs $\tilde{y}_r(t)$ from the reconstructed state space, and sample it in every

τ seconds, $\tilde{y}_r(t_0 + n\tau) = \tilde{y}_r[n] = \tilde{y}_{r,n}$ and $r = 1, 2, \ldots, m_r$ where m_r is the number of the reconstructed outputs of the system and $n \in \mathbb{Z}^+$, $t_0 \in \mathbb{R}^+ \cup \{0\}$.

2.4 Synchronization

Suppose that $f_i's$ in (2.6) are chaotic according to definition 2.2.4. Under suitable conditions of $\epsilon_{ij}(t)$, the network in (2.6) can be synchronized.

Definition 2.4.1 (Complete state synchronization)**.** The complete synchronization of network (2.6) occurs if

$$\begin{aligned} &\lim_{t\to\infty} \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| = 0 \quad i \neq j \\ &\lim_{t\to\infty} \frac{d\mathbf{x}_i(t)}{dt} \not\equiv 0 \quad \forall \quad i,\ j = 1, 2, \ldots, N. \end{aligned} \tag{2.22}$$

In complete synchronization, all network members follow the same trajectory asymptotically.

Definition 2.4.2 (Almost sure output complete synchronization)**.** The outputs of a complex network in (2.4) synchronize almost surely if

$$P\left\{\lim_{t\to\infty} \|\mathbf{y}_i(t) - \mathbf{y}_j(t)\| = 0\right\} = 1 \quad i \neq j \quad \forall \quad i = 1, 2, \ldots, N \tag{2.23}$$

where $P(\cdot)$ is the probability of the event.

Definition 2.4.3 (Cluster state synchronization)**.** Consider N systems form m clusters in the network in (2.6). The sets of these clusters are denoted as $G_1,\ G_2, \ldots,\ G_m \subset \mathcal{I}$ are mutually exclusive where $G_1 = \{1, 2, \ldots, r_1\}$, $G_2 = \{r_1 + 1, r_1 + 2, \ldots, r_2\}$, $\ldots$, $G_m = \{r_{m-1} + 1, r_{m-1} + 2, \ldots, r_m\}$, $r_1 \leq r_2 \leq \ldots \leq r_m$, the cluster synchronization occurs if

$$
\begin{aligned}
&\lim_{t\to\infty} \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| = 0 \qquad i, j \in G_{\mathcal{I}} \qquad i \neq j \\
&\lim_{t\to\infty} \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| \neq 0 \qquad i, j \notin G_{\mathcal{I}} \\
&\lim_{t\to\infty} \frac{d\mathbf{x}_i(t)}{dt} \not\equiv 0 \quad \forall \quad i = 1, 2, \ldots, N.
\end{aligned}
\tag{2.24}
$$

Here, $\mathcal{I}$ is the index set of an arbitrary cluster in the network. In cluster synchronization, the systems in the same cluster synchronize asymptotically, and there is no synchronization between the systems in different clusters.

Definition 2.4.4 (Almost sure output cluster synchronization)**.** Consider we have a complex network in (2.4), the outputs achieve almost sure cluster synchronization if

$$
\begin{aligned}
&P\left\{\lim_{t\to\infty} \|\mathbf{y}_i(t) - \mathbf{y}_j(t)\| = 0\right\} = 1 \quad i, j \in G_I \quad i \neq j \\
&P\left\{\lim_{t\to\infty} \|\mathbf{y}_i(t) - \mathbf{y}_j(t)\| \neq 0\right\} = 1 \quad i, j \notin G_I.
\end{aligned}
\tag{2.25}
$$

Assume that we have a network in (2.6). In general, in order to determine synchronization between two systems i and j, the synchronization error between the systems is defined as:

$$
e_{i,j}(t) = \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| \tag{2.26}
$$

Alternatively, the global synchronization error of a network is defined as:

$$
E(t) = \sqrt{\left(\frac{1}{N}\sum_{i,j} \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|^2\right)} \tag{2.27}
$$

2.5 Entropy

In this section, we have given definitions of the entropy in discrete and continuous-time random variables (Cover & Thomas, 2012). Shannon entropy measures the uncertainty about the outcome of an experiment by observing the random variable (Shannon, 1948). Clearly, the entropy quantifies the amount of information needed to get complete information about the outcome.

Consider a random vector $\mathbf{X} \in \mathbb{R}^d$ that has an underlying probability space. Let $\mathcal{S}_x$ be a support set of $\mathbf{X}$. Assume that $\mathcal{S}_x$ is countable and finite. In other words, $\mathcal{S}_X = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_K\}$, where K is a positive integer. Let $p_{\mathbf{X}}(\mathbf{x})$ be the probability distribution over $\mathcal{S}_{\mathbf{X}}$. A measure of a decrease of uncertainty by an observation drawn from $\mathcal{S}_{\mathbf{X}}$ is $\log \dfrac{1}{p_{\mathbf{X}}(\mathbf{x})}$ and it is called the entropy of $\mathbf{X}$. Formally, the entropy of $\mathbf{X}$ is computed by taking the expectation value over the outcomes:

$$H(\mathbf{X}) = E\left[\log \frac{1}{p_{\mathbf{X}}(\mathbf{x})}\right] = -E\left[\log p_{\mathbf{X}}(\mathbf{x})\right] \tag{2.28}$$

where $p_{\mathbf{X}}(\mathbf{x}) = \sum_{i=1}^{K} p_i \delta(\mathbf{x} - \mathbf{x}_i)$. The probality of drawing $\mathbf{x}_i$ is $p_i = P(\mathbf{x} = \mathbf{x}_i)$, $\sum_{i=1}^{K} p_i = 1$. Here, we use the convention that $0 \log(0) = 0$.

Definition 2.5.1. The (course-grained) entropy of a discrete random variable $\mathbf{X}$ is defined as:

$$H(\mathbf{X}) = -\sum_{\mathcal{S}_{\mathbf{X}}} p_{\mathbf{X}}(\mathbf{x}) \log p_{\mathbf{X}}(\mathbf{x}) = -\sum_{i=1}^{K} p_i \log p_i. \tag{2.29}$$

The entropy is measured in terms of *bits* if the base of the logarithm is 2 and in *nats* if the base of the logarithm is the Euler number e.

Lemma 2.5.2. $H(\mathbf{X}) \geq 0$.

Proof: $0 \leq p_{\mathbf{X}}(\mathbf{x}) \leq 1$ implies that $-\log(p_{\mathbf{X}}(\mathbf{x})) \geq 0$.

Definition 2.5.3. The joint entropy of a pair of discrete random variables (**X**,**Y**) is given as:

$$H(\mathbf{X}, \mathbf{Y}) = -\sum_{\mathcal{S}_{\mathbf{X}}} \sum_{\mathcal{S}_{\mathbf{Y}}} p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) \log p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}). \tag{2.30}$$

Here, $p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y})$ denotes the joint probability function of (**X**, **Y**).

Definition 2.5.4. If (**X**, **Y**) has the joint probability function $p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y})$, the conditional entropy of $H(\mathbf{Y} \mid \mathbf{X})$ is defined as:

$$H(\mathbf{Y} \mid \mathbf{X}) = -\sum_{\mathcal{S}_{\mathcal{X}}} \sum_{\mathcal{S}_{\mathcal{Y}}} p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) \log p_{\mathbf{XY}}(\mathbf{y} \mid \mathbf{x}). \tag{2.31}$$

The well-known identities of the entropy are listed below:

i) $H(\mathbf{X}, \mathbf{Y}) = H(\mathbf{X}) + H(\mathbf{Y} \mid \mathbf{X})$ (Chain rule) (2.32)

ii) $H(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n) = \sum_{i=1}^{n} H(\mathbf{X}_i \mid \mathbf{X}_{i-1}, \ldots, \mathbf{X}_1)$ (Chain rule for multiple random variables) (2.33)

iii) $H(\mathbf{X} \mid \mathbf{Y}) \leq H(\mathbf{X})$ (Conditioning reduces entropy) (2.34)

iv) $E[f(\mathbf{X})] \geq f(E[\mathbf{X}])$ if $f(\cdot)$ is a convex function (Jensen's inequality) (2.35)

v) $H(g(\mathbf{X})) \leq H(\mathbf{X})$ (Entropy of functions of a random variable) (2.36)

Theorem 2.5.5. *The coarse-grained entropy is preserved under the homeomorphisms, i.e., continuous and invertible transformations with continuous inverses.*

Proof: Let **X** and **Y** be two discrete random variables and $\mathbf{Y} = g(\mathbf{X})$ be a homeomorphism. Then, for every $\mathbf{y}_i \in \mathcal{S}_Y$, $p_{\mathbf{Y}}(\mathbf{y}) = \sum_{\mathbf{x}:g(\mathbf{x})=\mathbf{y}} p_{\mathbf{x}}(\mathbf{x})$ and

$p_{\mathbf{x}}(\mathbf{x}) = p_{\mathbf{X}}(g(\mathbf{x}))$. Thus,

$$\begin{aligned} H(\mathbf{Y}) &= -\sum_{\mathcal{S}_{\mathbf{Y}}} p_{\mathbf{Y}}(\mathbf{y}) \log p_{\mathbf{Y}}(\mathbf{y}) \\ &= -\sum_{\mathcal{S}_{\mathbf{Y}}} \sum_{\mathbf{x}:g(\mathbf{x})=\mathbf{y}} p_{\mathbf{X}}(\mathbf{x}) \log p_{\mathbf{Y}}(\mathbf{y}) \\ &= -\sum_{\mathcal{S}_{\mathbf{X}}} p_{\mathbf{X}}(\mathbf{x}) \log p_{\mathbf{X}}(g(\mathbf{x})) \\ &= -\sum_{\mathcal{S}_{\mathbf{X}}} p_{\mathbf{X}}(\mathbf{x}) \log p_{\mathbf{X}}(\mathbf{x}) = H(\mathbf{X}). \end{aligned} \tag{2.37}$$

Definition 2.5.6. The differential (fine-grained) entropy of a continuous random variable $\mathbf{X}$ is defined as:

$$h(\mathbf{X}) = -\int_{\mathcal{S}_{\mathbf{X}}} p_{\mathbf{X}}(\mathbf{x}) \log p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}, \tag{2.38}$$

with $p_{\mathbf{X}}(\mathbf{x})$ is the probability density function (pdf) of $\mathbf{X}$ and $\mathcal{S}_X$ is the corresponding support set. It is assumed that $p_{\mathbf{X}}(\mathbf{x})$ and the integral in (2.38) exist. $0 \log(0) = 0$ by convention. Additionally, $p_{\mathbf{X}}(\mathbf{x})$ is bounded and differentiable, i.e., $(p'_{\mathbf{X}}(\mathbf{x}) \leq L, \quad L \in \mathbb{R}^+)$. Note that, the differential entropy can be negative unlike $H(\mathbf{X})$ in (2.29).

Definition 2.5.7. The joint differential entropy of two continuous random variables $(\mathbf{X}, \mathbf{Y})$ with joint density function $p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y})$ is defined as:

$$h(\mathbf{X}, \mathbf{Y}) = -\int_{\mathcal{S}_{\mathbf{X}}} \int_{\mathcal{S}_{\mathbf{Y}}} p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) \log p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} \tag{2.39}$$

with $p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y})$ is the joint probability density of $(\mathbf{x}, \mathbf{y})$.

Definition 2.5.8. If $(\mathbf{X}, \mathbf{Y})$ has a joint density function $p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y})$, the conditional entropy of $h(\mathbf{X} \mid \mathbf{Y})$ is defined as:

$$h(\mathbf{X} \mid \mathbf{Y}) = -\int_{\mathcal{S}_{\mathbf{X}}} \int_{\mathcal{S}_{\mathbf{Y}}} p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) \log p_{\mathbf{XY}}(\mathbf{x} \mid \mathbf{y}) d\mathbf{x} d\mathbf{y}. \tag{2.40}$$

Here, $p_{\mathbf{XY}}(\mathbf{x} \mid \mathbf{y})$ is the conditional pdf and $p_{\mathbf{XY}}(\mathbf{x} \mid \mathbf{y}) = p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y})/p_{\mathbf{Y}}(\mathbf{y})$.

The entropy of a continuous random variable can be approximated via the binning or "coarse-graining", i.e., by partitioning the support set.

Theorem 2.5.9. *Let $\mathbf{X} \in \mathbb{R}^d$ be a continuous random variable with the density $p_{\mathbf{X}}(\mathbf{x})$ which is Riemann integrable. Suppose that the support set is divided into bins whose Lebesgue measure Δ and $\mathbf{x}^{\Delta}$ is the discretization of $\mathbf{x}$. Then, $p_{\mathbf{X}}(\mathbf{x}^{\Delta}) \approx p_{\mathbf{X}}(\mathbf{x})\Delta^d$. $H_{bin}(\mathbf{X}^{\Delta})$ is defined by:*

$$H_{bin}(\mathbf{X}^{\Delta}) \approx h(\mathbf{X}) - d \log \Delta. \tag{2.41}$$

Clearly, when Δ decreases, (2.41) approximates the differential entropy in (2.38).

Theorem 2.5.10. *Unlike the course-grained entropy, the differential entropy is not preserved under the homeomorphisms in general (Bossomaier et al., 2016).*

Proof: Let $\mathbf{X}' = f(\mathbf{X})$ be a homeomorphism for $\mathbf{X}$. $J_{\mathbf{x}}(\mathbf{x}') = \det\left(\frac{\partial \mathbf{x}}{\partial \mathbf{x}'}\right)$ is the Jacobi determinant. Then, $p'_{\mathbf{X}'}(\mathbf{x}') = p_{\mathbf{X}}(\mathbf{x})J_{\mathbf{x}}(\mathbf{x}')$ and the differential entropy of $\mathbf{X}'$ becomes:

$$\begin{aligned} h(\mathbf{X}') &= -\int_{S_{\mathbf{X}'}} p'_{\mathbf{X}'}(\mathbf{x}') \log p'_{\mathbf{X}'}(\mathbf{x}') d\mathbf{x}' \\ &= -\int_{S_{\mathbf{X}'}} p_{\mathbf{X}}(\mathbf{x})J_{\mathbf{x}}(\mathbf{x}') \log p_{\mathbf{X}}(\mathbf{x})J_{\mathbf{x}}(\mathbf{x}') d\mathbf{x}' \\ &= -\int_{S_{\mathbf{X}}} p_{\mathbf{X}}(\mathbf{x}) \log p_{\mathbf{X}}(\mathbf{x})J_{\mathbf{X}}(\mathbf{X}') d\mathbf{x} \\ &\neq h(\mathbf{X}). \end{aligned} \tag{2.42}$$

The theorem highlights an important difficulty when using the concept of differential entropy in continuous time signals since the value of the differential entropy is not invariant under continuous coordinate transformations presented here as homeomorphisms. We must find a way to obtained course grained entropy from the observations since our problem involves networks with underlying continuous

time dynamical systems. Luckily, the state space reconstruction techniques provide the right tool set.

CHAPTER THREE
SYNCHRONIZATION OF DYNAMICAL SYSTEMS

In the sequel, we have reviewed the literature on the synchronization of dynamical systems and represented the necessary tools to design synchronous chaotic dynamical networks, which will be used as a test bench. The results on complete and cluster synchronization represented here will be useful in demonstrating the main results of the information-theoretic approach to identifying synchronous systems.

Synchronization is a phenomenon that occurs in networks of coupled chaotic dynamical systems frequently. The conditions under which this happens are of interest. The various types of synchronization have been studied including complete (Pecora & Carroll, 1990), cluster or partial (Hasler et al., 1998), lag (Rosenblum et al., 1997) and phase (Rosenblum et al., 1996) synchronizations in the literature. Loosely, if every member of a network of dynamical systems behaves identically, the synchronization is complete.

In contrast, the dynamical systems are synchronous if they are in a particular sub-set of the network and are out of synchrony with other groups of dynamical systems in cluster synchronization.

The pioneering studies deal with the complete synchronization of small networks and finding stable synchronization conditions when the network topology changes (Pecora & Carroll, 2015; Wu & Chua, 1995). The researchers have proposed the *master stability function* (MSF) method in these studies. It is based on calculating eigenvalues of the coupling matrix of the network and Lyapunov exponents of the internal dynamics of the systems. These calculations can be complicated, especially for the networks with time-varying coupling terms and random or tangled connections (Belykh et al., 2004).

Another alternative proposed study, the *graph-based stability* (GBS) method, has been put forward to compute upper bounds for global synchronization in coupled chaotic networks (Belykh et al., 2005). GBS method makes a connection between

graph theory and synchronization. The proposed study has succeeded in synchronizing regular and complex networks (Belykh et al., 2004). Synchronization conditions of networks of linear chains, two and three dimension lattice networks, and cluster synchronization of nonidentical systems have been derived using GBS during the last decades (Belykh et al., 2006, 2003; Lu et al., 2010).

Besides, synchronization conditions can be obtained by utilizing an external controller. The networks consist of a controller, and input is applied to the state equations of the system. In a recent study, the authors have combined synchronization and pinning controller and proposed that synchronization can occur by adding a single pinning controller on a single node (Chen et al., 2007). Afterward, Wu et al. derived the synchronization conditions for linearly coupled networks with pinning control (Wu et al., 2008). Cluster synchronization conditions for linearly coupled networks have also been investigated via intermittent pinning control (Liu & Chen, 2011). In that study, an external control input drives the network nodes at specific times. In (Wu & Fu, 2014), the cluster synchronization of the network with nonidentical systems has been investigated, and group synchronization has been explored in the study via pinning control (Deng, 2014).

The cluster synchronization of coupled networks has been investigated extensively (Belykh et al., 2000; Yanchuk et al., 2001; Pogromsky et al., 2002). Ma and Liu have also proposed a new method to obtain cluster synchronization. The main difference of this study is that the coupling scheme consists of the arbitrarily selected cluster synchronization patterns (Ma et al., 2006). Recently, the authors have studied the existence conditions and the stability of cluster synchronization of the networks (Pecora et al., 2014; Schaub et al., 2016; Nishikawa & Motter, 2016; Wu et al., 2008; Liu & Chen, 2011; Liu & Li, 2017; Gambuzza et al., 2020). The relation between network symmetry and cluster synchronization has been investigated. It has been shown that cluster synchronization mainly depends on the network structure more than the individual dynamics (Pecora et al., 2014; Sorrentino et al., 2016). In a recent study, Gambuzza et al. have designed a distributed controller inspired by biological networks and designed a distributed controller that allows us to cluster specifically

chosen systems in the network (Gambuzza et al., 2019). Therefore, the design of dynamical networks with requested synchronization properties is of great scientific interest, as well as the identification of the system structure of such networks.

3.1 Preliminaries

We need to define some helpful matrix properties to analyze the proposed approaches in the literature about the synchronization of dynamical systems.

Definition 3.1.1. The connection matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$ is defined as *diffusive coupling matrix* (*DCM*) if the following property is held:

$$\sum_{j=1}^{N} a_{ij} = 0 \quad i = 1, 2, \ldots N, \text{ and } a_{ij} \geq 0 \text{ for } i \neq j, a_{ij} < 0 \text{ for } i = j \tag{3.1}$$

In other words, $\mathbf{A}$ has non-negative off-diagonal elements and the property that the sum of each row is zero.

Diffusely coupled systems utilizing a diffusive coupling matrix have the important property that the energy exchanged between the coupled systems tends to zero as the systems synchronize asymptotically. Once the systems synchronize, they evolve without interaction if no disturbance is in effect.

Definition 3.1.2. Let $\mathbf{A}$ be a diffusive coupling matrix, which additionally satisfies the property of $a_{ij} = 1$ for $i \neq j$, then it is called a *diffusive global coupling matrix* (*DGCM*).

3.2 Design of Synchronous Networks

There are numerous approaches to designing dynamical networks that have desired synchronization properties. In this section, we briefly summarize the design methods of synchronization of chaotic systems used in this thesis.

3.2.1 Global synchronization with master stability function

The *synchronous state* or the *synchronization manifold* of N systems in (2.6) is expressed by

$$\mathbf{x}_1 = \mathbf{x}_2 = \ldots = \mathbf{x}_N \tag{3.2}$$

for the global synchronization. Suppose that we have a network in (2.6). The state equation of the network can be written in terms of the *Laplacian matrix*:

$$\begin{aligned} \dot{\mathbf{x}}_i &= f_i(\mathbf{x}_i) + \epsilon \sum_{j=1}^{N} a_{ij}\kappa(\mathbf{x}_j - \mathbf{x}_i) \\ &= f_i(\mathbf{x}_i) - \epsilon \sum_{j=1}^{N} \mathcal{L}_{ij}\kappa\mathbf{x}_j \end{aligned} \tag{3.3}$$

Note that if $\mathcal{L}$ matrix is DCM, the system in (3.3) yields an invariant synchronization manifold, i.e., when a flow starts on this manifold, it remains there forever. Also, when the system in (3.3) completely synchronizes according to (2.22), the individual systems follow the same trajectory and the coupling terms vanish. MSF determines the network's stability and obtains the network's synchronization conditions given a coupling matrix $\mathbf{A}$ (Frasca et al., 2018). According to Pecora and Carroll, the assumptions for the MSF approach are (Pecora & Carroll, 1998):

i) All systems in the network are identical.

ii) All systems have the same coupling function.

iii) The synchronization manifold is an invariant manifold.

iv) All systems are approximated near the synchronous state by a linear operator.

MSF approach considers two crucial steps to derive the stability of the

Figure 3.1 The complete synchronization network example. Each circle represents a chaotic Chua system

synchronous state. As a first step, the master stability function (i.e., consists of the linearization around synchronous state and calculation of Lyapunov exponents of the system.) is computed considering individual dynamics and coupling function. Then, MSF is applied at critical points that are obtained by the eigenvalues of $\mathcal{L}$ and the coupling strength (Pecora & Carroll, 1998; Belykh et al., 2005; Frasca et al., 2018).

Consider a network that consists of $N = 10$ Chua systems in Figure 3.1. The equations of Chua system is given (2.13) and the parameters of the Chua system has been chosen as $m_0 = -1.27$, $m_1 = -0.68$, $\alpha = 10$, $\beta = 15$, $\gamma = 0.0385$. It satisfies all conditions of being a chaotic system for these parameters. The inner coupling matrix has been selected as $\boldsymbol{\kappa} = [[1, 0, 0], [0, 0, 0], [0, 0, 0]]$. For the network in Figure 3.1 be in complete synchronization, it has been found that the coupling strength must satisfy $\epsilon \geq 105$ using MSF (Frasca et al., 2018). We have illustrated the global synchronization error defined in (2.27) for $\epsilon = 1.27$ and $\epsilon = 105$, respectively in Figure 3.2.

3.2.2 A method to realize cluster synchronization

The authors in (Ma et al., 2006) have proposed a new method that ensures arbitrarily selected patterns in cluster synchronization by selecting an appropriate coupling scheme.

Consider a network in (2.6) that has N interacting identical systems. Let $\mathbf{A} = [a_{ij}]$ in (2.1) be a symmetric matrix in this network. Suppose that individual members of the network in (2.6) are in cluster synchronization according to definition 2.4.3. The

(a) Global synchronization error for $\epsilon = 1.27$

(b) Global synchronization error for $\epsilon = 105$. The inset plot indicates global synchronization error between 0 and 1 seconds

Figure 3.2 The global synchronization error in Figure 3.1 for various ϵ's

coupling connection matrix **A** must satisfy the following definition:

Definition 3.2.1. Let $\mathbf{A} \in \mathbb{R}^{N \times N}$ be a real matrix and has the form:

Figure 3.3 The cluster synchronization network. While systems 1 and 2 are in the same cluster, systems 3 and 4 form a different cluster. Each circle represents a chaotic Lorenz system.

$$\mathbf{A} = [a_{ij}] = \begin{bmatrix} 3\mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{12} & 5\mathbf{A}_{22} & \mathbf{A}_{23} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{23} & 5\mathbf{A}_{23} & \ddots & \dots & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \ddots & \ddots & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & 5\mathbf{A}_{m-1,m-1} & \mathbf{A}_{m-1,m} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_{m,m-1} & 3\mathbf{A}_{m,m} \end{bmatrix}. \tag{3.4}$$

Here, $\mathbf{A}_{i,i} \in \mathbb{R}^{m_i \times m_i}$, and $\mathbf{A}_{i,i} \in DGCM$, $\mathbf{A}_{i,i+1} \in \mathbb{R}^{m_i \times m_i+1}$, $\mathbf{A}^T_{i+1,i} = \mathbf{A}_{i,i+1} = (\mathbf{A}_{i,i}, 0)$ if $m_i < m_{i+1}$ and $\mathbf{A}^T_{i+1,i} = \mathbf{A}_{i,i+1} = \mathbf{A}_{i,i}$ if $m_i = m_{i+1}$ for $i = 1, 2, \dots, m$. Then, the following conditions hold for the eigenvalues of $\mathbf{A}$.

i) $\lambda_i = 0$, $i = 1, \dots, m$ with corresponding m eigenvectors $\xi_i = (\upsilon_{1i}, \dots, \upsilon_{mi})^T$ where $\upsilon_{ki} \in \mathbb{R}^{m_\ell}$, $\upsilon_{1i} = (1, 1, \dots, 1)$, if $\ell = i$. Otherwise, $\upsilon_{1i} = (0, 0, \dots, 0)$.

ii) the other eigenvalues, i.e., $\lambda_j < 0$, $j = m + 1, \dots, N$.

The details of the study have been summarized in (Ma et al., 2006). It is shown that the stability of the cluster synchronization manifold is derived by choosing a coupling scheme in (3.4). In light of this information, we can design a cluster synchronization network.

Let a network be defined with identical chaotic Lorenz systems as Figure 3.3 depicts. The network has four systems, and each circle represents a Lorenz system. The systems in the network are coupled through x-state variables with the coupling strength a_{ij} in Figure 3.3. The state equations are given by (2.8). Here, $\sigma = 10.0$, $b = 8/3$, $r = 28.0$ is chosen. Individual Lorenz systems show chaotic behavior for these parameters (Lorenz, 1963). The connection matrix $\mathbf{A} = [a_{ij}]$ is chosen by using the rules of (Ma et al., 2006):

$$\mathbf{A} = [a_{ij}] = \begin{bmatrix} -3 & 3 & -1 & 1 \\ 3 & -3 & 1 & -1 \\ -1 & 1 & -3 & 3 \\ 1 & -1 & 3 & -3 \end{bmatrix} \tag{3.5}$$

To ensure the cluster synchronization in Definition (2.4.3), the coupling strength ϵ should be reach a threshold value. According to Ma et al. design, this network has two clusters depending on ϵ. Hence, we choose the coupling strength as $\epsilon = 10$ to ensure cluster synchronization. The error in (2.26) for all systems in Figure 3.3 is illustrated in Figure 3.4. We can see that systems 1 and 2 are in the same cluster, but 3 and 4 form a different cluster. Thus, we can form clusters of chaotic dynamical systems by Ma et al.' method (Ma et al., 2006).

3.2.3 Cluster synchronization network via the distributed controller

Gambuzza et al. have proposed the distributed controller-based approach that the cluster synchronization only occurs in arbitrary sets while the remaining ones are not synchronized. The details of the study are given in (Gambuzza et al., 2019). Suppose that we have a network that consists N identical systems with d-dimensional oscillators defined by (2.6). The network equation can be defined in the Laplacian matrix in (2.3). Let n_2 be the number of nodes we want to synchronize and $n_1 = N - n_2$ be the remaining nodes in the network. $\mathcal{S}$ is denoted as a subset of nodes desired to synchronize. If

Figure 3.4 $e_{i,j}(t)$, i, j = 1, 2, 3, 4 for the systems in the cluster synchronization network

distributed controller term is added to the systems, (3.3) becomes,

$$\dot{\mathbf{x}}_i = f(\mathbf{x}_i) - \epsilon \sum_{i=1}^{N} \mathcal{L}_{ij} \kappa \mathbf{x}_j + \mathbf{u}_i \tag{3.6}$$

where $\mathbf{u}_i = \sum_{j=1}^{N} \mathcal{L}'_{ij} \kappa \mathbf{x}_j$ are the control inputs. $\mathcal{L}'_{ij}$ symbolizes the new link between i and j that is added or removed from the network. If a link is inserted, $\mathcal{L}'_{ij}$ will be equal to -1, $\mathcal{L}'_{ij} = 1$ in case of the addition of a link, and $\mathcal{L}_{ij} = 0$ otherwise. The new Laplacian ($\mathcal{L}" = \mathcal{L} + \mathcal{L}'$) matrix must satisfy the following conditions (Gambuzza et al., 2019):

i) The nodes in $\mathcal{S}$ are not directly connected. However, the neighborhood of each node in $\mathcal{S}$ must be the same.

ii) The coupling strength of the network is fixed, and it is directly related to the minimum number of links provided by the controller.

The proposed method has been tested on a network of Chua systems given by (2.13). The Chua systems' parameters have been selected the same as in the previous section 3.2.1.

Consider we have $N = 20$ identical Chua systems in the network as shown in Figure 3.5. Let us label the systems as $\{a, b, \ldots t, u\}$, randomly. The inner coupling matrix has been chosen as $\boldsymbol{\kappa} = [[1, 0, 0], [0, 1, 0], [0, 0, 0]]$ and the coupling coefficient has been selected as $\epsilon = 2$. If we desire to synchronize the systems labeled k and d, so our cluster members are $\mathcal{S} = \{k, d\}$. To achieve synchronization, the neighbors of k and d must be the same. While k^{th} node is connected to f^{th} and o^{th} nodes, d^{th} node is only neighbor with j. If we add the links $\{(k, j), (d, f), (d, o)\}$, the neighbors of the nodes k and d become same as $\{f, o, j\}$. Thus, it allows us to synchronize the systems d and k while the remaining ones are not synchronized. The global synchronization error in (2.27) $E(t)$ for the network and the error in (2.26) for d and k system $e_{d,k}(t)$ have been plotted without control and with control (i.e. adding necessary connections) in Figures 3.6 and 3.7, respectively.

