Cognitive Control in Cognitive Dynamic Systems and Networks

COGNITIVE CONTROL IN COGNITIVE DYNAMIC SYSTEMS AND NETWORKS

BY

SEYED MEHDI FATEMI BOOSHEHRI, M.Sc. (Computational Science),

Memorial University, St. John's, NL, Canada

A THESIS

SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2014) McMaster University

(School of Computational Science and Engineering) Hamilton, Ontario, Canada

TITLE:	Cognitive Control in Cognitive Dynamic Systems and Networks
AUTHOR:	Seyed Mehdi Fatemi Booshehri M.Sc. (Computational Science), Memorial University, St. John's, NL, Canada
SUPERVISOR:	Prof. Simon Haykin
NUMBER OF PAGES:	xiv, 152

To my family

Abstract

The main idea of this thesis is to define and formulate the role of cognitive control in cognitive dynamic systems and complex networks in order to control the directed flow of information. A cognitive dynamic system is based on Fuster's principles of cognition, the most basic of which is the so-called global perception-action cycle, that the other three build on. Cognitive control, by definition, completes the executive part of this important cycle. In this thesis, we first provide the rationales for defining cognitive control in a way that it suits engineering requirements. To this end, the novel idea of entropic state and thereby the two-state model is first described. Next, on the sole basis of entropic state and the concept of directed information flow, we formulate the learning algorithm as the first process of cognitive control. Most importantly, we show that the derived algorithm is indeed a special case of the celebrated Bellman's dynamic programming. Another significant key point is that cognitive control intrinsically differs from the generic dynamic programming and its approximations (commonly known as reinforcement learning) in that it is stateless by definition. As a result, the main two desired characteristics of the derived algorithm are described as follows: a) it is convergent to optimal policy, and b) it is free of curse of dimensionality.

Next, the predictive planning is described as the second process of cognitive control. The planning process is on the basis of shunt cycles (called mutually composite cycles herein) to bypass the environment and facilitate the prediction of future global perception-action cycles. Our results demonstrate predictive planning to have a very significant improvement to the functionality of cognitive control. We also deploy the explore/exploit strategy in order to apply a simplistic form of executive attention.

The thesis is then expanded by applying cognitive control into two different applications of practical importance. The first one involves cognitive tracking radar, which is based on a benchmark example and provides the means for testing the theory. In order to have a frame of reference, the results are compared to other cognitive controllers, which use traditional Q-learning and the method of dynamic optimization. In both cases, the new algorithm demonstrates considerable improvement with less computational load.

For the second application, the problem of observability in stochastic complex networks has been picked due to its importance in many practical situations. Having known cognitive control theory and its significant performance, the idea here is to view the network as the environment of a cognitive dynamic system; thereby, cognitive dynamic system with the cognitive controller plays a supervisory role over the network. The proposed methodology differs from the state-of-the-art in the literature in two accounts: 1) stochasticity both in modelling as well as monitoring processes, and 2) complexity in terms of edge density. We present several examples to demonstrate the information processing power of cognitive control in this context too.

The thesis will finish by drawing line for future research in three main directions.

Acknowledgements

Ph.D. is a journey, perhaps the most important one in academic life. Along this journey, being with a high-profile, well-known and well-respected supervisor is an asset and also a challenge at the same time. At this stage that I am becoming very close to the ultimate point, if I look back to the past four years or so, I see many challenges that have been overcome not just by myself, but with the helps from him. I must say that I have never been alone all through this way. Therefore, I would express my deep appreciation to my supervisor, Prof. Simon Haykin, for all what I have learnt from him and for all he has done for me.

Next, I would like to thank the chair of McMaster School of Computational Science and Engineering, Prof. Bartosz Protas, for all his kind support. With such gentle and supportive leadership, I see the future of the school to be brighter for years to come. I would also wish to express my appreciation to the Government of Ontario for the OGS scholarship, the National Science and Engineering Council for the support of my first two years, and the US Steel Canada (Stelco) for their generous award, which all provided me with sources of support and motivation throughout my Ph.D. studies at McMaster.

I would also like to thank all my beloved friends whose presence has always been priceless. Being with such a gifted group of friends has alleviated the loneliness, which

is an inevitable part and parcel of an immigrant life.

Last but by no means least, I would like to thank my father Mahmood, my mother Firoozeh, my brother Mohammad, and my sister Elaheh for their unconditional love and consistent support through all the years, without which all my achievements have simply been impossible.

Abbreviations

CDS	Cognitive Dynamic System(s)
EKF	Extended Kalman Filter
HEKF	Hybrid Extended Kalman Filter
CKF	Cubature Kalman Filter
UKF	Unscented Kalman Filter
RMSE	Root-mean Squared Error
MSE	Mean Squared Error
DP	Dynamic Programming
RL	Reinforcement Learning
TD	Temporal Difference
PAC	Perception-action Cycle
ER	Erdős-Rényi
LSB	Liu-Slotine-Barabasi method
SCC	Strongly Connected Component
ODE	Ordinary Differential Equation
IEEE	Institute of Electrical and Electronics Engineers
NSERC	National Science and Engineering Research Council

Contents

Abstract **iv**

Acknowledgements **vi**

Abbreviations **viii**

1 Introduction **2**

1.1 Theme and Objectives of Dissertation 2

1.2 Summary of Enclosed Articles 3

1.2.1 Paper I (Chapter II) 3

1.2.2 Paper II (Chapter III) 4

1.2.3 Paper III (Chapter IV) 5

1.3 Background and Coverage 5

1.3.1 Cognitive Dynamic Systems 6

1.3.2 Relevant Information about the Environment: The State Space Model 7

1.3.3 Entropic State of Perceptor: The Two-state Model 8

1.3.4 Why Cognitive Control? 9

1.3.5 Two Applications of Practical Importance 10

1.4 Scope of Research 11

2 Cognitive Control **13**

2.1 Abstract 13

2.2 Introduction 14

2.3 Control of Directed Information Flow 17

2.4 How Do We Define Cognitive Control? 25

2.5 The Two-state Model 28

2.6 Reinforcement Learning 30

2.7 Compositional Structure of Cognitive Control 35

2.7.1 Cognitive Control Integrated Inside CDS 36

2.7.2 RL in Cognitive Control 39

2.8 Computational Experiment on Cognitive Control 42

2.8.1 Experiment 1: Sub-optimality for reduced computational complexity 43

2.8.2 Experiment 2: Information-processing power of planning . . . 45

2.9 Concluding Remarks 47

2.9.1 Summarizing Highlights of the Paper 47

2.9.2 Comparison of Cognitive Control versus Adaptive Control and Neurocontrol 48

3 Cognitive Control: Theory and Application **51**

3.1 Abstract 51

3.2 Introduction 52

3.3 Cognitive Control 55

3.3.1 The Two-state Model 55
3.3.2 Cyclic Directed Information Flow 57
3.4 Formalism of The Learning Process in Cognitive Control 59
3.5 Cognitive Control Learning Algorithm Viewed as a Special Case of Bellman's Dynamic Programming 64
3.6 Optimality vs. Convergence-rate in Online Implementation 69
3.7 Formalism of the Planning Process in Cognitive Control 70
3.8 Explore/exploit Tradeoff for Cognitive Control 74
3.9 Structural Composition of the Cognitive Controller 76
3.10 Computational Experiment: Cognitive Tracking Radar 79
3.11 Conclusion 85
3.11.1 Cognitive Processing of Information 85
3.11.2 Linearity, Convergence, and Optimality 86
3.11.3 Engineering Application 87
3.12 Appendix A 89
3.13 Appendix B 90

4 Improving Observability of Stochastic Complex Networks under the Supervision of Cognitive Dynamic Systems 94

4.1 Abstract 94
4.2 Introduction 95
4.3 Brief Account on Network Science 99
4.3.1 Networks with Stochastic Dynamics 99
4.3.2 Two Basic Network Topologies of Practical Importance: 103
4.4 Observability of Stochastic Complex Networks 104

4.5 Complex Networks Viewed as the Environment of Cognitive Dynamic Systems . . . 109
4.5.1 Bayesian Perception of Networks: The Two-state Model . . . 111
4.5.2 Cyclic Directed Information Flow . . . 113
4.5.3 Summary of Cognitive Control . . . 114
4.6 Computational Experiments . . . 116
4.7 Summary and Discussion . . . 124