Figure 3.5 The cluster synchronization network via the distributed controller. Each node represents a Chua system

(a) Error between d and k systems

(b) Global synchronization error

Figure 3.6 Synchronization error between d and k system and global synchronization error without distributed control

The distributed controller provides cluster synchronization in selected nodes in the network, therefore, can be an alternative to designing clusters in dynamic networks.

(a) Error between d and k systems

(b) Global synchronization error

Figure 3.7 Synchronization error between d and k system and global synchronization error with distributed control

Undoubtedly, the dynamical systems other than chaotic dynamical systems are able to synchronize. However, the information they generate is predictable as in synchronization of systems of periodic oscillations which yields a little or no new information observed after a certain amount of duration. We examine the chaotic dynamical networks because their information generation features. Since the chaotic systems can exhibit unpredictable and complex behavior sensitivity to initial conditions, observations at current time, has a limited future prediction capacity. Every time we make a measurement we resolve new information about the future state of the systems. On the other hand, we are sure that observing one system in a network of synchronized systems will not provide additional information either. Hence, reconstruction of the network structure from limited observations can be challenging.

We use the designed chaotic networks in this section as test beds for analysis with average integrated causation entropy which is introduced in the next section. We assume that the underlying network is unknown except the outputs (observations) obtained from the designed systems. It is then possible to compare the obtained connectivity information by the original designs.

CHAPTER FOUR
INFORMATION THEORY

The basic definitions of information-theoretic tools such as mutual information, transfer entropy, and causation entropy have been represented in this chapter. Also, the estimation techniques of the information-theoretic measures in the literature have been explained in detail.

4.1 Information-Theoretic Tools

Mutual information (MI) between two random variables is defined as a reduction in uncertainty of one of the related random variables due to measurement of the other random variable (Cover & Thomas, 2012). MI quantifies the amount of information about one random variable by observing at least one other related random variable.

Definition 4.1.1. The MI, $I(\mathbf{X};\mathbf{Y})$, between two discrete random variables is defined as (Cover & Thomas, 2012):

$$\begin{aligned} I(\mathbf{X};\mathbf{Y}) &= \sum_{\mathcal{S}_X}\sum_{\mathcal{S}_Y} p_{xy}(x,y)\log\frac{p_{xy}(x,y)}{p_x(x)p_y(y)} \\ &= H(\mathbf{X}) + H(\mathbf{Y}) - H(\mathbf{X},\mathbf{Y}). \end{aligned} \tag{4.1}$$

$\mathcal{S}_\mathbf{X}$ and $\mathcal{S}_\mathbf{Y}$ are the support sets of $\mathbf{X}$ and $\mathbf{Y}$, respectively.

Definition 4.1.2. The MI, $I(\mathbf{X};\mathbf{Y})$, between two continuous random variables $\mathbf{X}$ and $\mathbf{Y}$ is defined as (Cover & Thomas, 2012):

$$\begin{aligned} I(\mathbf{X};\mathbf{Y}) &= \int_{\mathcal{S}_X}\int_{\mathcal{S}_Y} p_{xy}(\mathbf{x},\mathbf{y})\log\frac{p_{xy}(\mathbf{x},\mathbf{y})}{p_x(\mathbf{x})p_y(\mathbf{y})}d\mathbf{x}d\mathbf{y} \\ &= h(\mathbf{X}) + h(\mathbf{Y}) - h(\mathbf{X},\mathbf{Y}). \end{aligned} \tag{4.2}$$

The following properties can be shown to hold for MI using the definition directly:

i) $I(\mathbf{X};\mathbf{Y}) = I(\mathbf{Y};\mathbf{X})$ (Symmetry) (4.3)

ii) $0 \leq I(\mathbf{X};\mathbf{Y})$ (Nonnegativity) (4.4)

iii) $I(\mathbf{X};\mathbf{X}) = h(\mathbf{X})$ (4.5)

iv) $I(\mathbf{X};\mathbf{Y}) \leq min[h(\mathbf{X}), h(\mathbf{Y})]$ (4.6)

v) $I(X_1, X_2, \ldots, X_n; Y) = \sum_{i=n}^{n} I(X_i; Y \mid X_1, X_2, \ldots, X_{i-1})$ (Chain rule) (4.7)

Definition 4.1.3. The conditional MI of discrete random variables $\mathbf{X}, \mathbf{Y}$, given $\mathbf{Z}$ is given as (Frenzel & Pompe, 2007):

$$\begin{aligned} I(\mathbf{X};\mathbf{Y} \mid \mathbf{Z}) &= \sum_{\mathcal{S}_\mathbf{X}} \sum_{\mathcal{S}_\mathbf{Y}} \sum_{\mathcal{S}_\mathbf{Z}} p_{\mathbf{XYZ}}(\mathbf{x},\mathbf{y},\mathbf{z}) \log \frac{p_{\mathbf{XYZ}}(\mathbf{x},\mathbf{y},\mathbf{z}) p_\mathbf{Z}(\mathbf{z})}{p_{\mathbf{XZ}}(\mathbf{x},\mathbf{z}) p_{\mathbf{YZ}}(\mathbf{y},\mathbf{z})} \\ &= H(\mathbf{X},\mathbf{Z}) + H(\mathbf{Y},\mathbf{Z}) - H(\mathbf{Z}) - H(\mathbf{X},\mathbf{Y},\mathbf{Z}). \end{aligned} \tag{4.8}$$

Similarly, if $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ are continuous random variables with joint density $p_{\mathbf{XYZ}}(\mathbf{x},\mathbf{y},\mathbf{z})$, $I(\mathbf{X};\mathbf{Y} \mid \mathbf{Z})$ becomes,

$$\begin{aligned} I(\mathbf{X};\mathbf{Y} \mid \mathbf{Z}) &= \int_{\mathcal{S}_\mathbf{X}} \int_{\mathcal{S}_\mathbf{Y}} \int_{\mathcal{S}_\mathbf{Z}} p_{\mathbf{XYZ}}(\mathbf{x},\mathbf{y},\mathbf{z}) \log \frac{p_{\mathbf{XYZ}}(\mathbf{x},\mathbf{y},\mathbf{z}) p_\mathbf{Z}(\mathbf{z})}{p_{\mathbf{XZ}}(\mathbf{x},\mathbf{z}) p_{\mathbf{YZ}}(\mathbf{y},\mathbf{z})} d\mathbf{x} d\mathbf{y} d\mathbf{z} \\ &= h(\mathbf{X},\mathbf{Z}) + h(\mathbf{Y},\mathbf{Z}) - h(\mathbf{Z}) - h(\mathbf{X},\mathbf{Y},\mathbf{Z}). \end{aligned} \tag{4.9}$$

$I(\mathbf{X};\mathbf{Y} \mid \mathbf{Z}) \geq 0$, and the equality holds if and only if $\mathbf{X}$ and $\mathbf{Y}$ are independent given $\mathbf{Z}$.

If there exists a homeomorphism between the original state space and reconstructed state space, and then we estimate the information-theory measures in the reconstructed state space, the measures are preserved in the estimation of the measures using the state space. Therefore, following theorem is important.

Theorem 4.1.4. *Unlike* $h(\mathbf{X})$ *in* (2.38)*, fine-grained* $I(\mathbf{X};\mathbf{Y})$ *is preserved under homeomorphism (Kraskov et al., 2004).*

Proof: Let $\mathbf{X}' = f(\mathbf{X})$ and $\mathbf{Y}' = g(\mathbf{Y})$ be homeomorphisms for $\mathbf{X}$ and $\mathbf{Y}$, respectively. The Jacobi determinants of $\mathbf{X}$ and $\mathbf{Y}$ are defined as $J_{\mathbf{x}}(\mathbf{x}') = \det\left(\frac{\partial \mathbf{x}}{\partial \mathbf{x}'}\right)$ and $J_{\mathbf{y}}(\mathbf{y}') = \det\left(\frac{\partial \mathbf{y}}{\partial \mathbf{y}'}\right)$. The joint density function of $(\mathbf{X}', \mathbf{Y}')$ becomes:

$$p'_{\mathbf{X}'\mathbf{Y}'}(\mathbf{x}', \mathbf{y}') = p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) J_{\mathbf{x}}(\mathbf{x}') J_{\mathbf{y}}(\mathbf{y}'). \tag{4.10}$$

If $I(\mathbf{X}'; \mathbf{Y}')$ is evaluated using (4.2) we have:

$$\begin{aligned} I(\mathbf{X}'; \mathbf{Y}') &= \int_{\mathcal{S}_{\mathbf{x}'}} \int_{\mathcal{S}_{\mathbf{y}'}} p'_{\mathbf{X}'\mathbf{Y}'}(\mathbf{x}', \mathbf{y}') \log \frac{p'_{\mathbf{X}'\mathbf{Y}'}(\mathbf{x}', \mathbf{y}')}{p'_{\mathbf{X}'}(\mathbf{x}') p'_{\mathbf{Y}'}(\mathbf{y}')} d\mathbf{x}' d\mathbf{y}' \\ &= \int_{\mathcal{S}_{\mathbf{X}'}} \int_{\mathcal{S}_{\mathbf{Y}'}} p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) J_{\mathbf{x}}(\mathbf{x}') J_{\mathbf{y}}(\mathbf{y}') \log \frac{p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) J_{\mathbf{x}}(\mathbf{x}') J_{\mathbf{y}}(\mathbf{y}')}{p_{\mathbf{X}}(\mathbf{x}) J_{\mathbf{x}}(\mathbf{x}') p_{\mathbf{Y}}(\mathbf{y}) J_{\mathbf{y}}(\mathbf{y}')} d\mathbf{x}' d\mathbf{y}' \\ &= \int_{\mathcal{S}_{\mathbf{X}}} \int_{\mathcal{S}_{\mathbf{Y}}} p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) \log \frac{p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y})}{p_{\mathbf{X}}(\mathbf{x}) p_{\mathbf{Y}}(\mathbf{y})} d\mathbf{x} d\mathbf{y} \\ &= I(\mathbf{X}; \mathbf{Y}). \end{aligned} \tag{4.11}$$

When the density function is known, the information-theoretic measures such as entropy and MI of the systems are calculated from the definitions directly. The differential entropy and MI of well-known distributions are given below:

i) Let X be a uniform distribution in the real interval $[a, b]$. Then, $h(X) = \log(b - a)$.

ii) If X is a Gaussian random variable with zero mean and variance σ, the probability distribution function is $p_x(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/(2\sigma^2)}$. Then, $h(X)$ of the normal distribution is given by:

$$h(X) = \frac{1}{2} \log(2\pi e \sigma^2) \quad nats. \tag{4.12}$$

iii) If $\mathbf{X} = [X_1, X_2, \ldots, X_d]$ has a multivariate normal distribution whose mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{K}$, $h(\mathbf{X})$ is defined as:

$$h(\mathbf{X}) = h(X_1, X_2, \ldots, X_d) = h(\mathcal{N}_d(\boldsymbol{\mu}, \mathbf{K})) = \frac{1}{2}\log((2\pi e)^d det(\mathbf{K})) \quad nats. \quad (4.13)$$

Here, $det[\cdot]$ denotes the determinant of a given matrix.

iv) Let two Gaussian random variables (X, Y) have distribution with zero mean and covariance matrix $\mathbf{K}$,

$$\mathbf{K} = \begin{bmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{bmatrix}. \quad (4.14)$$

where the correlation coefficient is given as ρ. Then,

$$I(X;Y) = -\frac{1}{2}\log(1-\rho^2). \quad (4.15)$$

When $\rho = 0$, X and Y are independent and $I(X;Y) = 0$. When $\rho = \pm 1$, X and Y are strongly correlated and $I(X;Y)$ is infinite.

Due to the symmetry of MI, i.e., $I(\mathbf{X};\mathbf{Y}) = I(\mathbf{Y};\mathbf{X})$, MI does not indicate which way the information is flowing between the variables. Therefore, MI cannot measure the information flow between dynamic systems.

As a promising alternative, *transfer entropy* (TE) has been proposed to measure the direction of the information flow by adding transition (dynamic) probabilities rather than static probabilities. (Schreiber, 2001). A recent study has used the TE to measure information flow between industrial processes and detect their causality (Duan et al., 2013). Moreover, Wibral et al. have discussed the fundamental concepts of information theory and presented a summary of the applications of TE in neuroscience (Wibral et al., 2014). The authors have argued to characterize information flows in neuroscience and to quantify neural spike trains through information measures in (Spinney et al., 2017).

Let us denote $\{x_n\}_{k=1}^{N}$ and $\{y_n\}_{k=1}^{N}$ as a set of measurements, perhaps with nonuniform sampling of outputs, taken from two related systems. Here, $\mathbf{x}_n{}^{(k)} = (x_n, x_{n-1}, \ldots, x_{n-k+1})$ denotes as measurements at time steps from n to $n-k+1$ and $\mathbf{y}_n{}^{(\ell)} = (y_n, y_{n-1}, \ldots, y_{n-\ell+1})$ denotes as measurements at time steps from n to $n-\ell+1$. x_{n+1} is also a measurement taken at time step $n+1$. By rearranging the variables from $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ to $(X_{n+1}, \mathbf{Y}_n^{(\ell)}, \mathbf{X}_n^{(k)})$ in Equation (4.9), respectively, then TE at time n is defined as:

$$\begin{aligned} TE_{\mathbf{Y}\to\mathbf{X}} &= I(X_{n+1}, \mathbf{Y}_n^{(\ell)} \mid \mathbf{X}_n^{(k)}) \\ &= H(X_{n+1} \mid \mathbf{X}_n^{(k)}) - H(X_{n+1} \mid \mathbf{X}_n^{(k)}, \mathbf{Y}_n^{(\ell)}) \\ &= [H(X_{n+1}, \mathbf{X}_n^{(k)}) - H(\mathbf{X}_n^{(k)})] - [H(X_{n+1}, \mathbf{X}_n^{(k)}, \mathbf{Y}_n^{(\ell)}) - H(\mathbf{X}_n^{(k)}, \mathbf{Y}_n^{(\ell)})] \\ &= \sum_{x_{n+1}} \sum_{\mathbf{x}_n^{(k)}} \sum_{\mathbf{y}_n^{(\ell)}} p(x_{n+1}, \mathbf{x}_n^{(k)}, \mathbf{y}_n^{(\ell)}) \log \frac{p(x_{n+1} \mid \mathbf{x}_n^{(k)}, \mathbf{y}_n^{(\ell)})}{p(x_{n+1} \mid \mathbf{x}_n^{(k)})}. \end{aligned} \tag{4.16}$$

For the continuous random variables, TE becomes :

$$\begin{aligned} TE_{\mathbf{Y}\to\mathbf{X}} &= I(X_{n+1}, \mathbf{Y}_n^{(\ell)} \mid \mathbf{X}_n^{(k)}) \\ &= h(X_{n+1} \mid \mathbf{X}_n^{(k)}) - h(X_{n+1} \mid \mathbf{X}_n^{(k)}, \mathbf{Y}_n^{(\ell)}) \\ &= \left[h(X_{n+1}, \mathbf{X}_n^{(k)}) + h(\mathbf{X}_n^{(k)}, \mathbf{Y}_n^{(\ell)}) - h(\mathbf{X}_n^{(k)}) - h(X_{n+1}, \mathbf{X}_n^{(k)}, \mathbf{Y}_n^{(\ell)})\right] \\ &= \int_{\mathcal{S}} p(x_{n+1}, \mathbf{x}_n^{(k)}, \mathbf{y}_n^{(\ell)}) \log \frac{p(x_{n+1} \mid \mathbf{x}_n^{(k)}, \mathbf{y}_n^{(\ell)})}{p(x_{n+1} \mid \mathbf{x}_n^{(k)})} dx_{n+1} d\mathbf{x}_n^{(k)} d\mathbf{y}_n^{(\ell)}. \end{aligned} \tag{4.17}$$

The conditional probability densities $p(x_{n+1} \mid \mathbf{x}_n^{(k)}, \mathbf{y}_n^{(\ell)})$ and $p(x_{n+1} \mid \mathbf{x}_n^{(k)})$, and the integral exist (4.17) because we assumed the solution of the underlying differential equation system exists. TE is non-negative and is equal to zero if and only if $p(x_{n+1} \mid \mathbf{x}_n^{(k)}) = p(x_{n+1} \mid \mathbf{x}_n^{(k)}, \mathbf{y}_n^{(\ell)})$ holds. As opposed to MI, TE is not necessarily a symmetric measure, i.e., $TE_{\mathbf{X}\to\mathbf{Y}} \neq TE_{\mathbf{Y}\to\mathbf{X}}$, and it *may* indicate which way the information is flowing.

Theorem 4.1.5. *TE is preserved under homeomorphisms.*

Proof: Let $\mathbf{X}_n^{'(k)} = f(\mathbf{X}_n^{(k)})$, $\mathbf{Y}_n^{'(\ell)} = g(\mathbf{Y}_n^{(\ell)})$ and $X_{n+1}^{'} = h(X_{n+1})$ be homeomorphisms for $\mathbf{X}_n^{(k)}$, $\mathbf{Y}_n^{(\ell)}$ and X_{n+1}, respectively. The Jacobi determinants of $\mathbf{X}_n^{(k)}$, $\mathbf{Y}_n^{(\ell)}$ and X_{n+1} are defined as $J_{\mathbf{x}_n}(\mathbf{x}_n^{'}) = \det\left(\frac{\partial \mathbf{x}_n}{\partial \mathbf{x}_n^{'}}\right)$, $J_{\mathbf{y}_n}(\mathbf{y}_n^{'}) = \det\left(\frac{\partial \mathbf{y}_n}{\partial \mathbf{y}_n^{'}}\right)$ and $J_{x_{n+1}}(x_{n+1}^{'}) = \det\left(\frac{\partial x_{n+1}}{\partial x_{n+1}^{'}}\right)$. The joint density function of $(X_{n+1}, \mathbf{X}_n^{'(k)}, \mathbf{Y}_n^{'(\ell)})$ becomes:

$$p'(x_{n+1}', \mathbf{x}_n^{'(k)}, \mathbf{y}_n^{'(\ell)}) = p(x_{n+1}, \mathbf{x}_n^{(k)}, \mathbf{y}_n^{(\ell)}) J_{x_n}(x_n^{'}) J_{y_n}(y_n^{'}) J_{x_{n+1}}(x_{n+1}^{'}) \tag{4.18}$$

$$\begin{aligned}
TE_{Y'\to X'} &= \int_{\mathcal{S}} p'(x_{n+1}', \mathbf{x}_n^{'(k)}, \mathbf{y}_n^{'(\ell)}) \log \frac{p'(x_{n+1}' \mid \mathbf{x}_n^{'(k)}, \mathbf{y}_n^{'(\ell)})}{p'(x_{n+1}' \mid \mathbf{x}_n^{'(k)})} dx_{n+1}' d\mathbf{x}_n^{'(k)} d\mathbf{y}_n^{'(\ell)} \\
&= \int_{\mathcal{S}} p'(x_{n+1}', \mathbf{x}_n^{'(k)}, \mathbf{y}_n^{'(\ell)}) \cdots \\
&\quad \times \log \frac{p'(x_{n+1}', \mathbf{x}_n^{'(k)}, \mathbf{y}_n^{'(\ell)}) p'(\mathbf{x}_n^{'(k)})}{p'(\mathbf{x}_n^{'(k)}, \mathbf{y}_n^{'(\ell)}) p'(x_{n+1}', \mathbf{x}_n^{'(k)})} dx_{n+1}' d\mathbf{x}_n^{'(k)} d\mathbf{y}_n^{'(\ell)} \\
&= \int_{\mathcal{S}} p(x_{n+1}, \mathbf{x}_n^{(k)}, \mathbf{y}_n^{(\ell)}) J_{\mathbf{x}_n}(\mathbf{x}_n^{'}) J_{\mathbf{y}_n}(\mathbf{y}_n^{'}) J_{x_{n+1}}(x_{n+1}^{'}) \cdots \\
&\quad \times \log \frac{p(x_{n+1}, \mathbf{x}_n^{(k)}, \mathbf{y}_n^{(\ell)}) p(\mathbf{x}_n^{(k)})}{p(\mathbf{x}_n^{(k)}, \mathbf{y}_n^{(\ell)}) p(x_{n+1}, \mathbf{x}_n^{(k)})} dx_{n+1}' d\mathbf{x}_n^{'(k)} d\mathbf{y}_n^{'(\ell)} \\
&= \int_{\mathcal{S}} p(x_{n+1}, \mathbf{x}_n^{(k)}, \mathbf{y}_n^{(\ell)}) \log \frac{p(x_{n+1} \mid \mathbf{x}_n^{(k)}, \mathbf{y}_n^{(\ell)})}{p(x_{n+1} \mid \mathbf{x}_n^{(k)})} dx_{n+1} d\mathbf{x}_n^{(k)} d\mathbf{y}_n^{(\ell)} \\
&= TE_{\mathbf{Y}\to\mathbf{X}}.
\end{aligned} \tag{4.19}$$

Although the TE detects the information flow between two systems, it can be misleading when measuring the information flow of indirectly coupled systems, even in a small network, to infer causality (Sun & Bollt, 2014). The *causation entropy* (CE) measure has been proposed by Sun and Bollt to overcome this difficulty. It generalizes the TE and measures information flow from one system to another by conditioning on a third system, which is generally chosen related to these systems. The CE indicates if a third system directly affects the information flow between the two systems.

Definition 4.1.6. CE from X to Y conditioned on a third variable given Z is defined as:

$$CE_{X \to Y \mid Z} = H(Y_{n+1} \mid \mathbf{Z}_n^{(m)}) - H(Y_{n+1} \mid \mathbf{Z}_n^{(m)}, \mathbf{X}_n^{(k)}). \tag{4.20}$$

The CE from X to Y conditioned on Y and Z is defined as

$$CE_{X \to Y \mid (Y,\, Z)} = H(Y_{n+1} \mid \mathbf{Y}_n^{(\ell)}, \mathbf{Z}_n^{(m)}) - H(Y_{n+1} \mid \mathbf{X}_n^{(k)}, \mathbf{Y}_n^{(\ell)}, \mathbf{Z}_n^{(m)}). \tag{4.21}$$

If $k = \ell = m = 1$, then we have,

$$CE_{X \to Y \mid (Y,\, Z)} = H(Y_{n+1} \mid Y_n, Z_n) - H(Y_{n+1} \mid X_n, Y_n, Z_n). \tag{4.22}$$

The CE for continuous-time random variables is defined by:

$$\begin{aligned} CE_{X \to Y \mid Z} &= h(Y_{n+1} \mid \mathbf{Z}_n^{(m)}) - h(Y_{n+1} \mid \mathbf{Z}_n^{(m)}, \mathbf{X}_n^{(k)}) \\ &= [h(Y_{n+1}, \mathbf{Z}_n^{(m)}) - h(\mathbf{Z}_n^{(m)}) - h(Y_{n+1}, \mathbf{Z}_n^{(m)}, \mathbf{X}_n^{(k)}) + h(\mathbf{Z}_n^{(m)}, \mathbf{X}_n^{(k)})] \\ &= \int_{\mathcal{S}} p(y_{n+1}, \mathbf{x}_n^{(k)}, \mathbf{z}_n^{(m)}) \log \frac{p(y_{n+1} \mid \mathbf{x}_n^{(k)}, \mathbf{z}_n^{(m)})}{p(y_{n+1} \mid \mathbf{z}_n^{(m)})} dy_{n+1} d\mathbf{x}_n^{(k)} d\mathbf{z}_n^{(m)} \end{aligned} \tag{4.23}$$

The following properties hold for CE:

i) $CE_{X \to Y \mid Y} = TE_{X \to Y}$

ii) $CE_{X \to Y \mid \emptyset} = I(Y_{n+1}, X_n)$

Theorem 4.1.7. *CE is preserved under homeomorphisms.*

Proof: Let $\mathbf{Y}_n^{'(\ell)} = f(\mathbf{Y}_n^{(\ell)})$, $\mathbf{Z}_n^{'(m)} = g(\mathbf{Z}_n^{(m)})$ and $X_{n+1}^{'} = h(X_{n+1})$ be homeomorphisms for $\mathbf{Y}_n^{(\ell)}$, $\mathbf{Z}_n^{(m)}$ and X_{n+1}, respectively. The Jacobi determinants of $\mathbf{Y}_n^{(\ell)}$, $\mathbf{Z}_n^{(m)}$ and X_{n+1} are defined as $J_{\mathbf{y}_n}(\mathbf{y}_n^{'}) = \det\left(\frac{\partial \mathbf{y}_n}{\partial \mathbf{y}_n^{'}}\right)$, $J_{\mathbf{z}_n}(\mathbf{z}_n^{'}) = \det\left(\frac{\partial \mathbf{z}_n}{\partial \mathbf{z}_n^{'}}\right)$ and $J_{x_{n+1}}(x_{n+1}^{'}) = \det\left(\frac{\partial x_{n+1}}{\partial x_{n+1}^{'}}\right)$. The

joint density function of $(X_{n+1}, \mathbf{Y}_n^{'(\ell)}, \mathbf{Z}_n^{'(m)})$ is:

$$p'(x'_{n+1}, \mathbf{y}_n^{'(\ell)}, \mathbf{z}_n^{'(m)}) = p(x_{n+1}, \mathbf{y}_n^{(\ell)}, \mathbf{z}_n^{(m)}) J_{\mathbf{y}_n}(\mathbf{y}_n^{'}) J_{\mathbf{z}_n}(\mathbf{z}_n^{'}) J_{x_{n+1}}(x_{n+1}^{'}) \tag{4.24}$$

CE interm of transformed variables is then

$$\begin{aligned}
CE_{Y' \to X' \mid Z'} &= h(X'_{n+1} \mid Z'_n) - h(X'_{n+1} \mid Z'_n, Y'_n) \\
&= \int_{\mathcal{S}} p'(x'_{n+1}, \mathbf{y}_n^{'(\ell)}, \mathbf{z}_n^{'(m)}) \log \frac{p'(x'_{n+1} \mid \mathbf{y}_n^{'(\ell)}, \mathbf{z}_n^{'(m)})}{p'(x'_{n+1} \mid \mathbf{z}_n^{'(m)})} dx'_{n+1} d\mathbf{y}_n^{'(\ell)} d\mathbf{z}_n^{'(m)} \\
&= \int_{\mathcal{S}} p'(x'_{n+1}, \mathbf{y}_n^{'(\ell)}, \mathbf{z}_n^{'(m)}) \cdots \\
&\qquad \times \log \frac{p'(x'_{n+1}, \mathbf{y}_n^{'(\ell)}, \mathbf{z}_n^{'(m)}) p'(\mathbf{z}_n^{'(m)})}{p'(\mathbf{y}_n^{'(\ell)}, \mathbf{z}_n^{'(m)}) p'(x'_{n+1}, \mathbf{z}_n^{'(m)})} dx'_{n+1} d\mathbf{y}_n^{'(\ell)} d\mathbf{z}_n^{'(m)} \\
&= \int_{\mathcal{S}} p(x_{n+1}, \mathbf{y}_n^{(\ell)}, \mathbf{z}_n^{(m)}) J_{\mathbf{y}_n}(\mathbf{y}_n^{'}) J_{\mathbf{z}_n}(\mathbf{z}_n^{'}) J_{x_{n+1}}(x_{n+1}^{'}) \cdots \\
&\qquad \times \log \frac{p(x_{n+1}, \mathbf{y}_n^{(\ell)}, \mathbf{z}_n^{(m)}) p(\mathbf{z}_n^{(m)})}{p(\mathbf{y}_n^{(\ell)}, \mathbf{z}_n^{(m)}) p(x_{n+1}, \mathbf{z}_n^{(m)})} dx'_{n+1} d\mathbf{y}_n^{'(\ell)} d\mathbf{z}_n^{'(m)} \\
&= \int_{\mathcal{S}} p(x_{n+1}, \mathbf{y}_n^{(\ell)}, \mathbf{z}_n^{(m)}) \log \frac{p(x_{n+1} \mid \mathbf{y}_n^{(\ell)}, \mathbf{z}_n^{(m)})}{p(x_{n+1} \mid \mathbf{z}_n^{(m)})} dx_{n+1} d\mathbf{y}_n^{(\ell)} d\mathbf{z}_n^{(m)} \\
&= CE_{Y \to X \mid Z}.
\end{aligned} \tag{4.25}$$

Sun et al. have investigated the CE in detail, extended the CE idea, and defined the CE on a set of nodes rather than single random variables in (Sun et al., 2015).

Definition 4.1.8. When we measure the information transfer from the set of nodes $\mathcal{J}$ to the set of nodes $\mathcal{I}$ by conditioning on the set of nodes $\mathcal{K}$, the CE becomes (Sun et al., 2015):

$$C_{\mathcal{J} \to \mathcal{I} \mid \mathcal{K}} = h(I_{n+1} \mid K_n) - h(I_{n+1} \mid K_n, J_n). \tag{4.26}$$

Theorem 4.1.9 (CE properties)**.** *Consider a network in* (2.5) *and Markov properties hold. Let N_I be defined as set of parents of I (Sun et al., 2015). Then,*

1. *If* $\mathcal{J} \subset \mathcal{K}$*, then* $CE_{\mathcal{J}\to\mathcal{I}\mid\mathcal{K}} = 0$ *(Redundancy)* (4.27)

2. *If* $\mathcal{N}_{\mathcal{I}} \subset \mathcal{K}$*, then* $CE_{\mathcal{J}\to\mathcal{I}\mid\mathcal{K}} = 0$ *for any set of nodes* $\mathcal{J}$ *(No false positive)* (4.28)

3. *If* $\mathcal{J} \subset \mathcal{N}_{\mathcal{I}}$ *and* $\mathcal{J} \not\subset \mathcal{K}$*, then* $CE_{\mathcal{J}\to\mathcal{I}\mid\mathcal{K}} > 0$ *(True positive)* (4.29)

4. $CE_{\mathcal{J}\to\mathcal{I}\mid\mathcal{K}} = CE_{(\mathcal{K}\cup\mathcal{J})\to\mathcal{I}} - CE_{\mathcal{K}\to\mathcal{I}}$ *(Decomposition)* (4.30)

The results in Theorem 4.1.9 are proved by using the temporally, spatially, and faithfully Markov properties (Sun et al., 2015).