5 Conclusion **133**

5.1 Research Summary . . . 133
5.1.1 List of contributions . . . 133
5.1.2 Significance of the Research . . . 134
5.2 Future Research . . . 136
5.2.1 Topic I: Hierarchical Structures . . . 136
5.2.2 Topic II: The Impact of Cognitive Control on Risk Control . . . 137
5.2.3 Topic III: Information Supervisory of Real-world Complex Networks . . . 138

List of Figures

2.1 Schematic illustration of the information gap 22

2.2 Schematic structure of a cognitive control system 34

2.3 Computational experiment of Case Study 1. 44

2.4 Decreasing the entropic-state in a target tracking example 46

3.1 Block diagram of the global perception-action cycle 58

3.2 State transition in dynamic programming 67

3.3 Combined presence of feedback and feedforward information links . . 72

3.4 Cognitive control: Cyclic directed information flow and algorithmic process 77

3.5 The impact of planning on cognitive control in Scenario 1. 87

3.6 Comparative performance evaluation of the three different algorithms 88

4.1 Global perception-action cycle over a network 110

4.2 The network in example 1 117

4.3 Stochastic network observability without cognitive control (example 1) 127

4.4 Stochastic network observability with cognitive control (example 1) . 128

4.5 Histogram of the selected nodes in example 1 129

4.6 Stochastic observability in various configurations of Erdős-Rényi (ER) and scale-free networks 130

4.7 Graphical illustration of the network in example 3 131
4.8 Histogram of the redundant nodes in example 3 132

Declaration of Academic Achievement

This research presents analytical and computational work carried out solely by Seyed Mehdi Fatemi Booshehri, herein referred to as "the author," with advice and guidance provided by the academic supervisor Prof. Simon Haykin. Information that is presented from outside sources which has been used towards analysis or discussion, has been cited when appropriate, all other materials are the sole work of the author.

Chapter 1

Introduction

1.1 Theme and Objectives of Dissertation

In compliance with the terms and regulations of McMaster University, this dissertation has been assembled into a *sandwich thesis* format comprised of three journal articles. These articles represent the independent work of the author of this dissertation, Seyed Mehdi Fatemi Booshehri, henceforth referred to as "the author."

The articles in the dissertation follow a cohesive theme with a nice flow aimed at defining and expanding upon the current knowledge of cognitive control in cognitive dynamic systems and its practical applications. The general theme is based on the following:

i) To provide an overview and critical review of the related literature (Paper I).

ii) To introduce basic concepts and a rationale definition, which matches practical needs (Paper I and Paper II).

iii) To mathematically formulate the algorithmic processes of cognitive control with

desirable practical properties (Paper II).

iv) To computationally implement the formalized method in a challenging example pertaining to cognitive radar systems (Paper II).

v) To expand on the introduced paradigm of cognitive control and implement it as the information supervisor over stochastic complex networks, which is a completely different application of practical importance (Paper III).

In addition to the comprehensive literature review presented in Paper I, explanation of basic concepts with proper citations from related literature are also provided throughout the three papers. The published works contained in this dissertation invariably contain some overlap with regards to their coverage of relevant literature as well some aspects related to cognitive control itself. To address this overlap all the references are collected in a unified manner at the end of this thesis. Additionally, Section 1.3 is also dedicated to provide a cohesive overview of the entire thesis.

1.2 Summary of Enclosed Articles

The papers enclosed in this thesis are listed as follows:

1.2.1 Paper I (Chapter II)

Haykin, S., Fatemi M., Setoodeh, P., and Xu, Y.

Cognitive Control, *Proc. IEEE*, **100**(12), 3156–3169, December 2012.

Preface: This paper includes a comprehensive literature review of related fields to the new area of *cognitive control*. Most importantly, the concept of entropic state and

the engineering definition for cognitive control are discussed in this paper. The paper also includes a first-stage computational experiment involving the use of reinforcement learning in cognitive control, which makes the basis for the second paper.

1.2.2 Paper II (Chapter III)

Fatemi M., Haykin, S.

Cognitive Control: Theory and Application, *IEEE Access*, **2**, 698–710, June 2014.

Preface: After the literature review and conceptual definition of cognitive control in the first paper, this second paper mathematically formulates cognitive control for the first time. The formalism is on the basis of two basic concepts: the two-state model, and the entropic state of the perceptor, both of which are defined in Paper I. It is then proven in Paper II that the newly derived executive learning algorithm for cognitive control is indeed a special case of Bellman's dynamic programming; hence, the inheritance of dynamic programming's basic properties including convergence to optimal value. More importantly, the presented algorithm for cognitive control is *stateless* as opposed to both dynamic programming and the traditional reinforcement learning. The end result of this proposition is that the cognitive control learning algorithm becomes free of the so-called curse of dimensionality. Equally importantly, this paper discusses predictive planning as the second process of cognitive control as well as the explore/exploit tradeoff to improve the efficiency. The paper finishes with a benchmark computational experiment, which is built on the experiment presented in Paper I as well as the Q-learning and *dynamic optimization* methods for comparison.

1.2.3 Paper III (Chapter IV)

Fatemi M., Setoodeh, P., and Haykin, S.

Improving Observability of Stochastic Complex Networks under the Supervision of Cognitive Dynamic Systems, *IEEE Transactions on Network Science and Engineering*, paper submitted, October 25, 2014.

Preface: This paper expands on the results of the first two papers and provide a generic paradigm for improving observability in stochastic complex networks. Paper III first discusses the state-of-the-art for addressing the observability problem. In particular, it provides a discussion on the shortcomings of the state-of-the-art regarding stochasticity and complexity in terms of edge density. Next, it proposes to use the cognitive dynamic system paradigm, embodying Bayesian filtering and cognitive control, as the supervisor over complex stochastic networks. The results demonstrate the fact that the proposed paradigm provides a consistent and flexible technique for addressing the observability of stochastic complex networks under any desirable constraints, including restrictions on monitor nodes.

1.3 Background and Coverage

In this section, a brief introductory account is provided on the materials that are covered in the next chapters involving the three papers. Because comprehensive literature reviews are presented within the scholarly journal articles, specifically in Paper I, this section is only intended to serve as a cohesive overview of relevant topics covered by this dissertation.

1.3.1 Cognitive Dynamic Systems

The idea of *cognitive dynamic systems*, first described in Haykin (2006a) and then expanded in more detail in Haykin (2012b), is inspired by the brain. Cognitive dynamic systems are engineered dynamic systems on the basis of Fuster's *principles of cognition* Fuster (2003), namely perception-action cycle, memory, attention, and intelligence. However, in this thesis, the centre of focus is to complete the perception-action cycle by introducing *cognitive control*, which is responsible for selecting best cognitive actions in each of the perception-action cycles.

In a generic sense, any cognitive dynamic system deals with an environment of interest, which contains a number of *hidden states*. The perception-action cycle begins with the perceptor processing the incoming environmental *observables*, followed by *feedback information* about the environment sent to the controller by the perceptor to set the stage for the controller to act on the environment. The action selected by the controller, which is called cognitive action, naturally produces changes in the amount of information that the environmental observables contain, which in turn, sets the stage for a second perception-action cycle, and so it goes on. This distinctive cyclic behaviour of cognitive dynamic systems is continued until we reach a point where further *information gain* about the environment is too small to be of practical value, assuming that the environment is stationary. The perception-action cycle, just described, is said to be of a *global* kind, in that it embodies the environment within itself. Furthermore, the controller is said to be cognitive controller in that it controls the directed information flow throughout the entire system. The following points are also noteworthy:

- **Closed-loop feedback system**: From systems control perspective, the so-called perception-action cycle results in a closed-loop feedback system, where the cognitive controller sees the environment indirectly via the eyes of perceptor.

- **Dynamic system**: The entire system as a whole is *dynamic* in that it contains "change" of the environment from each global cycle of perception-action to the next.

- **Stochasticity**: The ever presence of uncertainty is unavoidable both as part of observables as well as in the mathematical modelling of the environment. Hence, the study of cognitive dynamic systems always involves dealing with probabilistic properties and formulations.

1.3.2 Relevant Information about the Environment: The State Space Model

In traditional sense, an environment contains a number of "targets" of interest, the quantified conditions of which result in a number of hidden "states." In actual fact, if the states are given precisely, they provide sufficient information about the environment; hence, the states are the minimal *relevant information* about the environment. In reality, unfortunately, the states are not normally available to us; rather, a number of observables are given, which are being updated in a cyclic manner. As a result, an important part of any cognitive dynamic system is its *preceptor*; its ultimate role is to reconstruct the states from the observables. Hence, the end result of the perceptor is the state *posterior*, which is the probability distribution of states conditioned on observables.