The authors in (Sun et al., 2015) have proposed the optimal CE (OCE) principle based on the properties in Theorem 4.1.9. The OCE principle dictates that the causal parents of a given node in a network consist of the minimal set of nodes that maximizes the CE. Based on this principle, the authors have compared OCE with the existing measures such as the Granger causality (Granger, 1988) and TE to infer the topology of a causal network. The OCE outperforms previously suggested measures at causal network inference. It has been emphasized that the revealing causal and non-causal parents of given nodes in a causal network is a combinatorial problem, and it can be solved using greedy optimization algorithms.

4.2 Information-Theoretic Tools in case of Synchronization

Here, we discuss what information theoretic measures yield in synchronization. Let X, Y, and Z be three outputs of related systems. If X and Y are synchronized, (4.1) becomes,

$$I(X;Y) = H(X) = H(Y). \tag{4.31}$$

If they are independent, $I(X;Y) = 0$.

No information can be gained about one of the systems by gathering the past observations of the other if two systems are synchronized. Hence, the TE in (4.16)

yields,

$$\begin{aligned} TE_{Y\to X} &= h(X_{n+1} \mid \mathbf{X}_n^{(k)}) - h(X_{n+1} \mid \mathbf{X}_n^{(k)}, \mathbf{Y}_n^{(\ell)}) \\ &= h(X_{n+1} \mid \mathbf{X}_n^{(k)}) - h(X_{n+1} \mid \mathbf{X}_n^{(k)}) = 0 \end{aligned} \tag{4.32}$$

for the synchronized systems. Similarly, if X and Z systems are synchronized, CE in (4.20) gives,

$$\begin{aligned} CE_{X\to Y \mid Z} &= h(Y_{n+1} \mid \mathbf{Z}_n^{(m)}) - h(Y_{n+1} \mid \mathbf{Z}_n^{(m)}, \mathbf{X}_n^{(k)}) \\ &= h(Y_{n+1} \mid \mathbf{Z}_n^{(m)}) - h(Y_{n+1} \mid \mathbf{Z}_n^{(m)}) = 0. \end{aligned} \tag{4.33}$$

Additionally, if X and Z systems are in the same cluster, the CE will be zero. On the other hand, there is always an exchange of information regardless of whether X and Y are in the same cluster or not when X and Z are in different clusters.

These observations can be exploited to infer network structure. In light of above discussion, the CE can be the measure to find paths between the systems in a dynamic network. However, CE is zero in the complete synchronization. The problem is discussed in detail later in Chapter 6.

CHAPTER FIVE
ESTIMATION OF ENTROPY BASED MEASURES

Information flow between the systems can be quantified using the measures derived from entropies by observing the state variables. Therefore, accurate estimation of information measures is essential when we aim to quantify the information flow between the systems. Previously, we have mentioned that if the density is known, the information-theoretic measures can be calculated directly. However, we rarely know the pdf in actual cases, and the information-theoretic measures generally have to be estimated from a finite number of samples.

Suppose there is no prior information about the pdf that is desired to be estimated. In this case, non-parametric entropy estimators can be used. First, the pdfs are estimated from the finite number of observations taken from the state variables or outputs. Then, the entropy and the measures derived are estimated using the pdfs. Histogram and kernel density estimator (KDE) are well-known and commonly used methods for non-parametric entropy estimators. However, an important drawback of these estimation methods is that they require many observations before yielding accurate enough results.

Alternatively, the entropy and related measures can be directly estimated using a finite number of discrete observations instead of extracting the pdfs in neighbor-based entropy estimators (Kozachenko & Leonenko, 1987; Singh et al., 2003).

k-nearest neighbor estimator (kNN) is a direct estimation method using the points in the distribution. It is assumed that the pdf is uniformly distributed *locally* in the sample space for the kNN estimator. If the pdf is not uniform or has a direction in sample space, the measures may not be estimated accurately (Lombardi & Pant, 2016). Besides, suppose the correlation between the related random variables is significant, and there is a functional relation between the variables. In that case, the kNN estimator may fail, as pointed out in (Gao et al., 2015; Lord et al., 2018).

The local entropy estimation techniques have been introduced to solve this problem

in recent years. (Lombardi & Pant, 2016; Gao et al., 2015; Lord et al., 2018). Gao et al. have proposed a new estimator that is robust to nonuniform distributions and works efficiently with limited data set (Gao et al., 2015). The authors have modified the kNN entropy estimator by adding a local correction term. When the density has an alignment in sample space, they have transformed the data through the principal component analysis and corrected the local volume of the density. Also, it is shown that the new estimator is more accurate than the standard kNN entropy estimator in nonuniform distributions because of the added local non-uniformity correction term.

Lord et al. have suggested the geometric kNN (g-kNN) entropy estimator similar to the proposed recipe by Gao et al. (Lord et al., 2018). The authors have been motivated by dynamical systems attractors, which may have directions in the sample space. They transformed the data using singular value decomposition to model this non-uniformity. They have chosen the ellipsoids as a local volume element in the new space and made a correction term in entropy estimation by computing the volume of ellipsoids. It has been demonstrated that the g-kNN entropy estimator outperforms the standard entropy estimator on the limited data and locally nonuniform distributions. Lombardi and Pant have proposed a kpN entropy estimator that combines with KDE and kNN estimator in a recent study. (Lombardi & Pant, 2016). It is based on the assumption that the data is locally Gaussian in sample space. The entropy estimation has been achieved by fitting local Gaussian densities for each sample in the distribution. In contrast, the data's overall probability distribution is estimated using the kNN estimator (Lombardi & Pant, 2016). The authors have shown that the kpN entropy estimator is superior to the standard entropy estimator in nonuniform distributions and high-dimensional data.

It has been pointed out that the accurate estimation of information-theoretic measures is not a straightforward task because of the variety of the estimators and their free fine-tuning parameters (Gencaga et al., 2015).

The estimation of information-theoretic measure, specifically TE depends on the distribution of the observations. Hence, the TE may not be estimated accurately when the observations are not uniformly distributed in the sample space. The TEs of

coupled autoregressive Gaussian processes and coupled Lorenz network have been estimated through histogram, Knuth estimator, and KDE (Gencaga et al., 2015). The performance of these estimators has been compared in (Gencaga et al., 2015).

Additionally, Xiong et al. have emphasized that the irrelevant interpretation of the system dynamics can be obtained when we estimate the entropy-based measures by choosing the wrong estimation methods or selecting the wrong parameters (Xiong et al., 2017). Therefore, accurate estimations and choosing the best estimation parameters are essential issues in observed dynamics. The authors have compared the performance of the linear estimator, KDE, and kNN estimator. The entropy, conditional entropy, and information storage of coupled Gaussian processes are estimated using these estimators. The authors have also presented the effects of non-stationarity and have discussed the limitations of the entropy estimators and their bias systematically (Xiong et al., 2017).

In summary, it is crucial to choose a robust estimator that exists in the literature and select the optimal parameters for the estimators. In this section, we have summarized the non-parametric entropy estimator and neighbor-based estimators. Then, we determined the optimal parameters of non-parametric and neighbor-based entropy estimators by estimating the information measures in linear and nonlinear systems. Obviously, the more accurate the estimation of the underlying information measures, the more accurate the quantification of the information flow.

5.1 Entropy Estimation Techniques

In this section, we begin by defining the quality of the measures. Then, we outline the entropy estimation techniques.

Assume that θ is an unknown parameter and a finite set of observations are gathered from independent and identical distribution (i.i.d.) such that $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. Here, we aim to estimate θ from the observed data. A point of estimator $\hat{\theta}$ is defined as

$$\hat{\theta} = T(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N). \tag{5.1}$$

It is expected that $\hat{\theta}$ is close to θ (Shanmugan & Breipohl, 1988).

Definition 5.1.1 (Bias)**.** If $E[\hat{\theta}] \neq \theta$, then $\hat{\theta}$ is biased. The bias of an estimator is defined as:

$$Bias(\hat{\theta}) = E[\hat{\theta}] - \theta. \tag{5.2}$$

Here, $E[\hat{\theta}]$ is expectation of $\hat{\theta}$ over the realizations. If $Bias(\hat{\theta}) = 0$, then the estimator is unbiased.

Definition 5.1.2 (Normalized Bias)**.** If $\theta \neq 0$, the normalized bias is :

$$NB(\hat{\theta}) = \frac{E[\hat{\theta}] - \theta}{\theta}. \tag{5.3}$$

It is expected that $\hat{\theta}$ also has a small statistical variation. Mean square error can measure the variation of $\hat{\theta}$.

Definition 5.1.3 (Mean Square Error (MSE))**.** MSE is defined as follows:

$$\begin{aligned} MSE(\hat{\theta}) &= E[(\hat{\theta} - \theta)^2] \\ MSE(\hat{\theta}) &= [Bias(\hat{\theta})]^2 + Variance(\hat{\theta}). \end{aligned} \tag{5.4}$$

If $Bias(\hat{\theta}) = 0$, then $MSE(\hat{\theta})$ equals to the variance of $\hat{\theta}$.

The standard error of $\hat{\theta}$ is also a measure of the quality of $\hat{\theta}$ and is the positive square root of the variance of $\hat{\theta}$.

Definition 5.1.4 (Normalized Standard Error)**.** If $\theta \neq 0$, the normalized standard error is:

$$NSE(\hat{\theta}) = \frac{\widehat{\sigma_\theta}}{\theta}. \tag{5.5}$$

Here, $\widehat{\sigma_\theta}$ is the positive square root of the variance of $\hat{\theta}$.

Definition 5.1.5 (Normalized Mean Standard Error)**.** If $\theta \neq 0$, the normalized mean standard error is:

$$NMSE(\hat{\theta}) = \frac{E[(\hat{\theta} - \theta)^2]}{\theta^2}. \tag{5.6}$$

If the estimator is unbiased, we use these measures to compare the different estimators. When NB of the estimation is low, the result yields that $NSE(\hat{\theta}) \approx \sqrt{NMSE(\hat{\theta})}$, and it is directly related the performance of the estimator. If we select the parameters of estimators whose NB values are low, we compare the NSE values of the estimators.

Now, we continue with the definition of entropy estimation techniques. Throughout this chapter, we denote the observations as $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N$ (i.e., possibly gathered from a continuous-time random process $\mathbf{X} \in \mathbb{R}^d$ by sampling.)

5.1.1 Histogram estimator

Suppose we have a large set of measurements, perhaps with a nonuniform sampling of outputs, taken from a system or the systems. When the range of data is divided into non-overlapping balls, the number of data inside the balls (frequency) is plotted with respect to the balls. It is called as *histogram* and can approximately represent a probability function. Let $\mathcal{S} \in \mathbb{R}^d$ is a set and $\mathcal{S}$ consists of non-overlapping balls $B_m \in M, m = 1, 2, \ldots, M$ whose diameter is ε. The unknown density function $p_x(\mathbf{x})$ of the random process $\mathbf{X}$ is estimated from N samples via histogram estimator. $\tilde{p}_x(\mathbf{x}, B_m)$ is the estimator of the density is given by:

$$\tilde{p}_x(\mathbf{x}, B_m) = \sum_{B_m \in M} \frac{\text{the number of points inside } B_m}{N\ vol(B_m)} \mathbb{I}(\mathbf{x}_i \in B_m) \qquad i = 1, 2, \ldots, N. \tag{5.7}$$

where $\mathbb{I}(\cdot)$ is an indicator function. $vol(\cdot)$ is the volume of B_m. Then, the entropy of a system is estimated using estimated density function in (5.7).

5.1.2 Kernel density estimator

Consider that we have N observations that are gathered from a process $\mathbf{X}$ with pdf $p_x(\mathbf{x})$. Then KDE is defined as follows:

$$\tilde{p}_x(\mathbf{x}, \mathbf{H}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{K}_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i) \tag{5.8}$$

Here, $\mathbf{H} \in \mathbb{R}^{d \times d}$ is positive definite symmetric bandwidth matrix. $\mathcal{K}_{\mathbf{H}}(\cdot)$ is a kernel function that is expressed as the weighted sum of base functions given by:

$$\mathcal{K}_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-\frac{1}{2}} \mathcal{K}(\mathbf{H}^{-\frac{1}{2}}\mathbf{x}) \tag{5.9}$$

where $\mathcal{K}$ is a kernel base function which satisfies the condition $\int \mathcal{K}(\mathbf{x}) d\mathbf{x} = 1$. There are various types of kernel base functions; the most common one is the Gaussian kernel base function:

$$\mathcal{K}(\mathbf{x}) = \left(\frac{1}{2\pi}\right)^{\frac{d}{2}} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{x}\right). \tag{5.10}$$

When the kernel base function is chosen as the Epanechnikov kernel, we have:

$$\mathcal{K}(\mathbf{x}) = \left(\frac{3}{4}\right)^d \prod_{j=1}^{d} (1 - \mathbf{x}_j^2)[u(\mathbf{x}_j + 1) - u(\mathbf{x}_j - 1)], \quad \mathbf{x}_j = [x_1, x_2, \ldots, x_d]^T. \tag{5.11}$$

where $u(\cdot)$ unit-step function. The entropy and the entropy-based measures are estimated using estimated pdf in (5.8).

5.1.3 k-nearest neighbor entropy estimator

Kozechenko and Leonenko have estimated entropy based on computing between distances of the first neighbor of observations from the distribution in one dimension in (5.12) (Kozachenko & Leonenko, 1987).

$$h(X) \approx \frac{1}{N-1} \sum_{i=1}^{N-1} \log(x_{i+1} - x_i) + \psi(1) - \psi(N), \quad x_i \in \mathbb{R} \tag{5.12}$$

where $\psi(x)$ is the digamma function. $\psi(x) = \Gamma(x)^{-1} d\Gamma(x)/dx$. It satisfies the recursion $\psi(x+1) = \psi(x) + 1/x$ and $\psi(1) = -C$, where $C = 0.57772156$ is Euler-Mascheroni constant. Singh et al. have extended entropy estimation in higher dimensions by using k^{th} nearest neighbors (Singh et al., 2003). Let $\mathbf{X}$ be a continuous-time random variable and its differential entropy is defined in (2.38). A possible choice of an unbiased estimator is:

$$\begin{aligned} \hat{h}(\mathbf{X}) &= -E[\log p_x(\mathbf{x}_i)] \\ &= -\frac{1}{N} \sum_{i=1}^{N} \log p_x(\mathbf{x}_i). \end{aligned} \tag{5.13}$$

Here, $p_x(\mathbf{x}_i)$ is unknown, and the main goal is estimating it through kNNs of $\mathbf{x}_i$. First, the probability distribution function of kNNs of each $\mathbf{x}_i$ is computed. Let us define $p_k(\varepsilon)$ is the distribution function for the distance ε between point $\mathbf{x}_i$ and its kth neighbor.

$p_k(\varepsilon)d\varepsilon$ is the probability that one point is inside the ball $[\varepsilon, \varepsilon + d\varepsilon]$, $(k-1)$ points are inside the ball $[0, \varepsilon]$ and $N-k-1$ points are greater its distance than k^{th} neighbor. The probability $p_k(\varepsilon)d\varepsilon$ is then:

$$\begin{aligned} p_k(\varepsilon)d\varepsilon &= \binom{N-1}{k}\binom{N-2}{k-1}\binom{N-k-1}{N-k-1}\frac{dP_i(\varepsilon)}{d\varepsilon}d\varepsilon(P_i(\varepsilon))^{k-1}(1-P_i(\varepsilon))^{N-k-1} \\ &= \frac{(N-1)!}{1!(N-2)!}\frac{(N-2)!}{(k-1)!(N-k-1)!}\frac{dP_i(\varepsilon)}{d\varepsilon}d\varepsilon(P_i(\varepsilon))^{k-1}(1-P_i(\varepsilon))^{N-k-1} \\ &= k\binom{N-1}{k}\frac{dP_i(\varepsilon)}{d\varepsilon}d\varepsilon(P_i(\varepsilon))^{k-1}(1-P_i(\varepsilon))^{N-k-1}. \end{aligned} \tag{5.14}$$

$P_i(\varepsilon)$ is the mass of the ε ball centered at $\mathbf{x}_i$. The expected value of the logarithm of $P_i(\varepsilon)$ is defined by:

$$\begin{aligned} E(\log P_i(\varepsilon)) &= \int_0^\infty d\varepsilon p_k(\varepsilon)\log(P_i(\varepsilon)) \\ &= \int_0^1 k\binom{N-1}{k}P_i(\varepsilon)(P_i(\varepsilon))^{k-1}(1-P_i(\varepsilon))^{N-k-1}\log(P_i(\varepsilon))d\varepsilon \\ &= \psi(k) - \psi(N) \end{aligned} \tag{5.15}$$

where, $\psi(\cdot)$ denotes the digamma function. Besides, the mass of $P_i(\varepsilon)$ can be given as:

$$\begin{aligned} P_i(\varepsilon) &= \int_{B(\varepsilon,\mathbf{x}_i)} p_x(\mathbf{x})d\mathbf{x} \\ P_i(\varepsilon) &\approx V p_x(\mathbf{x}_i) \end{aligned} \tag{5.16}$$

V is the volume of d-dimensional ε ball, i.e., $V = c_d\varepsilon_i^d$. If ε is equal to 1, then c_d is volume of unit-ball in given metric. c_d is equal to 1 for maximum norm and $c_d = \pi^{d/2}/\Gamma(1+d/2)/2^d$ for Euclidean distance. The estimation of $p_{\mathbf{x}}(\mathbf{x_i})$ can be carried out by:

$$
\begin{aligned}
p_{\mathbf{x}}(\mathbf{x_i}) &\approx \frac{P_i(\varepsilon)}{V} \\
p_{\mathbf{x}}(\mathbf{x_i}) &\approx \frac{P_i(\varepsilon)}{c_d \varepsilon_i^d}.
\end{aligned} \tag{5.17}
$$

The logarithm of the density function is equal to:

$$
\begin{aligned}
\log p_{\mathbf{x}}(\mathbf{x_i}) &\approx \log P_i(\varepsilon) - \log c_d - d \log \varepsilon_i \\
&\approx \psi(k) - \psi(N) - \log c_d - d \log \varepsilon_i.
\end{aligned} \tag{5.18}
$$

Taking expectations of both sides yields,

$$
\begin{aligned}
E[\log p_{\mathbf{x}}(\mathbf{x_i})] &\approx \psi(k) - \psi(N) - \log c_d - dE[\log \varepsilon_i] \\
E[\log p_{\mathbf{x}}(\mathbf{x_i})] &\approx \psi(k) - \psi(N) - \log c_d - \frac{d}{N} \sum_{i=1}^{N} \log \varepsilon_i
\end{aligned} \tag{5.19}
$$

Here, ε_i is twice the distance from x_i to k^{th} neighbor. Finally, the entropy of a random variable $\mathbf{X}$ is approximated as:

$$
\begin{aligned}
\hat{h}(\mathbf{X}) &= -E[\log p_x(\mathbf{x}_i)] \\
&= -\psi(k) + \psi(N) + \log c_d + \frac{d}{N} \sum_{i=1}^{N} \log \varepsilon_i.
\end{aligned} \tag{5.20}
$$

In a similar analogy, the entropy of $\mathbf{Z} = (X_1, X_2)$ joint random variable is computed by changing the variables as $d_z = d_x + d_y$ and $c_d = c_{d_x} c_{d_y}$:

$$
\hat{h}(X_1, X_2) = -\psi(k) + \psi(N) + \log(c_{d_{x_1}} c_{d_{x_2}}) + \frac{d_{x_1} + d_{x_2}}{N} \sum_{i=1}^{N} \log \varepsilon_i. \tag{5.21}
$$

The MI and the TE can be estimated using kNN-based estimators. Kraskov-Stögbauer-Grassberger (KSG) estimator is an extension of Singh's entropy estimator (Kraskov et al., 2004). Kraskov et al. have proposed two different methods to estimate MI by searching the neighbors in the marginal spaces. The authors have studied the performance of the estimators by varying dimensions, the number of points, and neighbors, respectively. The estimation of MI is given as in (5.22) and (5.23):

$$\hat{I}(X_1; X_2) = \psi(k) + \psi(N) - \frac{1}{N}\sum_{i=1}^{N}\left[\psi(\eta_{x_1}(i)) + \psi(\eta_{x_2}(i))\right] \tag{5.22}$$

$$\hat{I}(X_1; X_2) = \psi(k) + \psi(N) - \frac{1}{k} - \frac{1}{N}\sum_{i=1}^{N}[\psi(\eta_{x_1}(i)) + \psi(\eta_{x_2}(i))] \tag{5.23}$$

$\eta_{x_1}(i)$ and $\eta_{x_2}(i)$ are the number of points with $\|x_{1,i} - x_{1,j}\| \leq \varepsilon_{x_1}(i)/2$ and $\|x_{2,i} - x_{2,j}\| \leq \varepsilon_{x_2}(i)/2$, respectively. According to Kraskov et al., the MI of the d dimensional variable is described by:

$$\hat{I}(\mathbf{X}) \equiv (d-1)\psi(N) + \psi(k) - (d-1)/k - \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\psi(n_{x_j}(i)) \tag{5.24}$$

where $\eta_{x_j}(i)$ is the number of points inside the distance $\|x_{j,i} - x_{j,\tilde{j}}\| \leq \varepsilon_{x_j}(i)/2$ in marginal spaces $j = 1, 2, \ldots d$, respectively.

The TE has been estimated through the kNNs approach in a study by Zhu et al. (Zhu et al., 2015).

$$\widehat{TE}_{X_1 \to X_2} = \psi(k) + \frac{1}{N}\sum_{i=1}^{N}\left[\psi(\eta_{x_{2,n}}(i)+1) - \psi(\eta_{x_{1,n},x_{2,n}}(i)+1) - \psi(\eta_{x_{2,n+1},x_{2,n}}(i)+1)\right]. \tag{5.25}$$

Here, $\eta_{x_{2,n}}(i)$, $\eta_{x_{1,n},x_{2,n}}(i)$, $\eta_{x_{2,n+1},x_{2,n}}(i)$ are the number of points inside the ball centered $x_{2,n}$, $(x_{1,n}, x_{2,n})$, $(x_{2,n+1}, x_{2,n})$ of diameter $\varepsilon_{x_{2,n}}$, $\varepsilon_{(x_{1,n},x_{2,n})}$, $\varepsilon_{(x_{2,n+1},x_{2,n})}$, respectively.

We have proposed a kNN estimator for CE inspired by Kraskov and Zhu et al. Given the observations, the efficient estimation of CE can be achieved by using the kNN entropy estimation method practically.

$$\widehat{CE}_{X_1 \to X_2 \mid X_3} = \psi(k) + \frac{1}{N} \sum_{i=1}^{N} \Big[\psi(\eta_{x_{3,n}}(i) + 1) - \psi(\eta_{(x_{3,n},x_{2,n+1})}(i) + 1) - \psi(\eta_{(x_{3,n},x_{1,n})}(i) + 1) \Big]. \tag{5.26}$$

$\psi(\cdot)$ is the digamma function, k is the predefined number of neighbors, and N is the number of observation points. $\eta_{x_{3,n}}(i)$, $\eta_{(x_{3,n},x_{2,n+1})}(i)$, $\eta_{(x_{3,n},x_{1,n})}(i)$ are the number of points inside the balls $\mathcal{B}(x_{3,n}, \varepsilon_{x_{3,n}})$, $\mathcal{B}((x_{3,n}, x_{2,n+1}), \varepsilon_{(x_{3,n},x_{2,n+1})})$, $\mathcal{B}((x_{3,n}, x_{1,n}), \varepsilon_{(x_{3,n},x_{1,n})})$ where $\mathcal{B}(y, \varepsilon)$ is the ball centered at y with the radius ε. ε is determined by calculating the distance between ℓ^{th} point and the nearest k-neighbor of it.

Figure 5.1 illustrates the number of points determined inside the ball. ε is determined by calculating the distance between ℓth point and the nearest k-neighbor of it. $\mathcal{B}(\mathbf{x}_\ell, \varepsilon(\ell))$ is plotted on dashed rectangular using $\|\cdot\|_\infty$ norm. The number of points inside the ball is obtained as $\eta_{\mathbf{x}_\ell}(\ell) = 6$.

We have utilized the CE estimator in (5.26) in the following chapters since we have based our proposed measure on the estimation of CE that reveals the connections in the network. Additionally, the estimation of CE has been used for analyzing the information transfer between the inter- and intra-cluster nodes in the network.

5.1.4 kpN entropy estimator

Lombardi and Pant have proposed an entropy estimator which combines KDE with a kNN entropy estimator in (Lombardi & Pant, 2016). The probability mass function is computed for each $\mathbf{x}_i$ point inside p_k^{th} nearest neighbors of $\mathbf{x}_i$ via using Gaussian KDE.

Figure 5.1 An illustration of the number of points determination in the kNN estimator. The distance between $\mathbf{x}_\ell$ point and its k nearest neighbor point is calculated as $\varepsilon(\ell)$.

Similar to the kNN estimator, the pdf is estimated according to the distance between $\mathbf{x}_i^{th}$ and its kth nearest neighbor. The probability density inside $B(\varepsilon, \mathbf{x}_i)$ is defined as:

$$p_x(x) \approx \frac{1}{(2\pi)^d |\mathbf{K}|^d} e^{\left(-0.5(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{K}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)} \tag{5.27}$$

where $\boldsymbol{\mu}$ is mean vector and $\mathbf{K}$ is covariance matrix of p_k neighbors of $\mathbf{x}_i$. The pdf is assumed to be rational with a Gaussian function approximated by using ${p_k}^{\text{th}}$ nearest neighbors of $\mathbf{x}_i$ ($p_k \geq k$) (Lombardi & Pant, 2016). The density inside $B(\varepsilon, \mathbf{x}_i)$ is computed through this local Gaussian function:

$$p_x(\mathbf{x}) \approx p_x(\mathbf{x}_i) \frac{g(\mathbf{x})}{g(\mathbf{x}_i)} \tag{5.28}$$

where

$$g(\mathbf{x}) = e^{-0.5(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{K}^{-1}(\mathbf{x}-\boldsymbol{\mu})}, \tag{5.29}$$

and

$$g(\mathbf{x}_i) = e^{-0.5(\mathbf{x}_i-\boldsymbol{\mu})^T \mathbf{K}^{-1}(\mathbf{x}_i-\boldsymbol{\mu})}. \tag{5.30}$$

Accordingly, the mass of ε-ball is equal to:

$$P_i = p_x(\mathbf{x}_i)\frac{G_i}{g(\mathbf{x}_i)} \tag{5.31}$$

where

$$G_i = \int_{B(\varepsilon,\mathbf{x}_i)} g(\mathbf{x})d\mathbf{x} \tag{5.32}$$

In the kpN estimator, the density inside $B(\varepsilon, \mathbf{x}_i)$ is assumed Gaussian as opposed to the uniform density assumption in the kNN entropy estimator. The mean and the covariance of $\mathbf{x}_i$ inside the ball consists of p_k^{th} nearest neighbors are computed, to derive $g(\mathbf{x}_i)$ from (5.30). G_i in (5.32) contains the Gaussian definite integral inside $B(\varepsilon, \mathbf{x}_i)$ in (5.33). The entropy of $\mathbf{X}$ can be estimated by using (5.28) and (5.31):

$$\hat{h}(\mathbf{X}) = -E[\log p_x(\mathbf{x_i})] \approx -\psi(k) + \psi(N) - \frac{1}{N}\sum_{i=1}^{N}\log g(\mathbf{x}_i) + \frac{1}{N}\sum_{i=1}^{N}\log G_i. \tag{5.33}$$

We have to consider the optimal parameters of the estimators when estimating the entropy and the entropy-based measures. The number of observations is crucial in overall estimation methods since it is directly used in estimating pdf or entropy. Thus, the determination of the number of observations, diameter size, and the boundaries of the interval has a significant role in the estimation of pdf for the histogram estimator (Wand & Jones, 1994). In KDE, the number of observations, the selection of

bandwidth matrix **H**, and the kernel base function are parameters that change the performance of the estimators. Besides, KDE makes accurate estimations in low dimensions, and it can fail when the dimension increases due to the curse of dimensionality (Gray & Moore, 2003). When the observations are uniformly distributed locally, the kNN estimator succeeds in estimating the entropy, as mentioned before. The main advantage of the kNN estimator is to compute the entropy fast, even in the high dimensions. However, the entropy estimator can fail when the density has a specific alignment in the sample space. kpN entropy estimator can estimate the entropy more accurately for locally nonuniform data by computing the p_k neighborhood of the probability mass function of each point through local Gaussian kernel density functions. It can represent the tails of the data better if the data has spread in the probability distribution since this estimator also uses a Gaussian kernel density function. Therefore, it estimates the entropy more accurately. The number of observations, k-neighbors, and the number of p_k neighbors must be chosen carefully for the kpN entropy estimator.

5.2 Simulations and Results

This section compares the histogram, KDE, kNN, and kpN entropy estimators on the same data. We have used the coupled autoregressive processes and bistable system to test the performance of the estimators because we know the theoretical values of the entropies for these systems. Therefore, we have measured the performance of the estimators by comparing our estimations and the theoretical values. We have shown that the accuracy of the estimation is increased by using the selection of the optimal parameters in linear and nonlinear systems. Finally, we have performed the estimators on a real dataset because we do not know the underlying dynamics of the systems, and the values of entropies are unknown. We have illustrated our result by using statistical significance hypothesis tests.

We have measured the performance of entropy estimators by computing the normalized bias (NB) and normalized standard error (NSE) of the estimators. We

have assumed that if the estimation error does not exceed 5% of the theoretical value, it is considered acceptable and accurate. Simulations have been performed on a workstation in parallel with Intel®Xenon®E5630 processors in the Python environment.

5.2.1 Coupled autoregressive processes

Let us couple d linear autoregressive processes with Gaussian white noise increments, and their state variables are defined in matrix form:

$$\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n + \boldsymbol{\eta}_n \tag{5.34}$$

with $\mathbf{x}_n = [x_{1,n}, x_{2,n}, \ldots, x_{d,n}] \in \mathbb{R}^d$ is the state variables at time n, $\mathbf{A} \in \mathbb{R}^{d\times d}$ is the coupling matrix whose elements are real and $\boldsymbol{\eta}_n = [\eta_{1,n}, \eta_{2,n}, \ldots, \eta_{d,n}]$ is the Gaussian noise vector. The MI between two Gaussian processes is written in terms of entropy using (4.2) and (4.13) which are stated in Chapter 2:

$$\begin{aligned} I(\mathbf{X}_1;\mathbf{X}_2) &= h(\mathbf{X}_{1,n}) + h(\mathbf{X}_{2,n}) - h(\mathbf{X}_{1,n}, \mathbf{X}_{2,n}) \\ &= \frac{1}{2}\log\frac{det(\mathbf{K}(\mathbf{X}_{1,n}))det(\mathbf{K}(\mathbf{X}_{2,n}))}{det(\mathbf{K}(\mathbf{X}_{1,n}, \mathbf{X}_{2,n}))} \end{aligned} \tag{5.35}$$

Similarly, the TE is defined in terms of entropy and covariance matrix of autoregressive processes using (4.17) and (4.13) which are stated in Chapter 2:

$$\begin{aligned} TE_{X_1\to X_2} &= h(\mathbf{X}_{2,n+1}, \mathbf{X}_{2,n}^{(\ell)}) - h(\mathbf{X}_{2,n}^{(\ell)}) + h(\mathbf{X}_{2,n}^{(\ell)}, \mathbf{X}_{1,n}^{(k)}) - h(\mathbf{X}_{2,n+1}, \mathbf{X}_{2,n}^{(\ell)}, \mathbf{X}_{1,n}^{(k)}) \\ &= \frac{1}{2}\log\frac{det(\mathbf{K}(\mathbf{X}_{2,n+1}, \mathbf{X}_{2,n}^{(\ell)}))det(\mathbf{K}(\mathbf{X}_{2,n}^{(\ell)}, \mathbf{X}_{1,n}^{(k)}))}{det(\mathbf{K}(\mathbf{X}_{2,n+1}, \mathbf{X}_{2,n}^{(\ell)}, \mathbf{X}_{1,n}^{(k)}))det(\mathbf{K}(\mathbf{X}_{2,n}^{(\ell)}))}. \end{aligned} \tag{5.36}$$

where $\mathbf{x}_{1,n}^{(k)} = \{\mathbf{x}_{1,n}, \mathbf{x}_{1,n-1}, \mathbf{x}_{1,n-2}, \ldots, \mathbf{x}_{1,n-k+1}\}$ denotes the measurement of a random

variable $\mathbf{X}_{1,n}$ from n to $n-k+1$ ($k \in \mathbb{Z}^+$). If the covariance matrix and time-delayed covariance matrix are known, the theoretical values of MI and TEs between the systems in the network are estimated. In short, TE from $\mathbf{X}_1$ to $\mathbf{X}_2$ is equal to:

$$TE_{X_1 \to X_2} = \frac{1}{2} \log \frac{\left(\mathbf{K}(0)_{x_2x_2}\mathbf{K}(0)_{x_1x_1} - \mathbf{K}(0)^2_{x_1x_2}\right)\left(\mathbf{K}(0)^2_{x_2x_2} - \mathbf{K}(1)^2_{x_2x_2}\right)}{\mathbf{K}(0)_{x_2x_2}\left(\mathbf{K}(0)^2_{x_2x_2}\mathbf{K}(0)_{x_1x_1} + 2\mathbf{K}(1)_{x_2x_2}\mathbf{K}(0)_{x_1x_2}\mathbf{K}(1)_{x_2x_1} \cdots \right.} \\ \left. -\mathbf{K}(1)^2_{x_2x_2}\mathbf{K}(0)_{x_1x_1} - \mathbf{K}(0)_{x_2x_2}\mathbf{K}(0)^2_{x_1x_2} - \mathbf{K}(0)_{x_2x_2}\mathbf{K}(1)^2_{x_2x_1}\right) \tag{5.37}$$

Here, $\mathbf{K}(0)_{x_2x_2} = E[X_{2,n}X_{2,n}]$, $\mathbf{K}(0)_{x_1x_2} = E[X_{1,n}X_{2,n}]$, $\mathbf{K}(1)_{x_2x_2} = E[X_{2,n+1}X_{2,i}]$. Let us denote $\alpha_{x_1x_2} = \left(\mathbf{K}(0)_{x_2x_2}\mathbf{K}(1)_{x_2x_1} - \mathbf{K}(0)_{x_2x_1}\mathbf{K}(1)_{x_2x_2}\right)^2$ and $\beta_{x_1x_2} = \left(\mathbf{K}(0)_{x_2x_2}\mathbf{K}(0)_{x_1x_1} - \mathbf{K}(0)^2_{x_1x_2}\right)\left(\mathbf{K}(0)^2_{x_2x_2} - \mathbf{K}(1)^2_{x_2x_2}\right)$. Then, the TE is rewritten as follows (Sun et al., 2015):

$$TE_{X_1 \to X_2} = \frac{1}{2} \log\left(1 + \frac{\alpha_{x_1x_2}}{\beta_{x_1x_2} - \alpha_{x_1x_2}}\right). \tag{5.38}$$

The theoretical values of MI and TE are obtained by calculating the elements of the covariance matrix in (5.35) and (5.36). Let us define three coupled autoregressive processes using (5.34):

$$\begin{bmatrix} x_{1,n+1} \\ x_{2,n+1} \\ x_{3,n+1} \end{bmatrix} \begin{bmatrix} 0.5 & 0 & 0 \\ 0.5 & 0.6 & 0 \\ 0.2 & 0.7 & 0.3 \end{bmatrix} \begin{bmatrix} x_{1,n} \\ x_{2,n} \\ x_{3,n} \end{bmatrix} + \begin{bmatrix} \eta_{1,n} \\ \eta_{2,n} \\ \eta_{3,n} \end{bmatrix}. \tag{5.39}$$

The mean vector of noise has been chosen as $\boldsymbol{\mu} = [0, 0, 0]^T$ and the covariance matrix of the noise has been selected as $\mathbf{K} = [[1, 0, 0], [0, 1, 0], [0, 0, 1]]$. The time delays have been chosen as $k = \ell = 1$. MI of the systems are equal to $I(X_1; X_2) = 0.0256$ *nats*, $I(X_1; X_3) = 0.0118$ *nats*, and $I(X_2; X_3) = 0.1899$ *nats*,

respectively. While $TE_{X_2 \to X_1} = TE_{X_3 \to X_1} = TE_{X_3 \to X_2} = 0$, $TE_{X_1 \to X_2} = 0.092280$ *nats*, $TE_{X_1 \to X_3} = 0.03894$ *nats*, $TE_{X_2 \to X_3} = 0.28262$ *nats* for the system in (5.39).

The simulations have been repeated 1000 times. The information measures, namely, MI and TE, have been estimated for a different number of observations. The number of observations N has varied from 256 to 2048 in the histogram, KDE, and kNN estimators. kpN entropy estimator has three parameters: the number of observations, the number of k-neighbors, and the number of p_k neighbors. In general, when the number of observations is increased, the performance of the kpN estimator is increased (Lombardi & Pant, 2016). Therefore, the number of observations has been chosen large enough, e.g., $N = 4096$ and $N = 8192$, respectively, in the kpN entropy estimator.

To begin with, the histogram of MI and TEs between the systems has been estimated using the density function in (5.7). The estimations have been also evaluated for different diameters ε with no overlapping and $2^{-3} \leq \varepsilon \leq 2^{1}$. We have illustrated the NB and NSE of MI of the systems in Figure 5.2 and 5.3, respectively.

Figure 5.2 The NB of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using histogram estimator with respect to N versus ε

When the number of observations and the interval diameter are increased, the NBs of MI between the systems decrease. It is shown that the NSEs of MI between the

Figure 5.3 The NSE of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using histogram estimator with respect to N versus ε

systems are also decreased when the number of observations and the diameter of an interval are increased. The NSEs of MI are increased in case of $\varepsilon \gg 2$, but this result are not plotted in Figure 5.3. We have seen an interval of ε, which makes NB of MIs minimum. If the ε is greater than the support set of all observations, the histogram estimator has only one interval. It yields that accurate estimation can be obtained if the data is (almost) uniform. According to the figures, we have had the minimum NB and NSE of MI when $N = 2^{11}$ and $\varepsilon = 1.68$ among all parameters. As a result, the accurate estimations of MI have been obtained as the number of observations increased and the diameter is in the range of $1 \leq \varepsilon \leq 2$ in the histogram estimator.

Similarly, when we have examined Figures 5.4 and 5.5, the NBs of TEs have been decreased in case of the number of samples and the diameter of an interval is increased. The accurate estimations have been achieved for all N and $\varepsilon > 1$ in Figure 5.4. However, the NSEs of TE have been decreased when $N > 2^{10}$ and $\varepsilon > 1$ in Figure 5.5. It is obvious that the minimum NBs and NSEs of TEs have been obtained in the case of $(N, \varepsilon) = (2^{11}, 2)$. Besides, it has been noticed that the NB and the NSE of information measures of the systems X_1 and X_3 that are not directly connected systems are higher than the NB and the NSE of information measures of the systems that are directly connected, i.e., X_1 and X_2 or X_2 and X_3.

Figure 5.4 The NB of $\widehat{TE}_{X_1\to X_2}$, $\widehat{TE}_{X_1\to X_3}$, $\widehat{TE}_{X_2\to X_3}$ $\widehat{TE}_{X_2\to X_1}$ using histogram estimator with respect to N versus ε

Figure 5.5 The NSE of $\widehat{TE}_{X_1\to X_2}$, $\widehat{TE}_{X_1\to X_3}$, $\widehat{TE}_{X_2\to X_3}$ $\widehat{TE}_{X_2\to X_1}$ using histogram estimator with respect to N versus ε

As a second estimator, we have examined KDE. We have chosen the Gaussian kernel in (5.10) and Epanechnikov kernel in (5.11) as a kernel base functions. The simulations have been performed by varying the number of observations and the bandwidth matrix. The diagonal elements of bandwidth matrix has been chosen as $0.1 \leq h \leq 1$ where $|h| = |h_{ii}|\ \forall\ i = 1, 2, \ldots, d$. Other elements are zero $h_{ij} = 0\ \forall\ i \neq j$. The plots have been illustrated against the diagonal elements of the matrix. We have

Figure 5.6 The NB of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using KDE with Gaussian kernel base function for varying parameters of $(N, |h|)$

Figure 5.7 The NSE of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using KDE with Gaussian kernel base function for varying parameters of $(N, |h|)$

used a 2-dimensional kernel base function in estimating MI and a 3-dimensional kernel base function in estimation TEs. While the NBs and the NSEs of MI have been illustrated in Figures 5.6, 5.7, 5.8 and 5.9, the NBs and the NSEs of TEs between the systems have shown in Figures 5.10, 5.11, 5.12 and 5.13.

Figure 5.6 has indicated that the NB of MIs between the systems is fairly low for all N and $|h| > 0.3$ when we have estimated the MIs with the Gaussian kernel function. It is seen that NSE of MI is significantly low in Figure 5.7. It is expected that the

Figure 5.8 The NB of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using KDE with Epanechnikov kernel base function for varying parameters of $(N, |h|)$

Figure 5.9 The NSE of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using KDE with Epanechnikov kernel base function for varying parameters of $(N, |h|)$

performance of a KDE is high when we have estimated the measures with a Gaussian kernel density function, since the states of coupled autoregressive processes are driven by Gaussian white noise. If the kernel function is chosen as the Epanechnikov kernel base function, the NB of MIs between the systems is in an acceptable range for all N and $|h| > 0.6$ as shown in Figure 5.8. The NSE of MIs between the systems does not change considerably by fixing bandwidth and as the number of samples increases in Figure 5.9. However, if the bandwidth increases, the NSE of MIs between the systems decreases, as Figure 5.9 indicates. It is appropriate to choose any N and $|h| = 0.9$

Figure 5.10 The NB of $\widehat{TE}_{X_1\to X_2}$, $\widehat{TE}_{X_1\to X_3}$, $\widehat{TE}_{X_2\to X_3}$ $\widehat{TE}_{X_2\to X_1}$ using KDE with Gaussian kernel base function for varying parameters of $(N, |h|)$

Figure 5.11 The NSE of $\widehat{TE}_{X_1\to X_2}$, $\widehat{TE}_{X_1\to X_3}$, $\widehat{TE}_{X_2\to X_3}$ $\widehat{TE}_{X_2\to X_1}$ using KDE with Gaussian kernel base function for varying parameters of $(N, |h|)$

to get a more accurate estimation in Epanechnikov KDE. When we have compared Figures 5.7 and 5.9, it has been observed that the NSE of MIs in the estimation with Epanechnikov base function is greater than the NSE of MIs in the estimation with Gaussian function.

Similarly, when the TEs between the systems have been estimated with Gaussian KDE, the NB of TEs has been acceptable for all N and $|h| > 0.4$ in Figure 5.10. The NSE of TEs is smaller than 0.05 for all N and $|h|$ as Figure 5.11 illustrates. When the kernel function is Epanechnikov, the NB of TEs is small in case of $(N,\ |h|) > (2^9, 0.9)$

Figure 5.12 The NB of $\widehat{TE}_{X_1\to X_2}$, $\widehat{TE}_{X_1\to X_3}$, $\widehat{TE}_{X_2\to X_3}$ $\widehat{TE}_{X_2\to X_1}$ using KDE with Epanechnikov kernel base function for varying parameters of $(N, |h|)$

Figure 5.13 The NSE of $\widehat{TE}_{X_1\to X_2}$, $\widehat{TE}_{X_1\to X_3}$, $\widehat{TE}_{X_2\to X_3}$ $\widehat{TE}_{X_2\to X_1}$ using KDE with Epanechnikov kernel base function for varying parameters of $(N, |h|)$

in Figure 5.12. The NSE of TEs is also small when $((N,\ |h|) = (2^{10}, 0.8)$ is selected in Epanechnikov KDE as shown in Figure 5.13. The choice of Gaussian kernel function yields to obtain the best estimations since we observe the coupled Gaussian processes. We have compared Figures 5.8 and 5.12, it is seen that the dimension of kernel base function increases. Therefore, the NSE of estimations of information measures increases when the kernel dimension is increased.

The performance of the kNN entropy estimator has also been investigated. The information measures have been estimated for varying the number of observations and

Figure 5.14 The NB of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using kNN entropy estimator with respect to N versus k

Figure 5.15 The NSE of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using kNN entropy estimator with respect to N versus k

neighbors. The number of neighbors has varied from 1 to 10. The MIs and TEs between the systems have been estimated using (5.23) and (5.25), respectively. While Figures 5.14 and 5.15 have illustrated the NB and the NSE of MIs between the systems, the NB and the NSE of TEs have shown in Figures 5.16 and 5.17, respectively.

We have investigated the NB of MIs between the systems in Figure 5.14 and obtained good estimations when $N = 2^{10}$ and $k > 7$. Furthermore, the NSE of MIs is smaller than 0.02 in case of $N = 2^{10}$, $k > 7$ as Figure 5.15 depicts. The acceptable TE

Figure 5.16 The NB of $\widehat{TE}_{X_1\to X_2}$, $\widehat{TE}_{X_1\to X_3}$, $\widehat{TE}_{X_2\to X_3}$, $\widehat{TE}_{X_2\to X_1}$ using kNN entropy estimator with respect to N versus k

Figure 5.17 The NSE of $\widehat{TE}_{X_1\to X_2}$, $\widehat{TE}_{X_1\to X_3}$, $\widehat{TE}_{X_2\to X_3}$, $\widehat{TE}_{X_2\to X_1}$ using kNN entropy estimator with respect to N versus k

estimations have been obtained in $N = 2^{10}$ and $k > 7$ by looking at the NB of TEs in Figure 5.16. It is seen that the NSEs of TE are also smaller than 0.02, similar to that of the NSE of MIs in the kNN estimator as shown in Figure 5.17. Besides, kNN estimators have outperformed histogram estimators and KDE by inspecting the NB and NSE of MIs and TEs. kNN estimators are more robust than histogram estimators, and KDE, if the density is locally uniform, N and k are not too small.

Finally, we have focused on the performance of the kpN entropy estimator. We have chosen the number of neighbors between 1 and 10, and p_k/N has been selected between 0.01 and 0.1 with an increase of 0.01. The NB and NSE of MIs between the

Figure 5.18 The NB of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using kpN entropy estimator with respect to p_k/N versus k when $N = 4096$

Figure 5.19 The NB of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using kpN entropy estimator with respect to p_k/N versus k when $N = 8192$

systems have been illustrated for $N = 4096$ and $N = 8192$ in Figures 5.18, 5.19, 5.20, and 5.21. Moreover, Figures 5.22, 5.23, 5.24 and 5.25 have shown that the NB and the NSE of TEs between the systems for $N = 4096$ and $N = 8192$, respectively.

The results have shown that the NB of MIs between the systems is decreased when the number of samples increases. Hence, kpN has estimated the entropy-based measures accurately. We have obtained the acceptable estimations by investigating the NBs of MIs between the systems when $p_k/N = 0.05$ and for all k in Figures 5.18

Figure 5.20 The NSE of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using kpN entropy estimator with respect to p_k/N versus k when $N = 4096$

Figure 5.21 The NSE of $\hat{I}(X_1, X_2)$, $\hat{I}(X_1, X_3)$, $\hat{I}(X_2, X_3)$ using kpN entropy estimator with respect to p_k/N versus k when $N = 8192$

and 5.19. It is seen that the NSE of MIs between the systems takes minimum value when $N = 4096$, $p_k/N = 0.1$ and $k = 10$, $N = 8192$, $p_k/N = 0.1$ and $k = 10$ in Figures 5.20 and 5.21, respectively. The NB of TEs has been decreased when p_k/N has been increased and for all k in $N = 4096$ and $N = 8192$, as shown in Figures 5.22 and 5.23. Additionally, the NSE of TEs has become lower than 0.05 for all $p_k/N, k$ as Figures 5.24 and 5.25 illustrate. While the variation of p_k/N has not been changed, the NSE of MIs and TEs between the systems significantly, the NSEs of estimations have been altered by changing k. We have expected the performance of kpN estimators to be high since the local distribution is Gaussian in (5.34).

Figure 5.22 The NB of $\widehat{TE}_{X_1\to X_2}$, $\widehat{TE}_{X_1\to X_3}$, $\widehat{TE}_{X_2\to X_3}$ $\widehat{TE}_{X_2\to X_1}$ using kpN entropy estimator with respect to p_k/N versus k when $N = 4096$

Figure 5.23 The NB of $\widehat{TE}_{X_1\to X_2}$, $\widehat{TE}_{X_1\to X_3}$, $\widehat{TE}_{X_2\to X_3}$ $\widehat{TE}_{X_2\to X_1}$ using kpN entropy estimator with respect to p_k/N versus k when $N = 8192$

In general, estimators' performance increases as the number of observations increases, as expected in all estimators. The selection of appropriate volume size is important to achieve less NBs and NSEs in the histogram estimator. Using Gaussian KDE as expected in estimating the information measures of coupled Gaussian processes is advantageous. If the different kernel density function is used in estimating Gaussian distribution, the NB and the NSE are higher than the Gaussian kernel function. When the dimension of the data is increased in the Epanechnikov kernel function, it is observed that the NB of estimations is also increased.

If the estimations are carried out with enough observations, and kNN entropy

Figure 5.24 The NSE of $\widehat{TE}_{X_1\to X_2}$, $\widehat{TE}_{X_1\to X_3}$, $\widehat{TE}_{X_2\to X_3}$ $\widehat{TE}_{X_2\to X_1}$ using kpN entropy estimator with respect to p_k/N versus k when $N = 4096$

Figure 5.25 The NSE of $\widehat{TE}_{X_1\to X_2}$, $\widehat{TE}_{X_1\to X_3}$, $\widehat{TE}_{X_2\to X_3}$ $\widehat{TE}_{X_2\to X_1}$ using kpN entropy estimator with respect to p_k/N versus k when $N = 8192$

estimators are used, we have obtained the estimations with minimum errors. In the kpN entropy estimator, the choice of k and p_k/N change the NBs of estimations. However, the NSE slightly changes because of the variation of the p_k/N ratio. The NSE of estimations is changed for the different values of k. We also have determined the acceptable estimations for Gaussian distributions in the kpN estimator, which are used efficiently in both KDE and the kNN entropy estimator. Additionally, the NB and NSE of information measures of systems that are not directly coupled are greater than the NB and NSE of information measures of systems that are directly coupled such as $NB(\hat{I}(X_1, X_3)) > NB(\hat{I}(X_1, X_2))$, $NSE(\hat{I}(X_1, X_3)) > NSE(\hat{I}(X_1, X_2))$, $NB(\widehat{TE}(X_1 \to X_3)) > NB(\widehat{TE}(X_1 \to X_2))$,

$NSE(\widehat{TE}(X_1 \to X_3)) > NSE(\widehat{TE}(X_1 \to X_2))$ for all estimators.

The measures derived from entropies are successfully estimated using kNN entropy estimators. Furthermore, it has been clear that the performance of the Gaussian KDE is high for the coupled Gaussian autoregressive processes. Thus, the performance of the estimators is tested in a nonlinear bistable system in the next section.

5.2.2 Bistable system

We have studied the performance of entropy estimators by observing the output of a bistable system since it is driven by stochastic white noise and is a nonlinear system. We have expected that Gaussian KDE and kpN estimators perform well for coupled Gaussian distributions. However, the estimators' performance in such a nonlinear system is unclear.

Consider the system given in (2.4). The initial pdf of random variable $\mathbf{X}$ is denoted as $p(\mathbf{x}_0, 0) = p_0(\mathbf{x}_0)$. The pdf $p(\mathbf{x}, t)$ of (2.4) satisfies:

$$\frac{\partial p(\mathbf{x}, t)}{\partial x} = \frac{1}{2}\sum_{i,j=1}^{n} \frac{\partial^2}{\partial x_i \partial x_j}(a_{ij}p) + \ldots - \sum_{i=1}^{n} \frac{\partial}{\partial x_i}(f_i p) \quad \forall\, t > 0, \quad \mathbf{x} \in \mathbb{R}^n. \tag{5.40}$$

In general form, the Fokker-Planck operator is defined as (Soong, 1973):

$$\mathcal{L}_p = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2}{\partial x_i \partial x_j}\left(a_{ij}(\mathbf{x})(\cdot)\right) - \sum_{i=1}^{n} \frac{\partial}{\partial x_i}(f_i(\mathbf{x})(\cdot)) - \frac{\partial}{\partial t}(\cdot). \tag{5.41}$$

$$\mathcal{L}p_{st} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2}{\partial x_i \partial x_j}\left(a_{ij}p_{st}\right) - \sum_{i=1}^{n} \frac{\partial}{\partial x_i}(f_i(\mathbf{x})(p_{st})) = 0 \tag{5.42}$$

with p_{st} is the stationary pdf of a system in (2.4). Let us denote $\mathcal{L}_i$, $i = 1, 2, \ldots, N$ as

the partial differential operators, if (5.42) can be written in the form

$$\mathcal{L}p_{st} = \sum_{i=1}^{n} \mathcal{L}_i \left[\beta_i(x) p_{st}(x) + \gamma_i(x) \frac{\partial p_{st}(x)}{\partial x} \right] = 0 \tag{5.43}$$

then the solution of (5.43) satisfies,

$$\beta_i(x) p_{st}(x) + \gamma_i(x) \frac{\partial p_{st}(x)}{\partial x} = 0 \quad i = 1, 2, \ldots, n. \tag{5.44}$$

The solution of (5.44) is given as:

$$p_{st}(x) = c \prod_{i=1}^{n} \exp\left(- \int_0^{x_i} \frac{\beta_i(s)}{\gamma_i(s)} ds \right). \tag{5.45}$$

The state equation of the bistable system is:

$$dx = \frac{1}{2}(x - x^3)dt + \sqrt{\alpha} dW(t) \tag{5.46}$$

where α is a constant and determines the noise amplitude. The corresponding Fokker-Planck operator is then:

$$\mathcal{L}f_{st} = \frac{\alpha}{2} \frac{\partial^2 p_{st}}{\partial x^2} - \frac{\partial}{\partial x}[0.5(x - x^3)p_{st}]. \tag{5.47}$$

If we rearrange (5.47), we obtain :

$$\frac{\partial}{\partial x} \left(-\frac{1}{2}(x - x^3)p_{st}(x) + \frac{\alpha}{2} \frac{\partial p_{st}(x)}{\partial x} \right) = 0. \tag{5.48}$$

Here, $\mathcal{L}_1 = \dfrac{\partial}{\partial x}$, $\beta_1(x) = -0.5(x - x^3)$ and $\gamma_1(x) = 0.5\alpha$ for the bistable system. If we

replace the coefficients in (5.45), the stationary pdf becomes,

$$p_{st}(x) = c_1 \exp\left(-\frac{1}{\alpha}\left[-\frac{x^2}{2} + \frac{x^4}{4}\right]\right). \tag{5.49}$$

Since $\int p_{st}(x)dx = 1$, we derive another equation such as:

$$c_1 \int_{\infty}^{\infty} \exp\left(-\frac{1}{\alpha}\left[-\frac{x^2}{2} + \frac{x^4}{4}\right]\right) dx - 1 = 0 \tag{5.50}$$

In order to find c_1 coefficient, (5.50) is solved. Then, c_1 becomes,

$$c_1 = \frac{2}{\pi\left(I_{-\frac{1}{4}}\left(\frac{1}{8\alpha}\right) + I_{\frac{1}{4}}\left(\frac{1}{8\alpha}\right)\right)} \exp\left(\frac{-1}{8\alpha}\right). \tag{5.51}$$

By rearranging (5.49), the stationary probability density of the bistable system takes the form (Soong, 1973):

$$p_{st}(x) = \frac{2}{\pi\left(I_{-\frac{1}{4}}\left(\frac{1}{8\alpha}\right) + I_{\frac{1}{4}}\left(\frac{1}{8\alpha}\right)\right)} \exp\left(-\frac{\frac{1}{4} - x^2 + \frac{x^4}{2}}{2\alpha}\right). \tag{5.52}$$

$I_n(x)$ is the modified Bessel function of the first kind. The theoretical value of entropy is computed through the density formula in (5.52) using (2.38).

We have solved the stochastic differential equation in (5.46) numerically, by Euler-Maruyama method on fixed time intervals with $\Delta t = 0.1\ s$ (Kloeden & Platen, 2013). The simulations have been repeated 1000 times for different number of

Figure 5.26 (a) The NB and (b) the NSE of the $\hat{h}(X)$ of bistable system in (5.46) using histogram estimator with respect to N versus ε

observations $2256 \leq N \leq 4048$ by starting with different initial conditions. The first 2000 samples of the simulations have been removed as transients. α in (5.46) has been selected as 1. The entropy of the bistable system has been computed numerically as $h(X) = 1.351$ *nats* according to density function which is defined in (5.52). The entropy estimation has been validated on the system in (5.46) for histogram, KDE, kNN and kpN entropy estimators.

It has been shown that the NB of entropy decreases when the interval of diameter increases up to $\varepsilon \approx 1$ in histogram estimator. The NB of entropy has a minimum for all N and $\varepsilon = 1$ and the NSE of entropy is smaller than 0.01 for all (N, ε) pairs in Figure 5.26. The entropy of the bistable system is estimated via Gaussian KDE, and the NB and NSE are illustrated in Figure 5.27. There are different pairs of $(N, |h|)$ to minimize the NB of entropy as Figure 5.27. Besides, the NSE of entropy is acceptable for all estimations of Gaussian KDE. The Epanechnikov kernel function is used in the estimation of bistable entropy. Figure 5.28 has shown that the NB of entropy is quite low for a wide range of pairs of $(N, |h|)$. NSE of entropy is also smaller than 0.01.

Figure 5.27 (a) The NB and (b) NSE of $\hat{h}(X)$ of the bistable system in (5.46) using KDE with Gaussian kernel base function for varying parameters of $(N,\ |h|)$

Figure 5.28 (a) The NB and NSE of $\hat{h}(X)$ of the bistable system in (5.46) using KDE with Epanechnikov kernel base function for varying parameters of $(N,\ |h|)$

The entropy of bistable is estimated with high accuracy when the bandwidth of the Epanechnikov kernel function is limited.

In Figure 5.29, NB and NSE of the bistable system's entropy have been illustrated using the kNN entropy estimator. Although the NB of entropy is biased, the NSE of entropy has the lowest value in the kNN entropy estimator. The NB and NSE of

Figure 5.29 (a) The NB and NSE of $\hat{h}(X)$ of the bistable system in (5.46) using kNN entropy estimator with respect to N versus k

Figure 5.30 (a) The NB and NSE of $\hat{h}(X)$ of the bistable system in (5.46) using kpN entropy estimator with respect to p_k/N versus k when $N = 2048$

entropy of bistable system are lower than 0.02 in kpN entropy estimator as shown in Figure 5.30. It has been observed that the kpN entropy estimator estimated the entropy of the bistable system with the highest accuracy.