1.3.3 Entropic State of Perceptor: The Two-state Model

Clearly, posterior's shape is of the essence because it measures how much *informative* the posterior is. Posterior shape is indeed a "qualitative" measure for the overall performance of the perceptor itself in addition to the amount relevant information in the observables. In many practical situations, one or more of the following issues are involved:

- observables may not be informative enough, and/or
- sensory structure may add unwanted noise to the observables, and/or
- mathematical modelling of the environment may be imperfect.

Any of these issues will cause the posterior to become less informative. Therefore, a "quantitative" measure of lack of information in the posterior is required, which is defined as the *entropic state* of the perceptor. In many practical cases, Shannon entropy (Shannon (1948)) corresponding to the posterior is the best choice for the entropic state. Defining the entropic state results in having a *two-state model*, which embodies two (sets of) states:

- The first one is the traditional *target or physical state*, pertaining to a target of interest in the environment.
- The second one is the *entropic state* of the perceptor, discussed in this section.

In the next section, we define cognitive control as the paradigm for controlling the second state.

1.3.4 Why Cognitive Control?

The two-state model is an essential element in the definition of cognitive control. Because there are two separate states (each of which may be a scalar or a vector of real values), one can define two separate control processes, which may exist hand in hand: (i) state control and (ii) cognitive control. In a related manner, the following two points are worth mentioning:

- The realm of control theory, as we know it today and for ever more, is about controlling the state of environment, which naturally embodies estimation theory as an integrated part of the system. Broadly speaking, the literature of control includes adaptive control, stochastic control, fuzzy control, intelligent control, and others, each of which has established a legacy of its own in the literature. Returning to the two-state model, the traditional control is therefore to control the first state in the two-state model.

- On the other hand, cognitive control in cognitive dynamic systems is about controlling the entropic state. In a related manner in words, the function of cognitive control is to control the directed information flow in the system on a global cyclic basis. Cognitive control is therefore a new contribution to the literature. It is also noteworthy that in practice, both state control and cognitive control may exist side by side, specially so when we mimic the brain.

1.3.5 Two Applications of Practical Importance

Cognitive Radar Systems

As our first testbed, we will look into cognitive tracking radar systems. This application is based on a benchmark example, which was previously introduced in Haykin *et al.* (2011), then elaborated in Haykin *et al.* (2012c). The significance and importance of this application is due to the fact that it can be generalized to any problem, in which sensor properties are adjustable from one cycle of perception-action to the next. Having the sensory adjustments as the cognitive actions of such problems, the cognitive control theory can then be utilized to maximize the relevant information that is available to the perceptor through sensory measurements.

Observability of Stochastic Complex Networks under Practical Constraints

Study of complex networks, called *network science*, has been accelerated recently mostly due the impact that connected networks have had on different aspects of today's life. One problem of practical importance is the so-called observability problem, which is how to reconstruct the state of a network on the sole basis of observing a minimal number of its nodes. The problem becomes even more critical when the network is very large and relatively dense. Additionally, in reality, the ever presence of uncertainty and modelling imperfection cause much more difficulties in finding a practical solution to the observability problem.

In this thesis, knowing that we have solved the cognitive control problem and it is now available to us as a practical tool, we will go another step forward to tackle the observability of stochastic complex networks. To this end, the network will play the role of environment for the cognitive dynamic system; in return, the cognitive

dynamic system is the information supervisor over the network. This novel way of thinking will help us to find the best set of monitor nodes on a cyclic basis. An important point to note here is that working on the observability of complex networks is a very recent problem (although observability is not a new issue in systems and control theory). As a result, even the recent prominent work reported in Liu *et al.* (2013) is rather simplistic, dealing with only deterministic and sparse networks. In a generic sense, Paper III will therefore pave the way to overcome shortcomings of the state-of-the-art in the literature, and it provides a systematic and flexible way to address the observability problem.

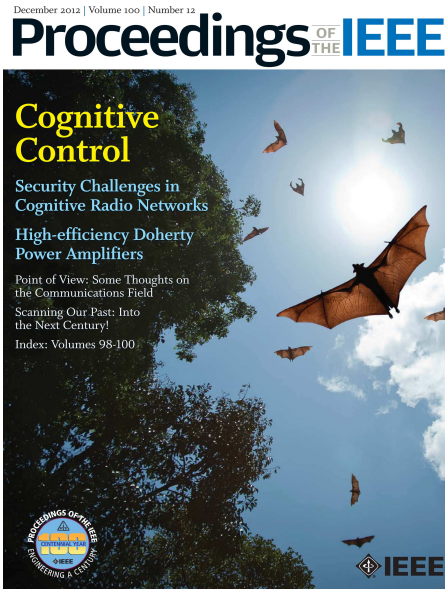
1.4 Scope of Research

Along the same line as the four annual reports to the committee and the research proposal of the author of this thesis, the research reported in this thesis is meant for defining, formulating, and implementing the new paradigm of cognitive control, as a new way of thinking. All the mathematical formulations, algorithmic design, and computational experiments are presented in the thesis. Thinking of cognition, the cognitive control paradigm is an intrinsic part of a perception-action cycle. However, the inclusion of other elements of cognition are outside the scope of this thesis and left for future work. More will be said on the future work in the final chapter of this thesis, specifically pertaining to Haykin *et al.* (2014), which involves the continuum of this thesis in the Cognitive Systems Laboratory at McMaster University.

The following chapter is a reproduction of an IEEE copyrighted, published paper*:

Haykin, S., Fatemi M., Setoodeh, P., and Xu, Y.

Cognitive Control, *Proc. IEEE*, **100**(12), 3156–3169, December 2012.



---

* The paper featured as the cover story of the issue.

Chapter 2

Cognitive Control

2.1 Abstract

This paper, entitled Cognitive Control, is inspired by how this same function manifests itself in the human brain and does so in a remarkable way.

After the Introduction, the paper addresses the many facets involved in the control of directed information flow in a dynamic system, culminating in the notion of information gap, defined as the difference between relevant information (useful part of what is extracted from the incoming measurements) and sufficient information representing the information needed for achieving minimal risk. The notion of information gap leads naturally to how cognitive control can itself be defined. Then, another important idea is described, namely the two-state model, in which one is the system's state and the other is the entropic state that provides an essential metric for quantifying the information gap. The entropic state is computed in the perceptual part (i.e., perceptor) of the dynamic system and sent to the controller directly as feedback information. This feedback information provides the cognitive controller

the information needed about the environment and the system to bring reinforcement leaning into play; reinforcement learning, incorporating planning as an integral part, is at the very heart of cognitive control. The stage is now set for a computational experiment, involving cognitive radar wherein the cognitive controller is enabled to control the receiver via the environment. The experiment demonstrates how reinforcement learning provides the mechanism for improved utilization of computational resources, and yet be able to deliver good performance through the use of planning. The paper finishes with concluding remarks.

2.2 Introduction

Much has been written on cognitive control in the neuroscience and psychology literature (see for example Mars *et al.* (2012) and Gardner *et al.* (1959)). In contrast, from an engineering perspective, cognitive control is in its very early stage of development. Looking back to the history of the field of control engineering in the twentieth century, we see a trend in the evolution of controllers from simple structures such as open-loop and proportional-integral-derivative (PID) controllers to much more sophisticated ones with features such as optimality, adaptivity, robustness, and intelligence to some extent.

Control systems are usually designed, based on a trade-off between optimality and robustness. In addition, it is desirable that the controller has the ability to change its behavior in accordance with new circumstances. Adaptive controllers and neurocontrollers have been proposed in the literature to address this issue. In adaptive control, the control problem is formulated in a way that the controller has some adjustable parameters. The controller is also equipped with an adaptation mechanism

for updating the parameters according to variations in dynamics of the system with which it interacts as well as the nature of disturbances. Adaptive control systems are inherently nonlinear due to the adaptation mechanism Aström and Wittenmark (1995); Landau *et al.* (2011); Sastry and Bodson (1989); Ioannou and Sun (1995).

While adaptive control is mainly based on parameterized mechanistic modeling, neurocontrollers are based on black-box modeling. Hybrid models (i.e. combination of mechanistic modeling and black-box modeling) can also be used, when finding mechanistic models is straight forward for some parts of the system and difficult for other parts. Control systems can benefit from neural networks in two different ways. One way is to implement the controller using a neural network. In this approach, the controller itself is a neural network; alternatively, the controller may not be a neural network but uses a neural network-based model of the system under study Nørgaard *et al.* (2000); Hrycej (1997); Lewis *et al.* (1998); Puskorius and Feldkamp (2001).