A comparative study has been presented for the expected value of estimations and the NMSEs of entropy in the case of various α in Figure 5.31. In these simulations,

Figure 5.31 (a) The expected value of the entropy estimations by using histogram, Gauss KDE, Epanechnikov KDE, and kpN entropy estimator. The dashed line shows the theoretical value of the entropy. (b) The NMSE error of the entropy estimations by using histogram, Gauss KDE, Epanechnikov KDE, and kpN entropy estimator.

the parameters are selected as in the following: $N = 2048$ for all estimators, $\varepsilon = 1$ in histogram, $|h| = 0.1$ in Gauss KDE, $|h| = 0.2$ in Epanechnikov KDE, $k = 10$ in kNN, $k = 10$, $p_k/N = 0.1$ in kpN estimators. As the noise level increases, the estimators' bias is approximately the same value in Figure 5.31 depicts. If the entropy is estimated via the histogram, the bias of this estimator decreases first and then increases. In addition, the NMSE of the estimators is too small for all levels of the noise. The NMSE of the histogram estimator is slightly more significant than the other estimators. We have seen that the histogram estimator is less robust than the other estimators in Figure 5.31.

In summary, we have confirmed that the performance of the histogram estimator depends on the (N, ε) pair. The bandwidth of kernel is limited in Epanechnikov KDE since the bistable system's support set is also bounded, it has been evident that the estimator's bias is reduced for the entropy of the bistable system. The tails of Gaussian KDE go to infinity. Thus, if the entropy of the bistable system is estimated with Gaussian KDE, it has been expected that the NB of entropy of it is higher than Epanechnikov KDE. We have determined that the kNN entropy estimator estimates the entropy and the entropy-based measures efficiently and with high accuracy, if the density is locally uniform regardless of the dimension of observation. On the other hand, the entropy estimations are biased when the system is nonlinear, and the observations are not distributed in sample space like the bistable system. We have observed that kpN estimator estimates entropy correctly in a nonlinear bistable system.

5.2.3 Temperature electricity consumption dataset

The performance of the entropy estimators has also been tested on a real dataset. In Figure 5.32, the dataset have been demonstrated for outside temperature versus electricity consumption in Turkey. The dataset also includes the causality between temperature and electricity consumption. In other words, it is clear that the variations in the ambient temperature cause the variations in the electricity consumption but not visa versa (Mooij et al., 2016).

Undoubtedly, the theoretical values of the entropy-based measures are unknown in the real dataset. As a well-known method, we have used the hypothesis tests to measure the statistical significance of the TE (Vicente et al., 2011; Lizier et al., 2011). The TEs have been estimated by surrogating the data to measure the performance of the entropy estimators in the literature (Bossomaier et al., 2016). The estimated TE between two time-series measures whether or not two variables are independent.

First, we have taken samples of the electricity consumption E, and the temperature T (Figure 5.32). Then, we have permuted only the time series of temperature denoted

Figure 5.32 A real dataset that shows the causality of Temperature $^{\circ}C \rightarrow$ Electricity Consumption [MW]

as T^* to destroy causality (Lizier et al., 2011) in the generated surrogate data series. The null hypothesis becomes,

$$\mathcal{H}_0 : TE_{T^* \rightarrow E} > TE_{T \rightarrow E}. \tag{5.53}$$

In other words,

$\mathcal{H}_0$: The temperature alteration does not affect electricity consumption.

The null hypothesis should be rejected in any estimation method since we do not know the distribution error of the TE estimations.

Table 5.1 The significance level $p_{\mathcal{H}_0}$ for the null hypothesis $\mathcal{H}_0$ with $N = 1024$, $N = 2048$ and $N = 4096$

The Estimators	**The Parameters**	**N=1024**	**N=2048**	**N=4096**
		$\mathbf{p}_{\mathcal{H}_0}$	$\mathbf{p}_{\mathcal{H}_0}$	$\mathbf{p}_{\mathcal{H}_0}$
Epanechnikov KDE	$\lvert h\rvert = 4.5$	0.493	0.248	0.026
Gaussian KDE	$\lvert h\rvert = 2.9$	0.244	0.041	0.001
Histogram	$\varepsilon = 4$	0.209	0.036	0.002
kpN	$k = 3$, $p_k/N = 0.1$	0.172	0.103	0.028
kNN	$k = 3$	0.037	0.003	0.002

The data T^* has been shuffled to generate 10000 different surrogate series, and then $TE_{T^*\to E}$ has been estimated to test the hypothesis $\mathcal{H}_0$. Let $p_{\mathcal{H}_0}$ be the significance level of $\mathcal{H}_0$. The null hypothesis $\mathcal{H}_0$ is rejected if $p_{\mathcal{H}_0} < \alpha_{\mathcal{H}_0}$. We have assumed that $\alpha_{\mathcal{H}_0} = 0.05$. In Table 5.1, the significance level $p_{\mathcal{H}_0}$ for the null hypothesis $\mathcal{H}_0$ have been illustrated. The number of samples in estimation has varied from 1024 to 4096, and we have tested the null hypothesis for all estimators whose parameters are specifically selected for their best estimation performance.

As the number of observations is increased, we have seen that the null hypothesis is rejected for any type of estimator in Table 5.1. While the histogram and Gauss KDE estimators have rejected the null hypothesis for $N \geq 2048$, the null hypothesis has been rejected for Epanechnikov KDE and the kpN in the case of $N \geq 4096$. The number of observations should be high enough to determine the causality between the systems in Table 5.1 to represent the underlying pdf in the histogram, Gauss KDE, Epanechnikov KDE, and kpN estimators. In the kNN estimator, the null hypothesis has been rejected for all N as shown in Table 5.1. Even though the kNN estimator has a slight bias, which can be canceled in expense of additional computational effort, it can be used effectively to obtain the causality between the systems for this real dataset.

As a result, the kNN entropy estimator has successfully estimated information measures in linear coupled systems, and kpN and Epanechnikov KDE have been efficient in estimating entropy in nonlinear systems. The number of observations has been the most crucial parameter that alters the estimators' performance in linear

systems. The parameters which symbolize the bandwidth, such as ε, $|h|$, and p_k/N do not indicate a specific optimal value for coupled autoregressive processes. However, we select the optimal ε, $|h|$, p_k/N parameters in bistable system. Hence, we can obtain accurate estimations by fewer observations and the optimal bandwidth parameters.

In the real dataset, the success of the entropy estimators has also been dependent on the number of observations. It is also found that when we estimate the TEs of the systems in the real dataset, the estimations by using the kNN entropy estimator have a statistical significance level for the small number of observations. We have obtained the causality for these systems using the kNN entropy estimator.

In summary, we have presented a framework for estimating information-theoretic measures in this chapter. It is seen that the kNN entropy estimator has a bias for a small number of samples; however, it estimates the measures quickly. On the other hand, the KDE or the kpN estimator has had a low bias, but the estimation times have been relatively long.

Clearly, there is a trade-off between the speed and the accuracy of the estimations, as usual. If the accuracy of the estimation is crucial, then the KDE or kpN estimator can be chosen. However, we deal with the chaotic dynamical systems and their information flow in a network. The volume of data is enormous, even in a small network. We are more interested in preserving the causality information effectively rather than estimating the value of the entropy in exact digits. The relative information measures between the systems in a network can quantify the information flow to reduce the bias. Hence, we have used the kNN entropy estimator in our following analysis for fast and accurate enough results. Now, our goal is to reveal paths in a chaotic dynamical network in complete synchronization using the information-theoretic tools represented up to this point.

CHAPTER SIX
UNFOLDING THE STRUCTURE OF CHAOTIC DYNAMICAL NETWORKS VIA CAUSATION ENTROPY

In this chapter, we address the problem of reconstructing the unknown network coupling scheme using the observations gathered from only a subset of individual systems in a network.

There are several approaches to the solution of the problem. In one approach, the response of an auxiliary network is adjusted via adaptive controllers to match the response of the network in question (Yu et al., 2006; Yu & Parlitz, 2011; Wu et al., 2016; Liu et al., 2022). The study in (Chen et al., 2009) addresses the intricate problem of synchronization in the network. Similar adaptive controller-based studies are widely used in network reconstruction, they are based on the assumption that the internal dynamics of each node and the coupling function are known a priori, and the system under study is observable usually (Yu et al., 2006).

Machine learning methods are also employed in solving network reconstruction problems. It has been demonstrated that the reconstruction of phase-coupled Kuramoto networks is possible using artificial neural networks. The transient dynamics of coupled oscillators by perturbing the systems in the network are used for obtaining information about the network structure in synchronization (Panaggio et al., 2019). Leguia et al. have proposed a network reconstruction approach by using the trajectories as features and then applying two independent feature ranking methods to rank the importance of each node for predicting the states of other nodes in the network to obtain the adjacency matrix. The proposed recipe has been reported to perform well with discrete-time maps and with respect to varying coupling strength, noise, trajectory length, and network size (Leguia et al., 2019).

As an effective alternative, the network topology can be inferred via information-theoretic measures. The existing information-theoretic measures are used in solving model-free problems frequently (Novelli & Lizier, 2021; Lizier et al., 2011; Stetter et al., 2012; Porfiri & Marín, 2017; Sun et al., 2015). One of such

measures is the TE suggested by Schreiber (Schreiber, 2001). The TE measures information flow from one system to a related system by observing the past values of the first system. It has been widely used in the reconstruction of networks in various fields such as neuroscience (Vicente et al., 2011), climate networks (Hlinka et al., 2013) and policy diffusion (Porfiri & Marín, 2017).

An alternative and closely related measure, the CE, allows us to measure information flow from one system to another by considering a third related system (Sun & Bollt, 2014). Sun et al. have proposed the optimal causation entropy principle based on the CE. According to this principle, the causal parents of a system in the network are the minimal set of nodes that maximizes the CE. The algorithm based on the optimal causation entropy principle has been shown to be capable of determining the causal parents of a given node in the network efficiently (Sun et al., 2015). Furthermore, the authors have illustrated that the algorithm based on the optimal causation entropy principle outperforms the methods based on the TE and Granger causality (Granger, 1988) in the coupled Gaussian processes networks. An important consequence is that the density of links and information diffusion rate are more important than the number of nodes when inferring the topology (Sun et al., 2015).

In recent years, Novelli and Lizier have compared the properties of the networks from microscopic to macroscopic scales by using MI, bivariate TE, and multivariate TE measures (Novelli & Lizier, 2021). In (Novelli & Lizier, 2021), the authors are not only seeking to infer the network topology via information-theoretical tools on the microscopic scale but also analyzing the structural properties of the network, such as the characteristic path length, clustering coefficient, small-worldness coefficient, degree distribution, and modularity in macroscopic scales. The authors have found that the multivariate TE outperforms MI and bivariate TE when identifying the structural properties of the network in large time series. Similarly, a simple test that infers the causal links has been introduced recently. The proposed test is based on the difference between two information-theoretic measures estimated directly from observations. The results have been validated on synthetic and real-world-time series with known causal relationships. Furthermore, the causal connections between

nonlinear systems could be recovered where the TE measure fails (Haaga et al., y 28).

Although the studies that infer the network structure in coupled discrete-time maps are relatively abundant, the work on reconstructing the network topology in a continuous-time synchronized network is rather thin. In this section, we address the problem of inferring network topology by only observing the systems' output when synchronizing the continuous-time network. We show that the difference of the course-grained average integrated causation entropies of the coupled systems can be used as a discriminative feature when reconstructing the topology in a directed network that exhibits complete synchronization. We assume that random information can be injected into the network via available inputs of the individual coupled systems, and a single output is accessible for each node. However, the individual dynamics of the nodes are not available directly. Nevertheless, the results indicate that the evolution of the injected entropy through the network can reveal the network structure in this model-free approach.

6.1 Network Reconstruction via Causation Entropy

Consider the network $\mathcal{G}$, as in definition 2.1.1. We assume that the state equations of the network is defined as (2.6) or (2.7) and the network is completely synchronized according to definition 2.4.1.

The outputs $\mathbf{y}_i(t)$ of (2.6) can be sampled in an optimal way to convert the system to a discrete one as in (2.7) so that the information-theoretic properties are preserved. We know that the differential entropy in (2.38) is not preserved under the diffeomorphisms in general according to Theorem 2.5.10. The TE and the CE are preserved according to Theorems 4.1.5 and 4.1.7, respectively. Hence, their estimations provided by observations in reconstructed phase space result in the same TE and the CE values as the estimation obtained by full state observations up to the numerical error.

We begin with reconstructing the outputs of $\mathbf{y}_i(t)$ by using the state space reconstruction techniques in Section 2.3. Then, TE or CE is estimated in the

reconstructed space. We claim that the adjacency matrix of $\mathbf{A}$ of a network in complete synchronization can be estimated by using the estimated outputs $\tilde{y}_{ir,n}$ via information theoretic measures. Note that no prior knowledge of the underlying dynamics (i.e., $\mathbf{f}$, $\mathbf{g}$ and ϕ_i) or coupling strength ϵ is assumed since we can estimate the required parameters from the observed outputs only.

Let $i,\ j,\ k \subset \mathcal{V}$ be any three network nodes that are selected randomly. To provide useful information during the synchronization, we can perturb the inputs to the systems at specific times. We redefine CE in (4.23) as:

$$CE_{\underline{i}\to j\mid k}(n,u_i) = h(\tilde{Y}_{j,n+1} \mid \tilde{\mathbf{Y}}^{(d)}_{k,n}, \boldsymbol{U}_i) - h(\tilde{Y}_{j,n+1} \mid \tilde{\mathbf{Y}}^{(d)}_{k,n}, \tilde{\mathbf{Y}}^{(d)}_{i,n}, \boldsymbol{U}_i) \tag{6.1}$$

A possible choice for input u_i is then,

$$u_i(n) = \begin{cases} \boldsymbol{\alpha}_i\delta(n) & \text{mod } (L,n) = 0,\ L \in \mathbb{Z}^+ \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

where $\delta(\cdot)$ is the Dirac delta function, and $\boldsymbol{\alpha}_i$ is uniformly distributed over $[-a,a]^{d_i}$, $a > 0$. The duration L between applying two inputs is chosen long enough to reach the synchronization manifold after the perturbation. The underlined symbol in (6.1), i.e, $\underline{i}$, indicates that the node i is perturbed.

Let us define the *integrated causation entropy* (ICE) to quantify total information flow in an interval (Canlı & Günel, 2020):

$$ICE_{\underline{i}\to j\mid k}(u_i) = E_n[CE_{\underline{i}\to j\mid k}(n,u_i)]. \tag{6.3}$$

Related to this measure, the *average integrated causation entropy* (AICE) measures the average total information flow over the random perturbations:

$$AICE_{\underline{i}\to j\mid k} = E_{u_i}\left[ICE_{\underline{i}\to j\mid k}(u_i)\right]. \tag{6.4}$$

The perturbations disturb the synchronization and excite the network by injecting new information into the system. As the information flows through the network en route to the re-synchronization, the connection information is revealed since the directly connected nodes contribute to the AICE less than the non-directed nodes. We propose that the difference of AICEs between the reconstructed outputs can be used to reveal the paths and, therefore, can be used to reconstruct the matrix $\mathbf{A}$.

Assume that the network $\mathcal{G}$ synchronizes asymptotically. Let us label the outputs randomly and choose three systems i, j, and k. Let $\tilde{y}_j[n+1] = \tilde{y}_{j,n+1}$, $\tilde{\boldsymbol{y}}_i^{(d)}[n] = \tilde{\boldsymbol{y}}_{i,n}^{(d)}$ and $\tilde{\boldsymbol{y}}_k^{(d)}[n] = \tilde{\boldsymbol{y}}_{k,n}^{(d)}$.

The following theorems hold given that the systems in $\mathcal{G}$ are identical.

Theorem 6.1.1. *Assume that the system i is perturbed with the input* (6.2).

i) If i is a source node for which $k_i^{out} = 1$ and $k_i^{in} = 0$ (Figure 6.1a), then

$$|AICE_{\underline{i}\to j \mid k} - AICE_{\underline{i}\to k \mid j}| > 0. \tag{6.5}$$

ii) If i is neither a source or nor a sink node where $k_i^{out} \geq 1$ and $k_i^{in} \geq 1$ (Figure 6.1b), then

$$|AICE_{\underline{i}\to j \mid k} - AICE_{\underline{i}\to k \mid j}| > 0. \tag{6.6}$$

iii) If i is a source node whose out-degree at least two $k_i^{out} \geq 2$ and in-degree is zero $k_i^{in} = 0$, certainly (Figure 6.1c), then

$$|AICE_{\underline{i}\to j \mid k} - AICE_{\underline{i}\to k \mid j}| = 0. \tag{6.7}$$

iv) If i is a sink node whose $k_i^{out} = 0$ and $k_i^{in} \geq 1$ (Figure 6.1d and 6.1e), then

$$|AICE_{\underline{i}\to j \mid k} - AICE_{\underline{i}\to k \mid j}| = 0. \tag{6.8}$$

Proof: The systems reach the synchronization manifold asymptotically after the perturbation caused by (6.2).

i) Let us assume that i drives j, and j drives k. Based upon the topology, there exists a homeomorphism between $\mathbf{x}_{k,n+1}$ and $\mathbf{x}_{j,n}$ such that $\mathbf{x}_{k,n+1} = \Phi_{kj}(\mathbf{x}_{j,n})$ (Arnold, 1992). Thus, $h(\tilde{Y}_{k,n+1} \mid \tilde{\boldsymbol{Y}}_{j,n}^{(d)}, \boldsymbol{U}_i) = h(\Phi_{kj}(\tilde{\boldsymbol{Y}}_{j,n}^{(d)}) \mid \tilde{\boldsymbol{Y}}_{j,n}^{(d)}, \boldsymbol{U}_i) = 0$ and $h(\tilde{Y}_{k,n+1} \mid \tilde{\boldsymbol{Y}}_{j,n}^{(d)}, \tilde{\boldsymbol{Y}}_{i,n}^{(d)}, \boldsymbol{U}_i) = h(\Phi_{kj}(\tilde{\boldsymbol{Y}}_{j,n}^{(d)}) \mid \tilde{\boldsymbol{Y}}_{j,n}^{(d)}, \tilde{\boldsymbol{Y}}_{i,n}^{(d)}, \boldsymbol{U}_i) = 0$. By using (6.1), (6.3) and (6.4),

$$\begin{aligned}
&\left|AICE_{\underline{i}\to j \mid k} - AICE_{\underline{i}\to k \mid j}\right| \\
&= \Bigg|E_{u_i}\Bigg[E_n\Bigg[h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{k,n}^{(d)}, \boldsymbol{U}_i) \; - h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{k,n}^{(d)}, \tilde{\boldsymbol{Y}}_{i,n}^{(d)}, \boldsymbol{U}_i) \ldots \\
&\quad - h(\Phi_{kj}(\tilde{\boldsymbol{Y}}_{j,n}^{(d)}) \mid \tilde{\boldsymbol{Y}}_{j,n}^{(d)}, \boldsymbol{U}_i) \; + h(\Phi_{kj}(\tilde{\boldsymbol{Y}}_{j,n}^{(d)}) \mid \tilde{\boldsymbol{Y}}_{j,n}^{(d)}, \tilde{\boldsymbol{Y}}_{i,n}^{(d)}, \boldsymbol{U}_i)\Bigg]\Bigg]\Bigg| \\
&= \Bigg|E_{u_i}\Bigg[E_n\Bigg[h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{k,n}^{(d)}, \boldsymbol{U}_i) - h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{k,n}^{(d)}, \tilde{\boldsymbol{Y}}_{i,n}^{(d)}, \boldsymbol{U}_i)\Bigg]\Bigg]\Bigg| > 0.
\end{aligned} \tag{6.9}$$

ii) Proof is similar to i) Let us assume that i drives j but i is driven by k. Considering the topology, there exists a homeomorphism between $\mathbf{x}_{j,n+1}$ and $\mathbf{x}_{k,n}$ such that $\mathbf{x}_{j,n+1} = \Phi_{jk}(\mathbf{x}_{k,n})$. Consequently, $h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{k,n}^{(d)}, \boldsymbol{U}_i) = h(\Phi_{jk}(\tilde{\boldsymbol{Y}}_{k}^{(d)}) \mid \tilde{\boldsymbol{Y}}_{k,n}^{(d)}, \boldsymbol{U}_i) = 0$ and $h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{k,n}^{(d)}, \tilde{\boldsymbol{Y}}_{i,n}^{(d)}, \boldsymbol{U}_i) = h(\Phi_{jk}(\tilde{\boldsymbol{Y}}_{k}^{(d)}) \mid \tilde{\boldsymbol{Y}}_{k,n}^{(d)}, \tilde{\boldsymbol{Y}}_{i,n}^{(d)}, \boldsymbol{U}_i) = 0$. By using (6.2) and (6.3),

$$
\begin{aligned}
&\left|AICE_{\underline{i}\to j\mid k} - AICE_{\underline{i}\to k\mid j}\right| \\
&= \left|E_{u_i}\left[E_n\left[h(\Phi_{jk}(\tilde{\boldsymbol{Y}}^{(d)}_{k,n}) \mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n}, \boldsymbol{U}_i) \;-h(\Phi_{jk}(\tilde{\boldsymbol{Y}}^{(d)}_{k,n}) \mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n}, \tilde{\boldsymbol{Y}}^{(d)}_{i,n}, \boldsymbol{U}_i)\ldots\right.\right.\\
&\quad \left.\left. - h(\tilde{Y}_{k,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{j,n}, \boldsymbol{U}_i) \;+h(\tilde{Y}_{k,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{j,n}, \tilde{\boldsymbol{Y}}^{(d)}_{i,n}, \boldsymbol{U}_i)\right]\right]\right| \\
&= \left|E_{u_i}\left[E_n\left[-h(\tilde{Y}_{k,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{j,n}, \boldsymbol{U}_i) + h(\tilde{Y}_{k,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{j,n}, \tilde{\boldsymbol{Y}}^{(d)}_{i,n}, \boldsymbol{U}_i)\right]\right]\right| > 0.
\end{aligned}
\tag{6.10}
$$

iii) Before the perturbations $\mathbf{x}_j = \mathbf{x}_k$. The presence of the input affects both systems equally and $\lim_{t\to\infty} \|\mathbf{x}_j(t) - \mathbf{x}_k(t)\| = 0$. Consequently, $h(\tilde{\boldsymbol{Y}}^{(d)}_{j,n} \mid \boldsymbol{U}_i) = h(\tilde{\boldsymbol{Y}}^{(d)}_{k,n} \mid \boldsymbol{U}_i)$, $h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n}, \boldsymbol{U}_i) = h(\tilde{Y}_{k,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{j,n}, \boldsymbol{U}_i)$. Besides, due to the homeomorphism between $\mathbf{x}_{j,n+1}$ and $\mathbf{x}_{i,n}$, for some Φ_{ji} we have $\mathbf{x}_{j,n+1} = \Phi_{ji}(\mathbf{x}_{i,n})$ and also, the homeomorphism between $\mathbf{x}_{k,n+1}$ and $\mathbf{x}_{i,n}$ yields $\mathbf{x}_{k,n+1} = \Phi_{ki}(\mathbf{x}_{i,n})$. Then by (6.3), (6.4), we can write:

$$
\begin{aligned}
&\left|AICE_{\underline{i}\to j\mid k} - AICE_{\underline{i}\to k\mid j}\right| \\
&= \left|E_{u_i}\left[E_n\left[h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n}, \boldsymbol{U}_i) \;- h(\Phi_{ji}(\tilde{\boldsymbol{Y}}^{(d)}_{i,n}) \mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n}, \tilde{\boldsymbol{Y}}^{(d)}_{i,n}, \boldsymbol{U}_i)\ldots\right.\right.\\
&\quad \left.\left. - h(\tilde{Y}_{k,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{j,n}, \boldsymbol{U}_i) + \;h(\Phi_{ki}(\tilde{\boldsymbol{Y}}^{(d)}_{i,n}) \mid \tilde{\boldsymbol{Y}}^{(d)}_{j,n}, \tilde{\boldsymbol{Y}}^{(d)}_{i,n}, \boldsymbol{U}_i)\right]\right]\right| \\
&= \left|E_{u_i}\left[E_n\left[h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n}, \boldsymbol{U}_i) - h(\tilde{Y}_{k,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{j,n}, \boldsymbol{U}_i)\right]\right]\right| \\
&= \left|E_{u_i}\left[E_n\left[h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n}, \boldsymbol{U}_i) - h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n}, \boldsymbol{U}_i)\right]\right]\right| = 0.
\end{aligned}
\tag{6.11}
$$

iv) Suppose that there is a path from j to i and the existence of a path between j and k is not clear. In this case, i and j are not independent. In case of the existence of a path between i and k, i and k are not independent, too. In either case, adding input to i does not destroy the synchronization of j and k; therefore, their conditional entropies are equal, $h(\tilde{\boldsymbol{Y}}^{(d)}_{j,n} \mid \boldsymbol{U}_i) = h(\tilde{\boldsymbol{Y}}^{(d)}_{k,n} \mid \boldsymbol{U}_i)$, $h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n}, \boldsymbol{U}_i) = h(\tilde{Y}_{k,n+1} \mid \tilde{\boldsymbol{Y}}^{(d)}_{j,n}, \boldsymbol{U}_i)$. Then,

Figure 6.1 The possible coupling schemes between the nodes i, j, and k. The solid line with the arrow illustrates that the link exists and must be in the indicated direction. The dashed line indicates that the existence of link is not clear yet. The dashed line with the arrow represents the existence of a link is not clear, but it must be in the arrow's direction if it exists.

$$\begin{aligned}
&\left|AICE_{\underline{i}\to j\mid k} - AICE_{\underline{i}\to k\mid j}\right| \\
&= \left|E_{u_i}\left[E_n\left[h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n}, \boldsymbol{U}_i) \;-h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}^{d}_{k,n}, \tilde{\boldsymbol{Y}}^{(d)}_{i,n}, \boldsymbol{U}_i)\ldots\right.\right.\right. \\
&\quad \left.\left.\left. -h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n}, \boldsymbol{U}_i) \;+h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n}, \tilde{\boldsymbol{Y}}^{(d)}_{i,n}, \boldsymbol{U}_i)\right]\right]\right| = 0.
\end{aligned} \tag{6.12}$$

Theorem 6.1.2. *Assume that the node i is perturbed.*

i) If i is a source node with out-degree at least two $k_i^{out} \geq 2$ and in-degree is certainly zero $k_i^{in} = 0$ (Figure 6.1c), then

$$|AICE_{\underline{i}\to j\mid k} - AICE_{k\to j\mid \underline{i}}| = 0. \tag{6.13}$$

ii) If i is a sink node with $k_i^{out} = 0$ and $k_i^{in} = 1$ (Figure 6.1d), then

$$|AICE_{\underline{i}\to j\mid k} - AICE_{k\to j\mid \underline{i}}| > 0. \tag{6.14}$$

iii) If i is a sink node with $k_i^{out} = 0$ and $k_i^{in} > 1$ (Figure 6.1e), then

$$|AICE_{\underline{i}\to j \mid k} - AICE_{k\to j \mid \underline{i}}| = 0. \tag{6.15}$$

Proof: By definition,

$$\begin{aligned} &\left|AICE_{\underline{i}\to j \mid k} - AICE_{k\to j \mid \underline{i}}\right| \\ &= \left|E_{u_i}\left[E_n\left[h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{k,n}^{(d)}, \boldsymbol{U}_i) - h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{i,n}^{(d)}, \boldsymbol{U}_i)\right]\right]\right| \end{aligned} \tag{6.16}$$

i) Similar to proof of Theorem 6.1.1 case iii) and due to the homeomorphism between $\mathbf{x}_{j,n+1}$ and $\mathbf{x}_{i,n}$, for some Φ_{ji} we have $\mathbf{x}_{j,n+1} = \Phi_{ji}(\mathbf{x}_{i,n})$. Thus,

$$\begin{aligned} &\left|E_{u_i}\left[E_n\left[h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{j,n}^{(d)}, \boldsymbol{U}_i) - h(\Phi_{ji}(\tilde{\boldsymbol{Y}}_{i,n}^{(d)}) \mid \tilde{\boldsymbol{Y}}_{i,n}^{(d)}, \boldsymbol{U}_i)\right]\right]\right| \\ &= \left|E_{u_i}\left[E_n\left[h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{j,n}^{(d)}, \boldsymbol{U}_i)\right]\right]\right| = 0. \end{aligned} \tag{6.17}$$

ii) Using similar facts in proof of Theorem 6.1.1 case iv), we can write $h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{k,n}^{(d)}, \boldsymbol{U}_i) = h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{j,n}^{(d)}, \boldsymbol{U}_i)$ and $h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{j,n}^{(d)}, \boldsymbol{U}_i) \neq h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{i,n}^{(d)}, \boldsymbol{U}_i)$. Hence, $|AICE_{\underline{i}\to j \mid k} - AICE_{k\to j \mid \underline{i}}| > 0$.

iii) Suppose at least two paths exist from j to i and k to i. Adding input to the i does not affect the future states of j. Hence, $h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{k,n}^{(d)}, \boldsymbol{U}_i) = h(\tilde{Y}_{j,n+1} \mid \tilde{\boldsymbol{Y}}_{i,n}^{(d)}, \boldsymbol{U}_i)$. The difference of AICEs becomes zero.