Typically, these controllers function well in structured environments and pre-specified conditions, for which they are designed. However, they will not function properly if the system of interest has unmodeled dynamics. In unstructured and/or highly uncertain environments, the presence of a human operator in the control loop is indispensable. In such environments, the controller often reaches points of surprise, for which it has not been programmed. This issue normally arises because the controller is unable to collect the sufficient information it requires to achieve its goals in a self-organized manner. Based on what is known in psychology and neuroscience, it appears that cognition is the needed functionality that should be built into control systems in order to reduce human intervention in the control loop. The article by Buss et al. is an endeavor to motivate the need for *cognitive control* in order to elevate

the use of automation to the next level Buss *et al.* (2011).

Cognitive control can be viewed as part of a more general framework, called *cognitive dynamic systems* (CDS) Haykin (2012a,b). The CDS theory is built on Fuster's paradigm of cognition, which states that a cognitive system, in its most general form, has five building blocks, namely the *perception-action cycle*, *memory*, *attention*, *intelligence*, and *language* Fuster (2003). The perception-action cycle is the backbone of any closed-loop feedback control system. It can be argued that an adaptive control system may well embody attention and intelligence as well, but lacks memory. Language is more relevant in the context of a network of cognitive agents.

A large percentage of the information processing in the brain is performed in the cortex and it plays a key role in processes attributed to cognition. Regarding the uniform appearance of the cortex, Mountcastle proposed that all regions of the cortex may use a basic information-processing algorithm to accomplish their tasks, regardless of the nature of the information-bearing sensory input. In other words, all kinds of sensory inputs (i.e. visual, auditory, etc) are coded in a standard form and fed to this basic processing algorithm Mountcastle (1998). Building on Mountcastle's theory, Fuster proposed the concept of *cognit* for knowledge representation in the cerebral cortex Fuster (2003).

The flow of information in our nervous system plays a critical role in sustaining our vital activities, performing our daily tasks and, even to some extent, determines who we are, especially so when it comes to memory formation. By the same token, the flow of information in man-made machines is of critical importance, regarding performance and robustness of the system. Therefore, controlling the flow of information deserves special attention in the study of cognitive control. Building on achievements of the

engineering and neuroscience communities for more than six decades, this paper is aimed at a new generation of control systems that are inspired by the human brain, hence the title: *cognitive control*.

The paper is organized as follows: Section II presents the tale of endeavors on directed information flow in control systems, which has led to the important concept of information gap. It is, in turn, related to the risk associated with an action or decision policy. Having the aim of reducing the information gap, Section III proposes a definition for cognitive control from an engineering perspective, with guidance from neuroscience. Another important notion, namely the two-state model, is then introduced in Section IV to take account of a quantitative measure of the information gap. Section V describes the reinforcement learning paradigm and its existence in mammalian brains in order to provide the background for Section VI, where the compositional structure of cognitive control is discussed. Section VII includes two computational experiments on cognitive control in a tracking radar system with emphasis on reinforcement learning and planning. Finally, Section VIII concludes the paper.

2.3 Control of Directed Information Flow

This section describes the endeavor of the engineering community to design systems with increased level of sophistication by looking into nature as the main source of inspiration. It is the story of evolution of ideas for more than half a century. It is the tale of *standing on the shoulders of giants* by building on well-established theories, modifying them to extend their applicability to new domains, revisiting them from new perspectives, and integrating them to form more general theoretical frameworks.

Adopting an interdisciplinary approach, after World War II, Wiener had come to the conclusion that the fields of control and communications are both centered around the notion of *information*, where *feedback* plays a key role in information manipulation and decision making Wiener (1965, 1950). To this end, he came up with the idea of inseparability of these two fields and tried to gather his own work on control and statistical signal processing Wiener (1964), Shannon's information theory Shannon (1948); Shannon and Weaver (1949), and Kolmogorov's work on prediction theory Shiryayev (1992a,b) under a unified umbrella. Wiener called this unifying framework *cybernetics*, which is rooted in the Greek word for *steersman*. As a result of Wiener's close collaboration with the engineer Bigelow and neurophysiologist Rosenblueth, the theory of cybernetics was based on the hypothesis that despite functional differences, machines and living organisms have similar behavioral mechanisms Wiener (1965); Seising (2008). Wiener also wished to highlight and draw attention to the similarities between the human nervous system on the one hand, and the computation and control in machines, on the other hand, to reach a new interpretation of man, man's knowledge of the universe, and society Wiener (1965). In light of these pioneering contributions of Wiener, Dupuy has justifiably argued that cognitive science has its roots in cybernetics Dupuy (2009).