Theorem 6.1.3. *Assume that the node i is perturbed.*

i) If i is a source node whose $k_i^{out} = 1$ and $k_i^{in} = 0$ (Figure 6.1a), and there exist a path from j to k or k to j, then

$$\begin{aligned} |AICE_{\underline{i}\to j \mid k} - AICE_{k\to j \mid \underline{i}}| > 0 \\ |AICE_{\underline{i}\to k \mid j} - AICE_{j\to k \mid \underline{i}}| > 0. \end{aligned} \tag{6.18}$$

ii) If i is neither a source node or nor a sink node whose $k_i^{out} \geq 1$ and $k_i^{in} \geq 0$ (Figure 6.1f), and there exist no direct path from j to k or k to j, then

$$\begin{aligned} |AICE_{\underline{i}\to j \mid k} - AICE_{k\to j \mid \underline{i}}| > 0 \\ |AICE_{\underline{i}\to k \mid j} - AICE_{j\to k \mid \underline{i}}| > 0. \end{aligned} \tag{6.19}$$

iii) If i is neither a source node or nor a sink node whose $k_i^{out} \geq 1$ and $k_i^{in} \geq 1$, and there exists a direct path from j to k or k to j (Figure 6.1g and 6.1h), then

$$\begin{aligned} |AICE_{\underline{i}\to j \mid k} - AICE_{k\to j \mid \underline{i}}| = 0 \\ |AICE_{\underline{i}\to k \mid j} - AICE_{j\to k \mid \underline{i}}| = 0. \end{aligned} \tag{6.20}$$

iv) If i is a source node whose $k_i^{out} = 1$ and k_i^{in} is not clear (Figure 6.1i), then

$$\begin{aligned} |AICE_{\underline{i}\to j \mid k} - AICE_{k\to j \mid \underline{i}}| > 0 \\ |AICE_{\underline{i}\to k \mid j} - AICE_{j\to k \mid \underline{i}}| = 0. \end{aligned} \tag{6.21}$$

Proof: By definition,

$$\begin{aligned}&\left|AICE_{\underline{i}\to j\mid k}-AICE_{k\to j\mid \underline{i}}\right|\\&=\left|E_{u_i}\left[E_n\left[h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n},\boldsymbol{U}_i)-h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}^{(d)}_{i,n},\boldsymbol{U}_i)\right]\right]\right|\end{aligned}\tag{6.22}$$

and

$$\begin{aligned}&\left|AICE_{\underline{i}\to k\mid j}-AICE_{j\to k\mid \underline{i}}\right|\\&=\left|E_{u_i}\left[E_n\left[h(\tilde{Y}_{k,n+1}\mid \tilde{\boldsymbol{Y}}^{(d)}_{j,n},\boldsymbol{U}_i)-h(\tilde{Y}_{k,n+1}\mid \tilde{\boldsymbol{Y}}^{(d)}_{i,n},\boldsymbol{U}_i)\right]\right]\right|.\end{aligned}\tag{6.23}$$

i) If i drives j, j drives k, and there is no link between i and k, with the similar line of reasoning, $\left|AICE_{\underline{i}\to j\mid k}-AICE_{k\to j\mid \underline{i}}\right|=\left|E_{u_i}\left[E_n\left[h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n},\boldsymbol{U}_i)\right]\right]\right|>0$ and $\left|AICE_{\underline{i}\to k\mid j}-AICE_{j\to k\mid \underline{i}}\right|=\left|E_{u_i}\left[E_n\left[-h(\tilde{Y}_{k,n+1}\mid \tilde{\boldsymbol{Y}}^{(d)}_{i,n},\boldsymbol{U}_i)\right]\right]\right|>0$.

ii) If i drives j but i is driven by k, and there is no link between i and k, using the homeomorphism between i and j, i.e., $\mathbf{x}_{j,n+1}=\Phi_{ji}(\mathbf{x}_{i,n})$, (6.22) becomes $\left|AICE_{\underline{i}\to j\mid k}-AICE_{k\to j\mid \underline{i}}\right|=\left|E_{u_i}\left[E_n\left[h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n},\boldsymbol{U}_i)\right]\right]\right|>0$. Also, by using homeomorphism between i and k, i.e., $\mathbf{x}_{k,n+1}=\Phi_{ki}(\mathbf{x}_{i,n})$, then (6.23) is equivalent to $\left|E_{u_i}\left[E_n\left[h(\tilde{Y}_{k,n+1}\mid \tilde{\boldsymbol{Y}}^{(d)}_{j,n},\boldsymbol{U}_i)\right]\right]\right|>0$.

iii) Consider i drives j, i is driven by k, and j drives k. It represents a cyclic scheme, and all nodes influence each other. In this case, there exists homeomorphisms between $\mathbf{x}_{j,n+1}$ and $\mathbf{x}_{k,n}$ for some Φ_{jk} and between $\mathbf{x}_{j,n+1}$ and $\mathbf{x}_{i,n}$ for some Φ_{ji}. Thus,

$$\begin{aligned}&\left|AICE_{\underline{i}\to j\mid k}-AICE_{k\to j\mid \underline{i}}\right|\\&=\left|E_{u_i}\left[E_n\left[h(\Phi_{jk}(\tilde{\boldsymbol{Y}}^{(d)}_{k,n})\mid \tilde{\boldsymbol{Y}}^{(d)}_{k,n},\boldsymbol{U}_i)-h(\Phi_{ji}(\tilde{\boldsymbol{Y}}^{(d)}_{i,n})\mid \tilde{\boldsymbol{Y}}^{(d)}_{i,n},\boldsymbol{U}_i)\right]\right]\right|=0.\end{aligned}\tag{6.24}$$

Similarly, the homeomorphisms between k and j, k and i yield $|AICE_{\underline{i}\to k\mid j}-AICE_{j\to k\mid \underline{i}}|=0$. A possible scheme is i drives j, i is driven by k, and k drives j. By using the homeomorphisms between the states, $|AICE_{\underline{i}\to j\mid k}-AICE_{k\to j\mid \underline{i}}|=0$. Similar to theorem 6.1.1, case iv), adding input to i does not affect the future states of k. Hence, $|AICE_{\underline{i}\to k\mid j}-AICE_{j\to k\mid \underline{i}}|=0$.

iv) Consider the case where a path exists from i to j; there is no path between j and k. Besides, assume that the existence of a path between i and k is unknown, but if it exists, the path must be from k to i which implies, $|AICE_{\underline{i}\to j \mid k} - AICE_{k\to j \mid \underline{i}}| > 0$. j and k are independent and adding pulse to i does not affect k system's states, therefore, $|AICE_{\underline{i}\to k \mid j} - AICE_{j\to k \mid \underline{i}}| = 0$.

There are many possibilities to infer information flow, as shown in Figure 6.1. The difference of $|AICE_{\underline{i}\to j \mid k} - AICE_{\underline{i}\to k \mid j}|$ distinguishes cases in Figures 6.1a and 6.1b from Figures 6.1c, 6.1d and 6.1e by using Theorem 6.1.1. Thus, we can remove the paths by checking this condition. However, Theorem 6.1.1 is not sufficient alone to classify all cases in Figure 6.1. As a second criterion, we check the difference of $|AICE_{\underline{i}\to j \mid k} - AICE_{k\to j \mid \underline{i}}|$, if (6.7) or (6.8) is satisfied. Theorem 6.1.2 separates cases in Figures 6.1c, and 6.1e from case in Figure 6.1d. If (6.5) or (6.6) is satisfied, two additional conditions are checked ($|AICE_{\underline{i}\to j \mid k} - AICE_{k\to j \mid \underline{i}}|$, $|AICE_{\underline{i}\to k \mid j} - AICE_{j\to k \mid \underline{i}}|$). Theorem 6.1.3 is used for deciding either the case in Figure 6.1a and 6.1f or 6.1g and 6.1h, or 6.1i is the correct scheme. Table 6.1 summarizes the difference of $AICEs$ in Figure 6.1.

6.2 Network Reconstruction Algorithm

Algorithm 1 summarizes how Theorems 6.1.1, 6.1.2 and 6.1.3 can be used to obtain the adjacency matrix.

We start with a coupling matrix corresponding to a fully-connected network and assume coupling in both directions between two nodes is equiprobable. We can detect the paths and then remove contrariwise paths in the algorithm since we can represent the information flow through paths in the view of Theorems 6.1.1, 6.1.2, and 6.1.3. In the end, non-existing paths are removed, and the remaining paths reconstruct the topology.

The estimation of $AICEs$ in Algorithm 1 requires ICE values in (6.3) that is based on the estimation of CEs defined in (6.1) in an interval. We can use the kNN entropy

Table 6.1 A summary of the difference of $AICEs$ for the networks in Figure 6.1

	$\lvert AICE_{\underline{i}\to j\mid k} - AICE_{\underline{i}\to k\mid j}\rvert$	$\lvert AICE_{\underline{i}\to j\mid k} - AICE_{k\to j\mid \underline{i}}\rvert$	$\lvert AICE_{\underline{i}\to k\mid j} - AICE_{j\to k\mid \underline{i}}\rvert$
	> 0	> 0	> 0
	> 0	inconclusive	inconclusive
	0	0	0
	0	> 0	> 0
	0	0	0
	> 0	> 0	> 0
	> 0	0	0
	> 0	0	0
	> 0	> 0	0

estimator in (5.26) to estimate the required CEs.

Since our numerical results have finite precision, the conditions (6.5) - (6.21) have to be checked against a threshold $0 < \epsilon_0 \ll 1$ numerically, so that e.g. (6.6) assumed to hold when $|AICE_{\underline{i}\to j \mid k} - AICE_{\underline{i}\to k \mid j}| < \epsilon_0$. Here, we have chosen i, j, and k randomly from the network. We have used the hypothesis tests to measure the statistical significance to determine ϵ_0 (Good, 2005). We have surrogated i^{th} output denoted as $i^\star$ by shuffling the observation points of i in time, keeping the remaining outputs unchanged. Here, we have considered the null hypothesis that $\mathcal{H}_0 : |AICE_{i^\star\to j \mid k} - AICE_{i^\star\to k \mid j}| = 0$. To test the null hypothesis, the empirical cumulative distribution function of $F_{|AICE_{i^\star\to j \mid k} - AICE_{i^\star\to k \mid j}|}(\epsilon_0)$ has been obtained. Then, the observation of $|AICE_{i^\star\to j \mid k} - AICE_{i^\star\to k \mid j}| > \epsilon_0$ has become significant when $\hat{F}_{|AICE_{i^\star\to j \mid k} - AICE_{i^\star\to k \mid j}|}(\epsilon_0) \leq \theta$ where θ is significance threshold. Otherwise, the null hypothesis has been rejected at $(1 - \theta)$. ϵ_0 has been determined only once at initialization step 3 of Algorithm 1.

We have utilized the *NetworkX* library to find paths, remove direct paths and eliminate the irrelevant candidate networks that are not connected graphs or that are isomorphisms (see definition 2.1.6) when updating the coupling matrix in the Algorithm 1 (Hagberg et al., t 02,l). Figure 6.2 and 6.3 are the subroutines called from Algorithm 1. Also, has_path in Figure 6.3 is a NetworkX function that returns True if a graph has a path from source to target, False otherwise.

We have used the kd-tree searching algorithm in (5.26) to determine the number of kNNs inside the ball for which the computational complexity is $O(M \log M)$ (Bentley, 1975). The complexity of $AICE_{i\to j|k}$ depends on the number random perturbations (K) in (6.4), L in (6.2) and the overlapping ratio that are needed in the estimation of CE. Let m be the number of the estimated CE in (6.3). Then, the complexity of estimating CE's becomes $O(K \times m \times M \times \log M)$. In the worst case, we have $2N(N-1)(N-2)$ triples of (i, j, k) in the estimation of AICEs, when the perturbed system is either i or k. Therefore, in the worst case, the complete procedure yields an algorithm of $O(N^3 \times K \times m \times M \times \log M)$. However, the expected complexity is much better on

Algorithm 1 Network Reconstruction Algorithm

Input: $\mathcal{Y}$: $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$: observed outputs,
L : the duration between two perturbations,
$\mathcal{I} = \{(i, j, k) \mid i, j, k \subset \mathcal{V} \text{ and } j \neq i, k \neq i, j \neq k\}$: index set of triples,
K : # perturbations,

Output: $\mathcal{A}$: A list of possible coupling matrices

Initialize: $\mathcal{A} = \{\mathbf{1}_{N\times N} - \mathbf{I}_{N\times N}\}$ % $\mathbf{1}_{N\times N}$ is the unit matrix, $\mathbf{I}_{N\times N}$ is the identity matrix.

```
procedure NETWORK_RECONSTRUCTION(I, Y, L, K)
    Obtain τ and d_e of systems by using y_i's, and reconstruct the state space and
    observe ỹ_i,
    Determine ε_0
    do
        Select (i, j, k) randomly from I
        successors = { }
        for each A in A do
            Append(successors,UPDATE_NETWORK_PATHS(A))
            if A == successors then
                return A
        A ← successors
    while length(A) ≥ 1 and I ≠ ∅
    return A
```

average since the calculation of AICEs is not needed for all triples of nodes in most cases.

6.3 Numerical Results

To validate the performance of the proposed algorithm, first, we have tested it on small networks that consist on 4 systems with various coupling schemes as depicted in Figure 6.4. Each circle represents a Chua system in Figure 6.4 and described by (2.11). The simulation parameters are shown in Table 6.2.

Additionally, the coupling function has been chosen as $\mathbf{h}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i - \mathbf{x}_j$ in (2.5). We have solved the differential equations numerically. The sampling time has been chosen as $t_s = 0.01$. The simulations have been performed in parallel in Python. We have estimated the CEs in every 8000 samples with an overlapping ratio of 90% of the frame length. The simulations have been repeated for $K = 1000$ realizations, and then the AICEs of subsystems have been estimated.

function UPDATE_NETWORK_PATHS(**M**) % Takes a connectivity matrix **M** and returns a list of possible connectivity matrices
% Create an empty list that will carry possible candidate networks
candidates = { }
Estimate $AICE_{\underline{i}\to j \mid k}$ and $AICE_{\underline{i}\to k \mid j}$ % Estimate AICEs using (6.4) K times
if $|AICE_{\underline{i}\to j \mid k} - AICE_{\underline{i}\to k \mid j}| > \epsilon_0$ **then**
Estimate $AICE_{k\to j \mid \underline{i}}$ and $AICE_{j\to k \mid \underline{i}}$
if $|AICE_{\underline{i}\to j|k} - AICE_{k\to j|\underline{i}}| > \epsilon_0$ and $|AICE_{\underline{i}\to k|j} - AICE_{j\to k|\underline{i}}| > \epsilon_0$ **then**
Append(candidates, CHECK_SYMMETRY_RMV(**M**, i, j, k))
Append(candidates, CHECK_SYMMETRY_RMV(**M**, i, k, j))
Append(candidates, CHECK_SYMMETRY_RMV(**M**, k, i, j))
Append(candidates, CHECK_SYMMETRY_RMV(**M**, j, i, k))
else if $|AICE_{\underline{i}\to j \mid k} - AICE_{k\to j \mid \underline{i}}| > \epsilon_0$ and $|AICE_{\underline{i}\to k \mid j} - AICE_{j\to k \mid \underline{i}}| \leq \epsilon_0$ **then** Append(candidates, CHECK_SYMMETRY_RMV(**M**, i, j, ∅))
Append(candidates,CHECK_SYMMETRY_RMV(**M**, i, k, ∅))
else
Append(candidates, CHECK_SOURCE_SINK(**M**, k, j, i))
Append(candidates, CHECK_SOURCE_SINK(**M**, j, i, k))
else
Estimate $AICE_{k\to j \mid \underline{i}}$
if $|AICE_{\underline{i}\to j \mid k} - AICE_{k\to j \mid \underline{i}}| > \epsilon_0$ **then**
Append(candidates, CHECK_SYMMETRY(**M**, i, j, k, ∅))
Append(candidates, CHECK_SYMMETRY(**M**, i, k, j, ∅))
else
Append(candidates, CHECK_SYMMETRY(**M**, i, j, k, i))
Append(candidates, CHECK_SYMMETRY(**M**, k, i, i, j))
$\mathcal{I} \leftarrow \mathcal{I} \setminus (i, j, k)$ % Remove (i, j, k) triple from the set $\mathcal{I}$
return candidates

Figure 6.2 Update network paths function is called in Algorithm 1.

The estimated AICEs(i, j, k), $i \neq j \neq k$, have been demonstrated for the network in Figure 6.4a and 6.4f shown in Figure 6.5. The estimated values of AICEs differ significantly, which led us to the proposition that the difference in AICEs can be a way to find paths and remove the indirect paths in the first place. The schemes in Figure 6.4 have been tested by applying Algorithm 1. The adjacency matrices of all schemes in Figure 6.4 have been found correctly.

The detailed steps of Algorithm 1 for the network in Figure 6.4f have been illustrated in Figure 6.6. Four systems in the network have been randomly labeled as $\{a, b, c, d\}$. The algorithm runs by adding the input to the systems, and it has been

```
function CHECK_SYMMETRY_RMV(M, p, q, r) % Takes a connectivity matrix M, check paths in M and remove paths in temporary matrix C = [c_ij] ∈ ℝ^{N×N} by checking symmetry
    C ← M
    if index3 == ∅ then
        if has_path(M, p, q) then        c_pq = 0
    else
        if has_path(M, p, q) and has_path(M, q, r) then
            c_pq = 0, c_qr = 0, c_pr = 0, c_rp = 0
    return C
function CHECK_SOURCE_SINK(M, p, q, r) % Takes a connectivity matrix M, check paths in M and remove paths in temporary matrix C = [c_ij] ∈ ℝ^{N×N} in source and sink node
    C ← M
    if has_path(M, p, q) and has_path(M, q, r) then
            c_pq = 0, c_qr = 0
    return C
function CHECK_SYMMETRY(M, p, q, r, s) % Takes a connectivity matrix M, check paths in M and remove paths in temporary matrix C = [c_ij] ∈ ℝ^{N×N} in symmetric source/sink node or sink nodes
    C ← M
    if s == ∅ then
        if has_path(M, q, p) then
            c_qp = 0, c_rp = 0, c_pr = 0
    else
        if has_path(M, p, q) and has_path(M, s, r) then
            c_pq = 0, c_sr = 0,
    return C
```

Figure 6.3 Check_symmetry_rmv, check_source_sink and check_symmetry functions are called in update network paths function.

Table 6.2 Simulation parameters for the networks Figure 6.4

The Simulation Parameters	The Value
α	10
β	15
m_0	-1.31
m_1	-0.75
$\boldsymbol{\kappa}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
L	32000
K	1000
τ	61
d_e	3

Figure 6.4 Small networks to validate the proposed network reconstruction algorithm. Possible schemes are illustrated: (a) node 1 influences the remaining nodes in a linear chain; (b) all nodes influence each other in a closed cycle; (c) node 1 drives the remaining nodes; (d) node 1 influences node 2 and 3, node 3 is influenced by node 2 and drives node 4; (e) while there is a closed cycle between 1, 2 and 3, node 4 is influenced by node 3. (f) while node 1 drives node 2, node 2 drives node 3 and 4, there is also a path from 4 to 3; (g) node 1 influences node 2 and 3, node 3 is influenced by node 2 and drives node 4, node 4 drives node 1; (h) is a fully-connected network.

(a) The estimated AICEs for the network in Figure 6.4a.

(b) The estimated AICEs for the network in Figure 6.4f.

Figure 6.5 The cube centered at i, j, k represents the value of $AICE_{i \to j \mid k}$ in volume and color. The largest $AICE_{i \to j \mid k}$ is normalized to 1 unit^3.

represented with an arrow in Figures 6.6. Then, the necessary AICEs have been estimated, and the possible schemes have been listed and updated. The algorithm has eliminated irrelevant candidate networks such as unconnected graphs. Note that eliminated irrelevant candidate networks have been colored pale. In the end, the

algorithm has returned the estimated adjacency matrix of the network.

We have assessed the algorithm's performance by determining the number of steps required to reach the results for randomly coupled networks. We have generated random graphs by using a well-known Erdös-Rényi model (Erdős et al., 1960), and created random couplings with probabilities $p = 0.1$, $p = 0.2$ and $p = 0.3$ for the networks with $N = 12$ nodes. Then, we generated 20 different random schemes for each link probability. Illustrative examples that are selected randomly for $p = 0.1$, $p = 0.2$ and $p = 0.3$ are depicted in Figure 6.7. We kept all simulation parameters for these networks the same. We illustrate the number of iterations required versus the link probability p in Figure 6.8. The number of iterations has been normalized with respect to the maximum number of possible iterations. The Algorithm 1 starts with a fully-connected graph and proceeds by removing impossible links to reconstruct the network. Thus, the denser the network, the more iterations are required to complete the algorithm.

In all trials, the algorithm reconstructed the networks correctly.

In this section, we have investigated the network reconstruction problem in continuous-time chaotic networks in complete synchronization. We assume all the connected systems are identical, the couplings are unidirectional and have the same strength, but there is no prior knowledge about underlying dynamics or coupling strength. Besides, we only know that the systems in the network are completely synchronized a priori. The approach relies on the perturbation of the individual nodes of the network with random inputs at specific times to disturb synchronization and then the reconstruction of the state space from the observed outputs.

The reconstructed outputs have been then used to estimate one of the model-free information-theoretic measures, namely, the AICE, to quantify information flow in an interval between the subsystems in the network. We showed that the differences in AICEs between the reconstructed outputs obtained after injecting random information into the system reveal the connectivity information. Furthermore, we developed an algorithm to reconstruct the adjacency matrix of a given network, and we have tested

Figure 6.6 Illustrative representation of the proposed algorithm for the network in Figure 6.4f. The algorithm starts with a fully-connected graph at Stage 1. The input that is added to a system is represented with a waive arrow. The tested AICEs of i, j, k are listed in the right part for each stage. The irrelevant networks are shown as pale colors.

Stage 4:

$|AICE_{\underline{a}\to c \mid d} - AICE_{\underline{a}\to d \mid c}| = 0.034 \leq \epsilon$

$|AICE_{\underline{a}\to c \mid d} - AICE_{d\to c \mid \underline{a}}| = 0.042 \leq \epsilon$

$|AICE_{\underline{a}\to d \mid c} - AICE_{c\to d \mid \underline{a}}| = 0.009 \leq \epsilon$

Stage 5:

$|AICE_{\underline{a}\to d \mid b} - AICE_{\underline{a}\to b \mid d}| = 0.039 \leq \epsilon$

$|AICE_{\underline{a}\to d \mid b} - AICE_{b\to d \mid \underline{a}}| = 0.017 \leq \epsilon$

$|AICE_{\underline{a}\to b \mid d} - AICE_{d\to b \mid \underline{a}}| = 2.41 > \epsilon$

Stage 6:

$|AICE_{\underline{d}\to c \mid b} - AICE_{\underline{d}\to b \mid c}| = 21.99 > \epsilon$

$|AICE_{\underline{d}\to c \mid b} - AICE_{b\to c \mid \underline{d}}| = 1.96 > \epsilon$

$|AICE_{\underline{d}\to b \mid c} - AICE_{c\to b \mid \underline{d}}| = 0.30 > \epsilon$

Figure 6.6 Continues

(a) p=0.1 (b) p=0.2 (c) p=0.3

Figure 6.7 Randomly selected Erdös-Rényi networks of $N = 12$ with the probability $p = 0.1$, $p = 0.2$, and $p = 0.3$, respectively.

Figure 6.8 The ratio between # of iterations and maximum # of possible iterations of the algorithm with respect to link probability.

the algorithm with the randomly coupled chaotic networks to demonstrate the validity of the approach. The results indicate that the proposed algorithm can be used to reconstruct the network en route to re-synchronization.

Another curious question is whether the detection of clusters can be done in a chaotic dynamical network in cluster synchronization without the knowledge of the underlying structure of the network via the CE. The next chapter deals with this issue.

CHAPTER SEVEN
DETECTION OF CLUSTERS OF CHAOTIC DYNAMICAL NETWORKS VIA CAUSATION ENTROPY

Determining the network structures by gathering the observation vectors of the network can be tricky. The internal dynamics of individual systems and the connections between the systems are generally unknown. Furthermore, it is not always possible to observe all state variables of the individual systems of the network. It is essential to characterize such networks and find their communities, to understand complex structures such as the biological nervous systems or large power distribution systems. The existing studies have focused on the network reconstruction and the measures of structural properties of the network (Yu et al., 2006; Timme, 2007; Alderisio et al., 2017; Lou & Suykens, 2011). Inferring the structure of the cluster synchronization network via observations has not been investigated in detail so far.

Our goal in this chapter is to obtain the cluster structure of a network using a limited number of observations gathered from the network's individual members. We investigate the information flow between the systems in the continuous-time coupled chaotic networks when the clusters are re-forming a synchronization pattern after a perturbation. By perturbing the nodes, we inject additional information into the whole system. The clustered individual nodes tend to behave similarly, even if they are not synchronous yet after the perturbation. Hence, the information exchange during the re-formation of the clusters carries valuable clues. The suggested approach exploits this fact.

We assume that the individual dynamics of the nodes and the number of clusters are unknown in the network. Additionally, to observe the formation of clusters, we can perturb the systems in the network individually at specific times. Besides, we assume we can observe only one output from each system in the network. The observations can then be used to reconstruct the phase space according to Takens' embedding theorem (Takens, 1981). We estimate the CE by using the reconstructed state space obtained from the observations. Finally, we estimate ICE and its expected value, i.e., AICE, to

detect the existence of cluster synchronization in the network.

We show that the AICE can be used to classify whether the systems belong to a particular cluster. We demonstrate that we obtain clusters in the coupled chaotic networks of Chua's systems by estimating the AICEs and clustering obtained AICE values using the basic k-means classification algorithm (Hartigan, 1975).

Furthermore, we discuss the problem of identifying new clusters due to changing coupling strengths of the network.

7.1 The Problem Statement

Consider a network with N coupled with identical dynamical systems; the network forms m clusters according to definition 2.4.3. Let $i,\ j \subset \mathcal{V}$ be any network nodes that are selected randomly. $i \sim j$ shows that i and j sub-systems are in the same cluster such that $i,\ j \in G_{\mathcal{I}}$. We perturb i^{th} system every t_n seconds. Since the synchronization manifold is stable, i and j are still members of the same cluster after the perturbations. On the other hand, consider that $i \in G_{\mathcal{I}}$ but $\ell \notin G_{\mathcal{I}}$. When the perturbation is repeated every t_n seconds, i and ℓ are different clusters again. Therefore, we assume that the cluster scheme in the network remains the same when we perturb each system individually.

We reconstruct the systems' state-space individually by observing only one of each system's output.

To analyze the information transfer between the inter- and the intra-cluster nodes, we estimate the CEs of the observations between two perturbations. The perturbations inject additional information, and we track how this information flows through the network as the clusters reform. In steady-state, the CE values would be zero identically without such perturbations, avoiding the determination of the membership because no new information would be generated by observing any cluster member. The information injected by the perturbations results in positive CE values on the way to synchronization. We propose that the response to such information injections must

be similar among the cluster members and can be exploited to determine cluster memberships.

The ICE measure in (6.3) is estimated to quantify total information flow while cluster synchronization re-occurs after a perturbation. The properties of $AICE$ in (6.4) in cluster synchronization are fairly similar to the CE. When i^{th} and k^{th} systems belong to the same cluster, $p(\tilde{y}_i^{(d_1)}) = p(\tilde{y}_k^{(d_1)})$. Therefore,

$$\begin{aligned} AICE_{i\to j\mid k} &= E_{u_i}\left[ICE_{i\to j\mid k}(u_i)\right] \\ &= E_{u_i}\left[E_n\left[h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}_{k,n}^{(d_2)}, \boldsymbol{U}_i) - h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}_{k,n}^{(d_2)}, \tilde{\boldsymbol{Y}}_{i,n}^{(d_1)}, \boldsymbol{U}_i)\right]\right] \\ &= E_{u_i}\left[E_n\left[h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}_{k,n}^{(d_2)}, \boldsymbol{U}_i) - h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}_{k,n}^{(d_2)}, \boldsymbol{U}_i)\right]\right] = 0. \end{aligned} \tag{7.1}$$

Otherwise,

$$AICE_{i\to j\mid k} = E_{u_i}\left[E_n\left[h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}_{k,n}^{(d_2)}, \boldsymbol{U}_i) - h(\tilde{Y}_{j,n+1}\mid \tilde{\boldsymbol{Y}}_{k,n}^{(d_2)}, \tilde{\boldsymbol{Y}}_{i,n}^{(d_1)}, \boldsymbol{U}_i)\right]\right] > 0, \tag{7.2}$$

since, $h(\tilde{Y}_{j,n+1}\mid \tilde{\mathbf{Y}}_k^{(d_2)}, \boldsymbol{U}_i) > h(\tilde{Y}_{j,n+1}\mid \tilde{\mathbf{Y}}_i^{(d_2)},\ \tilde{\mathbf{Y}}_k^{(d_1)}, \boldsymbol{U}_i)$.

There is no guarantee that transient responses generate stationary or even ergodic processes after perturbations. Therefore, any method based on time-averaging is not suitable for analysis directly. The perturbation in (6.2) can be viewed as restart of the system at every t_n. Hence, the averaging over n is averaging over an ensemble of perturbations, which corresponds to ensemble averaging over the random variable $ICE_{i\to j\mid k}$.

Since, the underlying probability density can be time-varying when calculating $AICE_{i\to j\mid k}$ from estimated values of $ICE_{i\to j\mid k}$, we set the time windows, $[t_n, t_{n+1}], n = 0, 1, \ldots$, to be overlapping. On the other hand, the methods given by (5.26), result in local estimations with respect to time inside these frames (Kraskov

et al., 2004; Zhu et al., 2015). The consequence is that the pdf variations along the transient trajectory are taken into account naturally. Therefore, the assumption of ergodicity is not strictly necessary. We only assume that the pdf is continuous with respect to time, and the overlapping ratio is chosen suitably to follow the time variations from frame to frame.

The estimated AICEs vary considerably depending on whether the systems indexed by i, j, and k belong to the same cluster or not. Therefore, we can estimate the AICEs for all systems in the network by using (6.4), to detect the members of the clusters in the network by classifying the estimations using k-means classification algorithm.

We determine the number of optimal clusters of AICEs by using the elbow method in the k-means algorithm (Hartigan, 1975). Let us denote the number of optimal clusters as N_c, and the centroid of these clusters is given as :

$$c_\ell^\star = \underset{c\in\mathbb{R}^+\cup\{0\}}{\arg\min} \sum_{\ell=1}^{N_c} \sum_{i,j,k} \left|AICE_{i\to j|k} - c\right|^2 \tag{7.3}$$

We assign all AICEs to the closest centroid by computing $\mathcal{I} = \underset{\ell}{\arg\min} \left|AICE_{i\to j|k} - c_\ell^\star\right|$, i.e. $i, k \in G_{\mathcal{I}}$.

The overall procedure has been summarized in Algorithm 2.

Algorithm 2 Algorithm for detection of the clusters

Input: $\mathcal{N} = \{1, 2, \ldots, N\}$: index set of nodes, $\mathcal{Y} = \{y_{a_1}, y_{a_2}, \ldots, y_{a_N}\}$: the systems' outputs , L: the frame length, K: the number of iterations, $0 < \sigma \leq 1$: the overlapping ratio.

Output: m: the number of clusters, $G_{\mathcal{I}}$: the members of the cluster $\mathcal{I}$, $\mathcal{I} \in \mathbb{Z}^+$

Initialize:

1: Obtain τ and d_e of individual systems for the reconstruction state space by using y_{a_i}'s,
2: Reconstruct the state space according to Takens' theorem
3: $\mathcal{I} \leftarrow 1$
4: **while** $\mathcal{N} \neq \emptyset$ **do**
5: **for** $i \in \mathcal{N}$ **do**
6: **for** $n \leftarrow 0$ to $K - 1$ **do**
7: **for** $j \in \mathcal{N}$ **do**
8: **for** $k \in \mathcal{N}$ **do**
9: **for** $[\ell, \ell + L]$, $\ell = t_n, t_n + \sigma L, t_n + 2\sigma L, \cdots, t_{n+1} - L$ **do**
10: Estimate $CE_{i \to j \mid k}(\ell)$ by using (5.26)
11: Estimate $ICE_{i \to j \mid k}(n)$ using (6.3)
12: Calculate $AICE_{i \to j \mid k} \quad \forall \quad j, k$ using (6.4)
13: Construct a matrix of $AICE_{i \to j \mid k}$, where each row represents j^{th} output, each column represents k^{th} output.
14: Sort columns and rows in ascending order.
15: Classify $AICE_{i \to j \mid k}$ by using k-means algorithm (7.3) **return** $G_{\mathcal{I}}$
16: Update $\mathcal{N}$ by removing the members of the cluster $\mathcal{I}$, i.e. $\mathcal{N} \leftarrow \mathcal{N} \setminus G_{\mathcal{I}}$
17: $\mathcal{I} \leftarrow \mathcal{I} + 1$
18: $m \leftarrow \mathcal{I}$
19: **return** m and $G_1, , G_2, \ldots, G_m$

The kd-tree searching algorithm used at step 10 in Algorithm 2 has computational complexity of $O(L \log L)$ (Bentley, 1975). To smooth ICE estimation at step 11, we use overlapping frames where $0 < \sigma \leq 1$ determines the successive overlapping frame ratio. Hence, the computational complexity for each iteration of (i, j, k) is $O(K \times M \times L \times \log L)$ where M is the number of frames. However, the procedure does not need to apply for every (i, j, k). The index set of newly obtained clusters can be removed as in step 16. The AICEs can be estimated without requiring the calculation of the AICEs of the previously determined clusters. Therefore, if the number of clusters is m, the complexity of the algorithm is $O(m \times N^2 \times K \times M \times L \times \log L)$. The variant of k-means classification algorithm used at step 15 is known to have time complexity

of $O(m \times N^2 \times P)$ on average, where P is the number of iterations until convergence (Lloyd, 1982). It can be replaced by any available clustering algorithm conveniently. The classification algorithm can return several clusters at each iteration of step 15 at once. The overall time complexity will reduce further. In the worst case, when $N = m$, the complexity becomes $O(N^3 \times K \times M \times L \times \log L)$, which corresponds to no synchronization at all. However, we note that, as the clusters are determined and removed from the index set, the iterations get faster, effectively.

7.2 Illustrative Examples

We have investigated the coupled chaotic networks which contain identical Chua systems defined in (2.13) to demonstrate the outcome. The individual systems show chaotic behavior in both networks according to definition 2.2.4. Each Chua system has been illustrated with a circle, and a specific color represents each cluster in Figure 7.1. The network can form 3 different clusters with 8 Chua systems for a particular choice of parameters in Figure 7.1a. The coupling function has been chosen as $h_i(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i - \mathbf{x}_j$. The parameters for the network shown in Figure 7.1a have been selected same as in Table 6.2. The connection matrix is given as:

$$\mathbf{A} = [a_{ij}] = \begin{bmatrix} -3 & 3 & -1 & 1 & 0 & 0 & 0 & 0 \\ 3 & -3 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & -10 & 5 & 5 & -2 & 1 & 1 \\ 1 & -1 & 5 & -10 & 5 & 1 & -2 & 1 \\ 0 & 0 & 5 & 5 & -10 & 1 & 1 & -2 \\ 0 & 0 & -2 & 1 & 1 & -6 & 3 & 3 \\ 0 & 0 & 1 & -2 & 1 & 3 & -6 & 3 \\ 0 & 0 & 1 & 1 & -2 & 3 & 3 & -6 \end{bmatrix} \tag{7.4}$$

For $\epsilon = 2$, the network has 3 clusters in Figure 7.1a by using design rules definition in 3.2.1.

Figure 7.1 Illustrative cluster synchronization networks, where each node represents a Chua system. There are 3 clusters in (a) given as $G_1 = \{1,\ 2\}$, $G_2 = \{3,\ 4,\ 5\}$, $G_3 = \{6,\ 7,\ 8\}$, 4 clusters in (b) given as $G_1 = \{1\}$, $G_2 = \{4\}$, $G_3 = \{6\}$, $G_4 = \{2,\ 3,\ 5,\ 7\}$.

The network with 7 Chua systems and 4 clusters has been illustrated in Figure 7.1b. Curiously, the clusters can be formed by network elements that are not directly coupled. The parameters have been shown in Table 7.1. According to Gambuzza et al. in Section 3.2.3, a choice of connection matrix as

$$\mathbf{A} = [a_{ij}] = \begin{bmatrix} -4 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & -2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & -3 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & -2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & -2 \end{bmatrix} \tag{7.5}$$

Table 7.1 Simulation parameters for the network 7.1b

The Simulation Parameters	The Value
α	10
β	15
γ	0.0385
m_0	-1.27
m_1	-0.68
$\boldsymbol{\kappa}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
L	32000
K	1000
τ	55
d_e	3

results in a scheme of 4 clusters when ϵ is equal to 0.5 for the network in Figure 7.1b.

The network that forms 17 clusters with 50 Chua systems has been shown in Figure 7.2a. The parameters for the Chua system in this network are the same as the parameters for the network in Figure 7.1a. The connection matrix for the network is represented as in Figure 7.2b, and it is designed by using the design rules in definition 3.2.1.

The simulations have been performed using the *odeint* function in Python's scipy library to solve the network equations (Jones et al., r 07). The sampling period has been selected as $t_s = 0.01 sec$. Note, if there exists t_f such that $\|\mathbf{x}_i(t-t_f)-\mathbf{x}_j(t-t_f)\| < \varepsilon \ll 1$, $\forall\ t > t_f,\ t \in (t_n,\ t_{n+1})$ between two perturbations, then $i \sim j \in G_{\mathcal{I}}$. Otherwise, there exists t_f such that $\|\mathbf{x}_i(t-t_f) - \mathbf{x}_\ell(t-t_f)\| \geq L_e$ for some $L_e > 0$, $\forall\ t > t_f,\ t \in (t_n,\ t_{n+1})$ between two perturbations, then $i \nsim \ell \notin G_{\mathcal{I}}$. To clarify, the chaotic behavior of the systems in the network has been guaranteed and the clustering scheme preserved after the perturbations.

(a)

(b)

Figure 7.2 An illustrative cluster synchronization network, where each node represents a Chua system. There are 17 clusters in (a) The connection matrix of (a) has been illustrated in (b).

Since, by assumption, we do not know the number of clusters in the network beforehand, we have labeled the outputs randomly with a_1, $a_2, \ldots, a_N$ for the

networks. The time lags have been selected as $d_1 = d_2 = 1$ in (7.1). The state spaces have been reconstructed according to Takens' theorem. We have estimated the CEs between the systems using a limited set of reconstructed state space samples according to (5.26). We have chosen the frame length to be $L = 8000$ observations. An overlap between the frames was 90% of the frame length (i.e., $\sigma = 0.1$). The number of neighbors in (5.26) has been selected as $k = 10$. The Algorithm 2 has been used to determine cluster membership.

Figure 7.3 The AICE from a_2^{th} system to other systems by observing any system ($AICE_{a_2 \to j|k}$ $\forall$ j, k) for the network in Figure 7.1a. They have been illustrated in a color bar when we are perturbing the node $\mathcal{I} = \{a_2\}$.

Since the networks in Figure 7.1a and Figure 7.1b are relatively small, we have skipped step 4 in the algorithm and calculated all AICEs for all nodes in the networks for illustrative purposes.

Resulting AICEs have been represented in Figures 7.3, 7.4, 7.5, 7.6 and 7.7 for the network in Figure 7.1a. Suppose we choose an arbitrary system from the network, e.g., a_3. We have the AICE's from a_3 to other systems in Figure 7.4. It has been observed that

Figure 7.4 The AICE from a_3^{th} system to other systems by observing any system ($AICE_{a_3 \to j|k}$ $\forall$ j, k) for the network in Figure 7.1a. They have been illustrated in a color bar when we are perturbing the node $\mathcal{I} = \{a_3\}$.

$$AICE_{a_3 \to j \mid a_1} \approx AICE_{a_3 \to j \mid a_3} \approx AICE_{a_3 \to j \mid a_5} \approx 0 \quad \forall\, j \tag{7.6}$$

$$\begin{aligned} &AICE_{a_3 \to j \mid a_2} \approx AICE_{a_3 \to j \mid a_4} \approx AICE_{a_3 \to j \mid a_6} \approx \\ &AICE_{a_3 \to j \mid a_7} \approx AICE_{a_3 \to j \mid a_8} \approx 325 \text{ nats.sec } j \in \{a_1,\ a_3,\ a_5\} \end{aligned} \tag{7.7}$$

One can deduce that a_3 is in the same cluster with a_1 and a_5 and it is not in the same cluster with $a_2,\ a_4,\ a_5,\ a_7,\ a_8$ ($a_3 \sim a_1$, $a_3 \sim a_5$ and $a_3 \nsim \{a_2,\ a_4,\ a_5,\ a_7,\ a_8\}$). There are also 2 different mid-levels in the color bars, as shown in Figure 7.7. It can be seen that $AICE_{a_3 \to a_2|a_6} \neq AICE_{a_3 \to a_4|a_6}$. We can infer that $a_3 \nsim a_2$, $a_3 \nsim a_4$ and $a_2 \nsim a_4$. By examining $AICE_{a_3 \to a_8|a_7} > AICE_{a_3 \to a_2|a_7} \gg 0$, we also say that $a_3 \nsim a_2 \nsim a_8$. A similar line of reasoning will yield the number of clusters and each cluster's members.

Figure 7.5 The AICE from a_5^{th} system to other systems by observing any system ($AICE_{a_5 \to j|k}$ $\forall$ j, k) for the network in Figure 7.1a.They have been illustrated in a color bar when we are perturbing the node $\mathcal{I} = \{a_5\}$.

When we analyze the AICEs from a_2, a_5 and a_6 to any other systems in Figure 7.7, it has been observed then

$$\begin{aligned} &AICE_{a_2 \to j \mid a_2} \approx AICE_{a_2 \to j \mid a_7} \approx AICE_{a_5 \to j \mid a_1} \approx AICE_{a_5 \to j \mid a_3} \\ &\approx AICE_{a_5 \to j \mid a_5} \approx AICE_{a_8 \to j \mid a_4} \approx AICE_{a_8 \to j \mid a_6} \approx AICE_{a_8 \to j \mid a_8} \approx 0 \quad \forall\, j \end{aligned} \tag{7.8}$$

While a_2 and a_7 are forming a cluster ($a_2 \sim a_7$), the third cluster consists of a_4, a_6 and a_8 ($a_4 \sim a_6 \sim a_8$). Also, a_5 forms a cluster with a_3 and a_1 ($a_5 \sim a_3 \sim a_1$). Besides,

$$\begin{aligned} &AICE_{a_2 \to a_7 \mid a_3} > AICE_{a_2 \to a_6 \mid a_3} > AICE_{a_2 \to a_1 \mid a_3} \gg 0 \\ &AICE_{a_8 \to a_4 \mid a_7} > AICE_{a_8 \to a_3 \mid a_7} > AICE_{a_8 \to a_2 \mid a_7} \gg 0. \end{aligned} \tag{7.9}$$

We have noticed that the AICEs are not equal when a_2 is in the different clusters with a_1 and a_6 by observing a_3. Even though a_2 and a_7 are in the same cluster, it is seen

Figure 7.6 The AICE from a_8^{th} system to other systems by observing any system ($AICE_{a_8 \to j|k}$ $\forall$ j, k) for the network in Figure 7.1a. They have been illustrated in a color bar when we are perturbing the node $\mathcal{I} = \{a_8\}$.

that the information transfer from a_2 to a_7 by observing a_3 is quite greater than zero. According to (7.9), the AICE of a_8 to a_4 by observing a_7 is considerably greater than the information transfer from a_8 to $\{a_3,\ a_2\}$ by observing a_7 . Hence, we have obtained that a_4 and a_8 are the members of the same cluster ($a_4 \sim a_8$). Also, we have known that $a_8 \nsim a_3 \nsim a_2$, and $a_8 \nsim a_7$ but $a_2 \sim a_7$ from previous outcomes. We observe that if $i \nsim j \sim k$, the $AICE_{i \to j \mid k} > 0$. Compared to the case, $i \nsim j \nsim k$, the information flow between the systems decreases when $i \nsim j \sim k$. As a consequence, we conclude that there are 3 clusters $G_1 = \{a_2,\ a_7\}$, $G_2 = \{a_1,\ a_3,\ a_5\}$, $G_3 = \{a_4,\ a_6,\ a_8\}$, respectively in Figure 7.7.

The AICEs have demonstrated in Figure 7.8 for the network in Figure 7.1b. We see that

$$AICE_{a_2 \to j \mid a_2} = 0 \quad \forall\ j, \tag{7.10}$$

Figure 7.7 The AICEs have been represented after the algorithm has been performed when we are perturbing the nodes in a_2, a_3, a_5 and a_8, respectively for the network as shown in Figure 7.1a.

obviously, the remaining AICE's are slightly greater than zero. If we choose any of the j in Figure 7.1b, we observe that

$$\begin{aligned} &AICE_{a_2\to j\mid a_4} > AICE_{a_2\to j\mid a_6} > AICE_{a_2\to j\mid a_1} \approx \\ &AICE_{a_2\to j\mid a_3} \approx AICE_{a_2\to j\mid a_5} \approx AICE_{a_2\to j\mid a_7} \gg 0. \end{aligned} \tag{7.11}$$

a_2 is a unique member of one of the clusters. Figure 7.8 indicates that

$$AICE_{a_4\to j\mid a_4} \approx AICE_{a_6\to j\mid a_6} \approx 0 \quad \forall\, j. \tag{7.12}$$

Moreover,

$$
\begin{aligned}
AICE_{a_4 \to j \mid k} &= 0 \quad \forall\, j,\ k = \{a_1,\ a_2,\ a_3,\ a_5,\ a_6,\ a_7\} \gg 0 \\
AICE_{a_6 \to j \mid k} &= 0 \quad \forall\, j,\ k = \{a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_7\} \gg 0.
\end{aligned} \tag{7.13}
$$

As a result, a_4 and a_6 are forming clusters individually.

$$
\begin{aligned}
AICE_{a_7 \to j \mid k} &\approx 0 \quad \forall\, j,\ k \in \{a_1,\ a_3,\ a_5,\ a_7\} \\
AICE_{a_7 \to j \mid k} &> 0 \quad \forall\, j,\ k \in \{a_2,\ a_4,\ a_6\}\,.
\end{aligned} \tag{7.14}
$$

Therefore, a_1, a_3, a_5 and a_7 belong to same cluster. We can conclude that the clusters are $G_1 = \{a_2\}$, $G_2 = \{a_4\}$, $G_3 = \{a_6\}$, $G_4 = \{a_1,\ a_3,\ ,a_5\ ,a_7\}$ for the network in Figure 7.1b.

Figure 7.8 The AICEs have been represented after the algorithm has been performed when we are perturbing the nodes in a_2, a_4, a_6 and a_7, respectively for the network as shown in Figure 7.1b.

Table 7.2 The results of k-means classification algorithm for $AICEs$ as shown in Figure 7.8 when $i = a_2$, $i = a_7$, $i = a_4$ and $i = a_6$, respectively $\forall\ j,\ k$ for the network in Figure 7.1b

i	**j**	**k**	$\boldsymbol{\ell}$	$\mathbf{c}^{\star}_{\ell}$
a_2	$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7$	a_2	1	0
	$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7$	$a_1,\ a_3,\ a_5,\ a_7$	2	61.80
	$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7$	a_6	3	159.95
	$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7$	a_4	4	272.61
a_7	$a_1,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7$	$a_1,\ a_3,\ a_5,\ a_7$	1	8.20
	$a_1,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7$	a_6	2	140.80
	$a_1,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7$	a_4	3	269.68
a_4	$a_4,\ a_6$	a_4	1	0
	$a_4,\ a_6$	a_6	2	243.77
a_6	a_6	a_6	1	159.52

Additionally, we have demonstrated the results of the k-means classification algorithm for the sake of repeatability. The k-means algorithm takes the $AICEs$ as an input, obtains the optimal centroids, and returns the members of the clusters. The clusters of the network by applying Algorithm 2 have been presented in Table 7.2 together with the obtained centroids.

When we inspect Figures 7.7 and 7.8, the AICEs of the systems slightly change in case of i and k are fixed but j is varying ($j \in \{a_1, \ldots, a_N\}$). Besides, if i and j are fixed but system k is changed, the AICEs of systems vary. The $AICE_{i \to j \mid k}$'s mainly depend on the choice of systems i and k, but not j.

The network in Figure 7.2a is substantially larger, and it is not easy to detect cluster schemes by inspection, only. Figure 7.9a illustrates the AICE obtained for this network when randomly chosen system a_{16} is perturbed. Once a cluster is determined, only AICEs for the remaining systems have to be extracted. In the first step, the members of the cluster G_1 has been found to be a_{16}, a_{34} and a_{46}. In the second iteration, Figure 7.9c shows the estimation of the $AICE_{a_4 \to j|k}$'s for $\mathcal{I} \setminus \{a_{16}, a_{34}, a_{46}\}$ when we are perturbing the randomly chosen system a_4. Note that, the some of AICEs ($AICE_{a_4 \to j|k}$ $j, k \in$

Figure 7.9 (a) The AICEs of the network as shown in Figure 7.2a when we perturb a_{16} in the first iteration of algorithm. (b) The sorted AICEs of a_{16}. (c) The corresponding AICEs of a_4 were chosen to be perturbed in the second iteration. (d) The sorted AICEs for $\mathcal{I} \setminus \{a_{16}, a_{34}, a_{46}\}$.

$\{a_{16},\ a_{34},\ a_{46}\}$) are not estimated and not shown in Figure 7.9c because we know the cluster members as $G_1 = \{a_{16}, a_{34}, a_{46}\}$ in the first iteration (Figure 7.9b), already.

For the cluster to which the perturbed node belongs, AICEs tend to be zero. But, the rest of the sorted $AICE \neq 0$ also yield valuable information about the remaining clusters. Therefore, obtaining more than one cluster at each iteration of step 4 is possible. Additionally, when AICEs are sorted, the matrix includes clues to choose

which system is to be perturbed next, as shown in inset indicators of Figures 7.9b and 7.9d. The nodes corresponding to indicated diagonal blocks have very close AICE values, hence harder to classify. Such groups tend to be classified as synchronous clusters, but they may not be. One can choose the next node to perturb from such a set of nodes instead of choosing a random node at each iteration. This reduces the number of overall iterations. For clarity, we only removed the nodes for which AICE tends to zero; hence, we did not exploit this fact.

When the algorithm is completed, the following clusters are obtained, which is expected as the work of Ma et al. indicates (Ma et al., 2006).

$$
\begin{aligned}
&G_1 = \{a_{16}, a_{34}, a_{46}\} && G_2 = \{a_4, a_{19}\} && G_3 = \{a_9, a_{21}, a_{23}, a_{24}, a_{26}\} \\
&G_4 = \{a_6, a_{18}, a_{20}\} && G_5 = \{a_{25}, a_{29}, a_{43}\} && G_6 = \{a_{17}, a_{49}\} \\
&G_7 = \{a_{27}, a_{28}, a_{32}\} && G_8 = \{a_5, a_8, a_{14}, a_{36}, a_{44}\} && G_9 = \{a_1, a_3, a_{37}\} \\
&G_{10} = \{a_{11}, a_{31}, a_{42}\} && G_{11} = \{a_2, a_7, a_{39}\} && G_{12} = \{a_{15}, a_{40}, a_{48}\} \\
&G_{13} = \{a_{10}, a_{30}, a_{41}\} && G_{14} = \{a_{13}, a_{22}, a_{47}\} && G_{15} = \{a_{33}, a_{38}\} \\
&G_{16} = \{a_{35}, a_{45}\} && G_{17} = \{a_{12}, a_{50}\}
\end{aligned}
\tag{7.15}
$$

In this section, we have examined the problem of detecting clusters in coupled continuous-time chaotic networks via information measures. We assume that sub-systems in the network are forming clusters, and the number of clusters and the memberships of the nodes in the network are unknown. The individual systems are perturbed, and their outputs can be observed in a limited time interval. We have reconstructed the state space of systems by using the observations according to Takens' embedding theorem. We have proposed the AICE measure to distinguish the clusters of the networks. Since there is a diffeomorphism (i.e., differentiable transformation with differentiable inverse) between the original state space and reconstructed state space, we estimate the AICEs in the reconstructed state, which is equivalent to the AICEs of the entire state space. We found that the AICEs of the

systems vary depending on whether the sub-systems in the network belong to the same cluster. Besides, when we classify the AICEs of all systems in the network with the k-means algorithm, we can distinguish the clusters from the observations only, and no information is required about the individual node model. We have shown in our numerical results that the AICEs can distinguish the clusters and obtain the members of clusters regardless of whether the members of clusters are directly connected or not. The proposed procedure successfully determines the cluster scheme in the continuous-time dynamical coupled networks of chaotic systems.

7.3 Detection of Cluster Synchronization when the Clusters Change

We have addressed the problem of detection of cluster synchronization using observation samples in a coupled continuous-time chaotic network when the coupling's strengths are changed. Additionally, it has been shown that the clusters can be determined even if the network forms new clusters in case of the changing coupling strengths. The CE has been estimated with observation vectors obtained from the outputs of the nodes of the network to detect cluster synchronization.

We have chosen a coupled chaotic network, and the state equations are given by using (2.6). Consider this network with five identical chaotic Lorenz systems given as (2.8). The systems in the network are coupled with x_{i1}-variables. The parameters for the Lorenz system have been selected as $\sigma = 10.0$, $b = 8/3$, $r = 28.0$. It is known that each system shows chaotic behavior for these parameters. The connection matrix has been chosen as follows:

$$\mathbf{A} = [a_{ij}] = \begin{bmatrix} -2\alpha_{11} & \alpha_{11} & \alpha_{11} & 2\alpha_{12} & 3\alpha_{12} \\ \alpha_{11} & -2\alpha_{11} & \alpha_{11} & 2\alpha_{12} & 3\alpha_{12} \\ \alpha_{11} & \alpha_{11} & -2\alpha_{11} & \alpha_{12} & 4\alpha_{12} \\ \alpha_{21} & 2\alpha_{21} & 3\alpha_{21} & -2\alpha_{22} & 2\alpha_{22} \\ 3\alpha_{21} & 2\alpha_{21} & \alpha_{21} & 2\alpha_{22} & -2\alpha_{22} \end{bmatrix} \tag{7.16}$$

Here, the parameters of the connection matrix have been selected as $\alpha_{11} = 2.5$, $\alpha_{12} = \alpha_{21} = 0.5$. It is known that the network can form different cluster schemes by changing the coupling strength α_{22} (Belykh et al., 2008). For example, the network has four clusters when α_{22} is equal to 0.5. However, if α_{22} is selected as 3.0, the network forms two clusters. Both cases for the network have been illustrated in Figure 7.10.

The systems in the network are randomly labeled as $\{a, b, c, d, f\}$. The observations have been gathered, and phase space has been reconstructed according to Takens' theorem. Hence, the first minimum of time-delayed MI and the embedding dimension have been estimated for the collected observation samples. It is assumed that the chaotic attractor forms for every system. The reconstruction of the data part has been repeated for increasing values of α_{22}. Finally, we have estimated the CEs for each system in the reconstructed space by overlapping frames.

τ and d_e have been estimated as 16 and 3 for the individual Lorenz systems. We have waited long enough for the transients to die out (325 sec). Then, we repeated the reconstruction process for the observation vectors by changing the α_{22} coupling strength. We have estimated the CEs for each node from the reconstructed observation vectors by taking $40sec$ frames. The differential equations have been solved numerically. The sampling time is $t_s = 0.01$. We have repeated the simulations 1000 times.

We have investigated the CEs from given b^{th} system to other systems, $CE_{b \to j \mid f} \approx 0 \; \forall \; j = \{a, b, c, d, f\}$, in $(i, k) = (b, f)$ column. It is shown that b and f are in the same cluster for all α_{22} between 1.2 and 2.2. When $\alpha_{22} = 1.2$, all CEs except $CE_{b \to j \mid f} \approx 0 \; \forall \; j = \{a, b, c, d, f\}$ are greater than zero. Hence, there is no other system inside clusters b and f. When α_{22} is increased, $CE_{b \to j \mid c} \; \forall \; j = \{a, b, c, d, f\}$ approaches to zero. In this case, b, c, and f systems form a new cluster. It is seen that the CEs of the systems are equal to zero when the new cluster occurs and is indicated by a black arrow in Figure 7.11.

In Figure 7.12, $i = c$ and the CEs are illustrated by fixing i^{th} system and for all j systems and for different pairs of (i, k), respectively, in each column. The CEs from c^{th}

(a) Cluster synchronization network. There are 4 different clusters for $\alpha_{11} = 2.5$, $\alpha_{12} = \alpha_{21} = 0.5$, $\alpha_{22} = 0.5$.

(b) Cluster synchronization network. There are 2 different clusters for $\alpha_{11} = 2.5$, $\alpha_{12} = \alpha_{21} = 0.5$, $\alpha_{22} = 3.0$.

Figure 7.10 The coupled chaotic networks. Each circle represents a Lorenz system, and the arrows indicate the coupling strengths ϵ_{ij} between i^{th} and j^{th} systems.

system to other systems are greater than zero $CE_{c\to j|k} \neq 0 \; \forall \; j, k = \{a, b, c, d, f\}, k \neq c$ in case of $\alpha_{22} = 1.2$. When α_{22} equals to 2.2, $CE_{c\to j|b} \approx CE_{c\to j|f} \approx 0 \; \forall \; j = \{a, b, c, d, f\}$. We have obtained that the cth system forms a new cluster with b and f systems.

Figure 7.13 has depicted $i = d$ and the CEs are illustrated by fixing i^{th} system and for all j systems and for different pairs of (i, k), respectively. The CEs from d^{th} system to other systems are also greater than zero $CE_{d\to j|k} \neq 0 \; \forall \; j, k = \{a, b, c, d, f\}$, $k \neq d$ when $\alpha_{22} = 1.2$. In case of α_{22} is greater than 2.0, the CEs from d^{th} system to given a^{th} system become zero $CE_{d\to j|a} = 0 \; \forall \; j = \{a, b, c, d, f\}$. The result indicated that the dth and a^{th} system belong to the same cluster.

The CEs have been illustrated by fixing $i^{th} = a$ system and for all j systems and for different pairs of (i, k), respectively in Figure 7.14. It is seen that the CEs from a^{th} system to other systems are greater than zero $CE_{a\to j|k} \neq 0 \; \forall \; j, k = \{a, b, c, d, f\}$, $k \neq a$ when $\alpha_{22} = 1.2$. a^{th} system is the only member of its cluster in case of $\alpha_{22} = 1.2$. When α_{22} is increased, $CE_{a\to j|d} \; \forall \; j \in \{a, b, c, d, f\}$, approach to zero. We have determined that a^{th} and d^{th} system are forming a new cluster.

Finally, Figure 7.15 have demonstrated the CEs are illustrated by fixing $i^{th} = f$

Figure 7.11 The errors between the outputs and the CEs of the systems versus time for varying coupling parameters when $i = b$. The upper row shows the error between the outputs versus time for varying coupling parameter α_{22}, i.e. $e_{ij}(t) = y_i(t) - y_j(t)$, $i = \{b\}$, $\forall\ j \in \{a, c, d, f\}$, respectively. The CEs of the systems versus time is illustrated for varying coupling parameter α_{22} at the bottom row. While the right axis of the bottom row shows α_{22}, the left axis of the bottom row illustrates the CEs from i^{th} system to j^{th} by observing k^{th} system. In each column, $i = b$ and the CEs are illustrated by fixing i^{th} system and for all j systems and for different pairs of (i, k), respectively.

Figure 7.12 The errors between the outputs and the CEs of the systems versus time for varying coupling parameter when $i = c$.The upper row shows that the error between the outputs versus time for varying coupling parameter α_{22}, i.e. $e_{ij}(t) = y_i(t) - y_j(t)$, $i = \{c\}$, $\forall\ j \in \{a, b, d, f\}$, respectively. The CEs of the systems versus time are illustrated at the bottom row for varying coupling parameter α_{22}. While the the right axis of the bottom row shows α_{22}, and the left axis of the bottom row illustrates the CEs from i^{th} system to j^{th} by observing k^{th} system.

Figure 7.13 The errors between the outputs and the CEs of the systems versus time for varying coupling parameter when $i = d$.The upper row shows that the error between the outputs versus time for varying coupling parameter α_{22}, i.e. $e_{ij}(t) = y_i(t) - y_j(t)$, $i = \{d\}$, $\forall\ j \in \{a, b, c, f\}$, respectively. The CEs of the systems versus time are illustrated at the bottom row for varying coupling parameter α_{22}. While the the right axis of the bottom row shows α_{22}, and the left axis of the bottom row illustrates the CEs from i^{th} system to j^{th} by observing k^{th} system.

Figure 7.14 The errors between the outputs and the CEs of the systems versus time for varying coupling parameter when $i = a$. The upper row shows the error between the outputs versus time for varying coupling parameter α_{22}, i.e. $e_{ij}(t) = y_i(t) - y_j(t)$, $i = \{a\}$, $\forall\ j \in \{b, c, d, f\}$, respectively. The CEs of the systems versus time are illustrated at the bottom row for varying coupling parameter α_{22}. While the the right axis of the bottom row shows α_{22}, and the left axis of the bottom row illustrates the CEs from i^{th} system to j^{th} by observing k^{th} system.

Figure 7.15 The errors between the outputs and the CEs of the systems versus time for varying coupling parameter when $i = f$. The upper row shows the error between the outputs versus time for varying coupling parameter α_{22}, i.e. $e_{ij}(t) = y_i(t) - y_j(t)$, $i = \{f\}$, $\forall\ j \in \{a, b, c, d\}$, respectively. The CEs of the systems versus time are illustrated at the bottom row for varying coupling parameter α_{22}. While the the right axis of the bottom row shows α_{22}, and the left axis of the bottom row illustrates the CEs from i^{th} system to j^{th} by observing k^{th} system.

system and for all j systems and for different pairs of (i, k), respectively. When $1.2 \leq \alpha_{22} \leq 2.2$, $CE_{f \to j \mid b} \approx 0 \; \forall \; j = \{a, b, c, d, f\}$, in $(i, k) = (f, b)$ column. If $\alpha_{22} \geq 2$, $CE_{f \to j \mid c} \approx 0 \; \forall \; j = \{a, b, c, d, f\}$ and f^{th}, b^{th} and c^{th} system belong to the same cluster.

To sum up, we determined clusters in the network by estimating the CE of the systems. The CE measure can effectively distinguish the systems forming synchronization clusters in the network in case of varying coupling strengths.

CHAPTER EIGHT
CONCLUSION

Complex networks are everywhere. Studying complex networks attracts considerable interest in various scientific disciplines, from economics to biology, mechanics to electricity. In particular, various studies of complex networks are combined with the synchronization phenomenon. The individual node dynamics of a network can be chaotic in a synchronous network. The coupled chaotic dynamical systems are particularly useful to model and understand complex networks.

Underlying dynamical systems' models in coupled networks are unknown in general. Preliminary information about the coupling connections and the coupling strengths are barely known.

The main goal of this thesis is to reveal the network structure by gathering the observations of a subset of systems with predefined restrictions. Inferring network structure in synchronized chaotic networks can be possible using the information theory as a model-free approach. The information flow between the subsystems is evaluated through information measures. If the information measures are estimated correctly, the information flow of a network can be represented better, and the unknown desired network properties can emerge.

Motivated by this issue, we have represented a framework of the information theory measures and their estimators in the literature. The performance of histogram, KDE, kNN entropy estimator, and kpN entropy estimator have been investigated for the MI and the TE of linear and nonlinear systems. The TE of a real dataset has also been estimated via these methods. We have discussed the statistical significance of the estimation methods of TE by using a hypothesis test. kNN estimator has a considerable accuracy and a speed for estimation; therefore, we have utilized kNN estimator to measure the information flow between the subsystems in the network.

There are several methods and algorithms that allow us to understand the complete and cluster synchronization phenomena, as well as a variety of measures that

determine information flow in discrete-time maps. However, no unique substantial measure quantifies information flow in complete, and cluster synchronization in the continuous-time coupled chaotic networks extensively.

The main goal of this thesis is to reveal the unknown coupling connections in directed complete synchronized networks from the observations under the assumptions that underlying dynamics or coupling strength is unknown. We have also assumed that the individual members of the network are identical; the coupling strength is the same as for all members; the input can be added to each system at specific times. Moreover, one single output of the individual node systems is observed.

Under these assumptions, we have perturbed each system at specific times to destroy synchronization that allows us to obtain valuable information about the network connections. Then, we have reconstructed the state space according to Takens' theorem that ensures the TE or CE of the systems in the reconstructed space is the same as the original state space. We analytically showed the differences of AICEs between the reconstructed outputs obtained after injecting random information into the system, which can be used to distinguish the paths. Then, we have presented the theorems on AICEs that yield information about the graph's topology. We have extended the proposed theorems to develop an algorithm to reconstruct the adjacency matrix of a given network. We have used the proposed algorithm on randomly coupled chaotic networks in complete synchronization. The proposed algorithm correctly finds the adjacency matrix and reconstructs the directed network en route to re-synchronization.

As a second goal, we have dealt with detecting the clusters in cluster synchronization using observations from a single output. While we know the number of systems in the network, the number of clusters, the connectivity matrix, and coupling strength are unknown. Besides, the clusters are not necessarily directly coupled. The proposed AICEs between the systems have also been estimated in the reconstructed state space by following the same procedure in the previous approach.

We have classified the AICEs with the k-means algorithm. An original algorithm has been presented for the detection of clusters via this approach. We have demonstrated the results in cluster synchronization networks. Unsurprisingly, the clusters and their membership can be effectively found using the proposed algorithm.

The contributions of this thesis are summarized as follows:

i) We have suggested a kNN-based CE estimator based on the existing estimator for efficient estimation.

ii) We have shown that TE or CE measures are preserved under the homeomorphism so that we can estimate the necessary measures by reconstructing state space via observations of a single output from each system.

iii) We have proposed the AICE measure based on CE to represent the information flow.

iv) We have derived the analytical properties of the difference of AICE that enables us to distinguish the paths in directed networks.

v) An algorithm that reveals the paths of the network and finds the adjacency matrix has been presented.

vi) The proposed AICE has also been used for distinguishing the clusters in the network.

vii) A new algorithm that computes necessary AICEs with a k-means clustering algorithm has been developed.

viii) The detection of cluster pattern change using the CE has been shown.

The synchronization of a chaotic network via information theory enlightens us to understand the complex networks and causes us to ask many open research questions.

8.1 Future Work

One important open question is the determination of connectivity in bidirectional networks given the observations from the node outputs. That problem is more challenging because the directionality of information can not be inferred directly, especially in networks that contain loops. As of yet, we have not investigated the role of AICE in the networks with bidirectional couplings.

Cluster synchronization is of vital importance for modeling complex networks. In the light of current studies, the network reconstruction of the cluster synchronization for directional or bidirectional networks has not been investigated yet.

The proposed AICE measure is promising to allow us to detect the coupling connections in cluster synchronization. In this respect, a more generalized algorithm can be constructed that takes complete synchronization as a particular case.

Interest in the efficient estimation of entropy-related measures is relatively new. We believe a more efficient and faster estimation of the measures will prove helpful in carrying out the proposed analysis in real-time and further extend entropy's role in signal analysis and control applications.

REFERENCES

Abarbanel, H. (2012). *Analysis of observed chaotic data*. Springer Science & Business Media.

Alderisio, F., Fiore, G., & Bernardo, M. d. (2017). Reconstructing the structure of directed and weighted networks of nonlinear oscillators. *Physical Review E*, *95*(4), 042302. https://doi.org/10.1103/PhysRevE.95.042302.

Arnold, V. I. (1992). *Ordinary differential equations*. Springer Science & Business Media.

Belykh, I., Hasler, M., Lauret, M., & Nijmeijer, H. (2005). Synchronization and graph topology. *International Journal of Bifurcation and Chaos*, *15*(11), 3423–3433. https://doi.org/10.1142/S0218127405014143.

Belykh, I. V., Belykh, V. N., & Hasler, M. (2006). Generalized connection graph method for synchronization in asymmetrical networks. *Physica D: Nonlinear Phenomena*, *224*(1), 42–51. https://doi.org/10.1016/j.physd.2006.09.014.

Belykh, V. N., Belykh, I. V., & Hasler, M. (2000). Hierarchy and stability of partially synchronous oscillations of diffusively coupled dynamical systems. *Physical Review E*, *62*(5), 6332. https://doi.org/10.1103/physreve.62.6332.

Belykh, V. N., Belykh, I. V., & Hasler, M. (2004). Connection graph stability method for synchronized coupled chaotic systems. *Physica D: Nonlinear Phenomena*, *195*(1-2), 159–187. https://doi.org/10.1016/j.physd.2004.03.012.

Belykh, V. N., Belykh, I. V., Hasler, M., & Nevidin, K. V. (2003). Cluster synchronization in three-dimensional lattices of diffusively coupled oscillators. *International Journal of Bifurcation and Chaos*, *13*(04), 755–779. https://doi.org/10.1142/S0218127403006923.

Belykh, V. N., Osipov, G. V., Petrov, V. S., Suykens, J. A., & Vandewalle, J. (2008). Cluster synchronization in oscillatory networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *18*(3), 037106. https://doi.org/10.1063/1.2956986.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the Association for Computing Machinery*, *18*(9), 509–517. https://doi.org/10.1145/361002.361007.

Blekhman, I. (1964). The problem of synchronization of dynamical systems. *Journal of Applied Mathematics and Mechanics*, *28*(2), 239–265. https://doi.org/10.1016/0021-8928(64)90160-1.

Bollt, E. M. (2012). Synchronization as a process of sharing and transferring information. *International Journal of Bifurcation and Chaos*, *22*(11), 1250261. https://doi.org/10.1142/S0218127412502616.

Bossomaier, T., Barnett, L., Harré, M., & Lizier, J. T. (2016). *An introduction to transfer entropy*. Springer Science & Business Media.

Canlı, Ö., & Günel, S. (2020). Can we detect clusters of chaotic dynamical networks via causation entropy? *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *30*(6), 063127. https://doi.org/10.1063/1.5139695.

Casdagli, M., Eubank, S., Farmer, J. D., & Gibson, J. (1991). State space reconstruction in the presence of noise. *Physica D: Nonlinear Phenomena*, *51*(1), 52–98. https://doi.org/10.1016/0167-2789(91)90222-U.

Chen, L., Lu, J., & Tse, C. K. (2009). Synchronization: An obstacle to identification of network topology. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *56*(4), 310–314. https://doi.org/10.1109/TCSII.2009.2015381.

Chen, T., Liu, X., & Lu, W. (2007). Pinning complex networks by a single controller. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *54*(6), 1317–1326. https://doi.org/10.1109/TCSI.2007.895383.

Chua, L. O., Komuro, M., & Matsumoto, T. (1986). The double scroll family. *IEEE Transactions on Circuits and Systems*, *33*(11), 1072–1118. https://doi.org/10.1109/TCS.1986.1085869.

Cover, T. M., & Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Cruz, J. M., & Chua, L. O. (1992). A CMOS IC nonlinear resistor for Chua's circuit. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, *39*(12), 985–995. https://doi.org/10.1109/81.207719.

Deng, G. (2014). Pinning group synchronization in complex dynamical networks with different groups of oscillators. *Mathematical Problems in Engineering*, *2014*. https://doi.org/10.1155/2014/864193.

Duan, P., Yang, F., Chen, T., & Shah, S. L. (2013). Direct causality detection via the transfer entropy approach. *IEEE Transactions on Control Systems Technology*, *21*(6), 2052–2066. https://doi.org/10.1109/TCST.2012.2233476.

Erdős, P., Rényi, A., et al. (1960). On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, *5*(1), 17–60.

Eroglu, D., Lamb, J. S., & Pereira, T. (2017). Synchronisation of chaos and its applications. *Contemporary Physics*, *58*(3), 207–243. https://doi.org/10.1080/00107514.2017.1345844.

Fortuna, L., Frasca, M., & Xibilia, M. G. (2009). *Chua's circuit implementations: Yesterday, today and tomorrow*. World Scientific.

Frasca, M., Gambuzza, L. V., Buscarino, A., & Fortuna, L. (2018). *Synchronization in networks of nonlinear circuits: Essential topics with MATLAB® code*. Springer.

Frenzel, S., & Pompe, B. (2007). Partial mutual information for coupling analysis of multivariate time series. *Physical Review Letters*, *99*(20), 204101. https://doi.org/10.1103/PhysRevLett.99.204101.

Fujisaka, H., & Yamada, T. (1983). Stability theory of synchronized motion in coupled-oscillator systems. *Progress of Theoretical Physics*, *69*(1), 32–47.

Gambuzza, L. V., Frasca, M., & Latora, V. (2019). Distributed control of synchronization of a group of network nodes. *IEEE Transanctions Automatic Control*, *64*(1), 362–369. https://doi.org/10.1109/TAC.2018.2828780.

Gambuzza, L. V., Frasca, M., Sorrentino, F., Pecora, L. M., & Boccaletti, S. (2020). Controlling symmetries and clustered dynamics of complex networks. *IEEE Transactions on Network Science and Engineering*, *8*(1), 282–293. https://doi.org/10.1109/TNSE.2020.3037039.

Gao, S., Ver Steeg, G., & Galstyan, A. (2015). Efficient estimation of mutual information for strongly dependent variables. In *Artificial Intelligence and Statistics*, 277–286.

Gencaga, D., Knuth, K. H., & Rossow, W. B. (2015). A recipe for the estimation of information flow in a dynamical system. *Entropy*, *17*(1), 438–470. https://doi.org/10.3390/e17010438.

Good, P. (2005). *Permutation, parametric and bootstrap tests of hypotheses*. Springer Science & Business Media.

Granger, C. W. (1988). Some recent development in a concept of causality. *Journal of Econometrics*, *39*(1-2), 199–211.

Gray, A. G., & Moore, A. W. (2003). Nonparametric density estimation: Toward computational tractability. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, *SIAM*, 203–211. https://doi.org/10.1137/1.9781611972733.19.

Haaga, K. A., Diego, D., Brendryen, J., & Hannisdal, B. (2021, May 28). *A simple test for causality in complex systems*. Arxiv. https://arxiv.org/abs/2005.01860.

Hagberg, A., Swart, P., & S Chult, D. (2022, April 06). *NetworkX: Network analysis in Pythor*. NetworkX. https://networkx.org/.

Hagberg, A., Swart, P., & S Chult, D. (2022, August 02). *Exploring network structure, dynamics, and function using networkx*. Osti. https://www.osti.gov/biblio/960616.

Hartigan, J. A. (1975). *Clustering algorithms*. John Wiley & Sons.

Hasler, M. (1995). Engineering chaos for encryption and broadband communication. *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences*, *353*(1701), 115–126. https://doi.org/10.1098/rsta.1995.0094.

Hasler, M., Maistrenko, Y., & Popovych, O. (1998). Simple example of partial synchronization of chaotic systems. *Physical Review E*, *58*(5), 6843. https://doi.org/10.1103/PhysRevE.58.6843.

Hlinka, J., Hartman, D., Vejmelka, M., Runge, J., Marwan, N., Kurths, J., & Paluš, M. (2013). Reliability of inference of directed climate networks using conditional mutual information. *Entropy*, *15*(6), 2023–2045. https://doi.org/10.3390/e15062023.

Huygens, C. (1673). *The Pendulum clock*. Iowa State University Press.

Jones, E., Oliphant, T., Peterson, P., et al. (2019, October 07). *SciPy: Open source scientific tools for Python*. Scipy. http://www.scipy.org/.

Kantz, H., & Schreiber, T. (2004). *Nonlinear time series analysis*. Cambridge university press.

Kloeden, P. E., & Platen, E. (2013). *Numerical solution of stochastic differential equations*. Springer Science & Business Media.

Kozachenko, L., & Leonenko, N. N. (1987). Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, *23*(2), 9–16.

Krämer, K.-H., Datseris, G., Kurths, J., Kiss, I. Z., Ocampo-Espindola, J. L., & Marwan, N. (2021). A unified and automated approach to attractor reconstruction. *New Journal of Physics*, *23*(3), 033017. https://doi.org/10.1088/1367-2630/abe336.

Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical Review E*, *69*(6), 066138. https://doi.org/10.1103/PhysRevE.69.066138.

Kuzmanovski, V., Todorovski, L., & Džeroski, S. (2018). Extensive evaluation of the generalized relevance network approach to inferring gene regulatory networks. *GigaScience*, *7*(11), giy118. https://doi.org/10.1093/gigascience/giy118.

Latora, V., Nicosia, V., & Russo, G. (2017). *Complex networks: Principles, methods and applications*. Cambridge University Press.

Leguia, M. G., Levnajić, Z., Todorovski, L., & Ženko, B. (2019). Reconstructing dynamical networks via feature ranking. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *29*(9), 093107. https://doi.org/10.1063/1.5092170.

Liu, H., Li, Y., Li, Z., Lü;, J., & Lu, J.-A. (2022). Topology identification of multilink complex dynamical networks via adaptive observers incorporating chaotic exosignals. *IEEE Transactions on Cybernetics*, *52*(7), 6255–6268. https://doi.org/10.1109/TCYB.2020.3042223.

Liu, X., & Chen, T. (2011). Cluster synchronization in directed networks via intermittent pinning control. *IEEE Transactions on Neural Networks*, *22*(7), 1009–1020. https://doi.org/10.1109/TNN.2011.2139224.

Liu, X., & Li, S. (2017). Cluster synchronization for linearly coupled nonidentical systems with delays via aperiodically intermittent pinning control. *IEEE Access*, *5*, 4179–4189. https://doi.org/10.1109/ACCESS.2017.2681800.

Lizier, J. T., Heinzle, J., Horstmann, A., Haynes, J.-D., & Prokopenko, M. (2011). Multivariate information-theoretic measures reveal directed information structure and task relevant changes in fMRI connectivity. *Journal of Computational Neuroscience*, *30*(1), 85–107. https://doi.org/10.1007/s10827-010-0271-2.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129–137. https://doi.org/10.1109/TIT.1982.1056489.

Lombardi, D., & Pant, S. (2016). Nonparametric k-nearest-neighbor entropy estimator. *Physical Review E*, *93*(1), 013310. https://doi.org/10.1103/PhysRevE.93.013310.

Lord, W. M., Sun, J., & Bollt, E. M. (2018). Geometric k-nearest neighbor estimation of entropy and mutual information. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *28*(3), 033114. https://doi.org/10.1063/1.5011683.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, *20*(2), 130–141.

Lorenz, E. N., & Haman, K. (1996). The essence of chaos. *Pure and Applied Geophysics*, *147*(3), 598–599.

Lou, X., & Suykens, J. A. (2011). Finding communities in weighted networks through synchronization. *Chaos*, *21*(4), 043116. https://doi.org/10.1063/1.3655371.

Lu, W., Liu, B., & Chen, T. (2010). Cluster synchronization in networks of coupled nonidentical dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *20*(1), 013120. https://doi.org/10.1063/1.3329367.

Ma, Z., Liu, Z., & Zhang, G. (2006). A new method to realize cluster synchronization in connected chaotic networks. *Chaos*, *16*(2), 023103. https://doi.org/10.1063/1.2184948.

Mooij, J. M., Peters, J., Janzing, D., Zscheischler, J., & Schölkopf, B. (2016). Distinguishing cause from effect using observational data: methods and benchmarks. *The Journal of Machine Learning Research*, *17*(1), 1103–1204. https://doi.org/10.5555/2946645.2946677.

Morgül, Ö. (1995). Inductorless realisation of Chua oscillator. *Electronics Letters*, *31*(17), 1403–1404. https://doi.org/10.1049/el:19950975.

Nishikawa, T., & Motter, A. E. (2016). Network-complement transitions, symmetries, and cluster synchronization. *Chaos*, *26*(9), 094818. https://doi.org/10.1063/1.4960617.

Novelli, L., & Lizier, J. T. (2021). Inferring network properties from time series using transfer entropy and mutual information: Validation of multivariate versus bivariate approaches. *Network Neuroscience*, *5*(2), 373–404. https://doi.org/10.1162/netn_a_00178.

Paluš, M., Komárek, V., Hrnčíř, Z., & Štěrbová, K. (2001). Synchronization as adjustment of information rates: Detection from bivariate time series. *Physical Review E*, *63*(4), 046211. https://doi.org/10.1103/PhysRevE.63.046211.

Panaggio, M. J., Ciocanel, M.-V., Lazarus, L., Topaz, C. M., & Xu, B. (2019). Model reconstruction from temporal data for coupled oscillator networks. *Chaos:*

An Interdisciplinary Journal of Nonlinear Science, *29*(10), 103116. https://doi.org/10.1063/1.5120784.

Pecora, L. M., & Carroll, T. L. (1990). Synchronization in chaotic systems. *Physical Review Letters*, *64*(8), 821. https://doi.org/10.1016/S0370-1573(02)00137-0.

Pecora, L. M., & Carroll, T. L. (1991). Driving systems with chaotic signals. *Physical Review A*, *44*(4), 2374. https://doi.org/10.1103/PhysRevA.44.2374.

Pecora, L. M., & Carroll, T. L. (1998). Master stability functions for synchronized coupled systems. *Physical Review Letters*, *80*(10), 2109. https://doi.org/10.1103/PhysRevLett.80.2109.

Pecora, L. M., & Carroll, T. L. (2015). Synchronization of chaotic systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *25*(9), 097611. https://doi.org/10.1016/S0370-1573(02)00137-0.

Pecora, L. M., Sorrentino, F., Hagerstrom, A. M., Murphy, T. E., & Roy, R. (2014). Cluster synchronization and isolated desynchronization in complex networks with symmetries. *Nature Communications*, *5*, 4079. https://doi.org/10.1038/ncomms5079.

Pikovsky, A., Rosenblum, M., & Kurths, J. (2002). *Synchronization: A universal concept in nonlinear science*. Cambridge University Press.

Pogromsky, A., Santoboni, G., & Nijmeijer, H. (2002). Partial synchronization: from symmetry towards stability. *Physica D: Nonlinear Phenomena*, *172*(1-4), 65–87. https://doi.org/10.1016/S0167-2789(02)00654-1.

Porfiri, M., & Marín, M. R. (2017). Information flow in a model of policy diffusion: an analytical study. *IEEE Transactions on Network Science and Engineering*, *5*(1), 42–54. https://doi.org/10.1109/TNSE.2017.2731212.

Rosenblum, M. G., Pikovsky, A. S., & Kurths, J. (1996). Phase synchronization of chaotic oscillators. *Physical Review Letters*, *76*(11), 1804. https://doi.org/10.1103/PhysRevLett.76.1804.

Rosenblum, M. G., Pikovsky, A. S., & Kurths, J. (1997). From phase to lag synchronization in coupled chaotic oscillators. *Physical Review Letters*, *78*(22), 4193. https://doi.org/10.1103/PhysRevLett.78.4193.

Rulkov, N. F., Sushchik, M. M., Tsimring, L. S., & Abarbanel, H. D. (1995). Generalized synchronization of chaos in directionally coupled chaotic systems. *Physical Review E*, *51*(2), 980. https://doi.org/10.1103/PhysRevE.51.980.

Schaub, M. T., O'Clery, N., Billeh, Y. N., Delvenne, J.-C., Lambiotte, R., & Barahona, M. (2016). Graph partitions and cluster synchronization in networks of oscillators. *Chaos*, *26*(9), 094821. https://doi.org/10.1063/1.4961065.

Schreiber, T. (2001). Measuring information transfer. *Physical Review Letters*, *85*(2), 461–464. https://doi.org/10.1103/PhysRevLett.85.461.

Shanmugan, K. S., & Breipohl, A. M. (1988). *Random signals, detection, estimation and data analysis*. John Wiley & Sons.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, *27*(3), 379–423.

Singh, H., Misra, N., Hnizdo, V., Fedorowicz, A., & Demchuk, E. (2003). Nearest neighbor estimates of entropy. *American Journal of Mathematical and Management Sciences*, *23*(3-4), 301–321. https://doi.org/10.1080/01966324.2003.10737616.

Soong, T. T. (1973). *Random differential equations in science and engineering*. Elsevier.

Sorrentino, F., Pecora, L. M., Hagerstrom, A. M., Murphy, T. E., & Roy, R. (2016). Complete characterization of the stability of cluster synchronization in complex dynamical networks. *Science Advances*, *2*(4), e1501737. https://doi.org/10.1126/sciadv.1501737.

Spinney, R. E., Prokopenko, M., & Lizier, J. T. (2017). Transfer entropy in continuous time, with applications to jump and neural spiking processes. *Physical Review E*, *95*(3), 032319. https://doi.org/10.1103/PhysRevE.95.032319.

Stetter, O., Battaglia, D., Soriano, J., & Geisel, T. (2012). Model-free reconstruction of excitatory neuronal connectivity from calcium imaging signals. *PLOS Computational Biology*, *8*(8), 1–25. https://doi.org/10.1371/journal.pcbi.1002653.

Strogatz, S. H. (2001). Exploring complex networks. *Nature*, *410*(6825), 268–276. https://doi.org/10.1038/35065725.

Strogatz, S. H. (2018). *Nonlinear dynamics and chaos: With applications to physics, biology, chemistry, and engineering*. CRC Press.

Sun, J., & Bollt, E. M. (2014). Causation entropy identifies indirect influences, dominance of neighbors and anticipatory couplings. *Physica D: Nonlinear Phenomena*, *267*, 49–57. https://doi.org/10.1016/j.physd.2013.07.001.

Sun, J., Taylor, D., & Bollt, E. M. (2015). Causal network inference by optimal causation entropy. *SIAM Journal on Applied Dynamical Systems*, *14*(1), 73–106. https://doi.org/10.1137/140956166.

Takens, F. (1981). Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, 366–381.

Timme, M. (2007). Revealing network connectivity from response dynamics. *Physical Review Letters*, *98*(22), 224101. https://doi.org/10.1103/PhysRevLett.98.224101.

Timme, M., & Casadiego, J. (2014). Revealing networks from dynamics: an introduction. *Journal of Physics A: Mathematical and Theoretical*, *47*(34), 343001. https://doi.org/10.1088/1751-8113/47/34/343001.

Uzal, L. C., Grinblat, G. L., & Verdes, P. F. (2011). Optimal reconstruction of dynamical systems: A noise amplification approach. *Physical Review E*, *84*(1), 016223. https://doi.org/10.1103/PhysRevE.84.016223.

Vicente, R., Wibral, M., Lindner, M., & Pipa, G. (2011). Transfer entropy—a model-free measure of effective connectivity for the neurosciences. *Journal of Computational Neuroscience*, *30*(1), 45–67. https://doi.org/10.1007/s10827-010-0262-3.

Wand, M. P., & Jones, M. C. (1994). *Kernel smoothing*. Chapman & Hall/CRC.

Wang, W.-X., Lai, Y., & Grebogi, C. (2016). Data based identification and prediction of nonlinear and complex dynamical systems. *Physics Reports*, *644*, 1–76. https://doi.org/10.1016/j.physrep.2016.06.004.

Wibral, M., Vicente, R., & Lindner, M. (2014). Transfer entropy in neuroscience. In *Directed information measures in neuroscience*, 3–36.

Wiggins, S. (2003). *Introduction to applied nonlinear dynamical systems and chaos*. Springer Science & Business Media.

Wu, C. W., & Chua, L. O. (1995). Synchronization in an array of linearly coupled dynamical systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, *42*(8), 430–447. https://doi.org/10.1109/81.404047.

Wu, R., & Jiang, L. (2021). Recovering dynamic networks in big static datasets. *Physics Reports*, *912*, 1–57. https://doi.org/10.1016/j.physrep.2021.01.003.

Wu, W., Zhou, W., & Chen, T. (2008). Cluster synchronization of linearly coupled complex networks under pinning control. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *56*(4), 829–839. https://doi.org/10.1109/10.1109/TCSI.2008.2003373.

Wu, X., Zhao, X., Lü, J., Tang, L., & Lu, J.-a. (2016). Identifying topologies of complex dynamical networks with stochastic perturbations. *IEEE Transactions on Control of Network Systems*, *3*(4), 379–389. https://doi.org/10.1109/TCNS.2015.2482178.

Wu, Z., & Fu, X. (2014). Cluster synchronization in community networks with nonidentical nodes via edge-based adaptive pinning control. *Journal of the Franklin Institute*, *351*(3), 1372–1385. https://doi.org/10.1016/j.jfranklin.2013.11.011.

Xiong, W., Faes, L., & Ivanov, P. C. (2017). Entropy measures, entropy estimators, and their performance in quantifying complex dynamics: Effects of artifacts,

nonstationarity, and long-range correlations. *Physical Review E*, *95*(6), 062114. https://doi.org/10.1103/PhysRevE.95.062114.

Yanchuk, S., Maistrenko, Y., & Mosekilde, E. (2001). Partial synchronization and clustering in a system of diffusively coupled chaotic oscillators. *Mathematics and Computers in Simulation*, *54*(6), 491–508. https://doi.org/10.1016/S0378-4754(00)00276-7.

Yu, D., & Parlitz, U. (2011). Inferring network connectivity by delayed feedback control. *PloS one*, *6*(9), e24333. https://doi.org/10.1371/journal.pone.0024333.

Yu, D., Righero, M., & Kocarev, L. (2006). Estimating topology of networks. *Physical Review Letters*, *97*(18), 188701. https://doi.org/10.1103/PhysRevLett.97.188701.

Zhu, J., Bellanger, J.-J., Shu, H., & Le Bouquin Jeannès, R. (2015). Contribution to transfer entropy estimation via the k-nearest-neighbors approach. *Entropy*, *17*(6), 4173–4201. https://doi.org/10.3390/e17064173.