Wiener learned much from the experience gained through working on anti-aircraft guns during World War II. In the beginning, human operators were responsible for gun-pointing, based on line-of-sight tracking of aircraft. Later on, this human-centered gun-pointing system was replaced with an automatic one by directly coupling a radar to the anti-aircraft gun. However, it would still not seem to be practical to completely remove the human operator from the control loop, especially when the

behavior of another human being (i.e. the enemy) needed to be counteracted. By increasing the speed and maneuverability of an aircraft, providing a degree of autonomy for directing the fire was indispensable. However, the system needed to predict the trajectory of the aircraft to make sure that the missile would hit the target at some point of time in the future. Wiener and Bigelow knew that both pilots and gunners would learn their opponent's pattern of behavior and, based on that behavior, improve their own performance over the course of time. To this end, they needed to understand how pilots and gunners were thinking, so as to design a system that would be able to somehow mimic human behavior Wiener (1965); Seising (2008).

Since feedback acts like a double-edged sword, Wiener and Bigelow noticed that as they were pushing for improving the performance of the system, it was possible for the system to become unstable and show oscillatory behavior. They wondered if a similar phenomenon had been observed regarding the nervous system of human beings. In other words, they wondered if there was any nervous-system disorder in which there was no sign of tremor at rest but during an action, the patient was starting to shake more and more severely till he/she could not perform the task. Rosenblueth's answer to this question was *intention tremors* associated with the cerebellum, which is responsible for controlling organized muscular activities. From this pathological resemblance, Wiener, Bigelow, and Rosenblueth concluded that intentional actions in both machines and living organisms can be explained with feedback. They also proposed a behavioral approach for studying systems. This approach is based on an abstract model of the system of interest, which determines the relationship between its input and output. In this abstract model, the output of the system represents any change it causes to the surrounding environment and its input represents the effect

of the external events on the system. In this context, feedback provides the means for information manipulation Wiener (1965); Seising (2008).

As mentioned before, Shannon's information theory is one of the pillars of cybernetics. Shannon's information theory was originally developed to mathematically formalize the transmission of signals through a communication channel. The theory provides a quantitative measure of the amount of information, which depends only on the probabilistic structure of the communication channel under study. Information theory has found diverse applications beyond just transmission and compression of data.

Howard emphasized that from a control or decision-making point-of-view, the probabilistic nature of uncertainties as well as their (economic) impacts on the decision-maker must be taken into account and a theory that only deals with probabilities of outcomes may not completely describe the importance of uncertainty to the decision-maker. When it comes to allocation of computational resources for information processing, the *value of information* is of critical importance Howard (1966).

As Corning stated Corning (2001), "Shannon's information is blind to the functional properties of the information." According to Corning, the lack of a functional definition of information is the main cause that the full potential of Wiener's cybernetics paradigm Wiener (1965, 1950) has not been realized. Corning suggested the notion of *control information*, which is defined in Corning (2001) as follows:

> "Control information is not a thing but an attribute of the relationships between things. It is the capacity (*know-how*) to *control the acquisition, disposition, and utilization* of matter/energy in purposive (cybernetic) processes."

Building on this line of thinking, information is the useful or *relevant* portion of the data. Here, usefulness or relevance finds a meaning only in the context of a perceptual task aimed for performing decision-making or control Soatto (2009). Also, the notion of relevance plays a key role in feature extraction, dimension reduction, and learning. It can be quantified using the concept of *sufficient statistics*, which was proposed by Fisher Fisher (1922) for parametric distributions. To be more precise, if all the information about the parameters of such distributions can be captured by some functions of a statistical sample, those functions will be considered as sufficient statistics Shamir *et al.* (2010). In this context, the coarsest sufficient partition of random variables, which is drawn from the corresponding distributions, is called *minimal sufficient statistic* Lehmann and Casella (1998). The sufficiency of a statistic for a particular task is specified by the *risk* associated with a control or decision policy. In statistical decision theory, risk is usually defined as the expected loss (or cost) Berger (1985).

Hence, the value of data must be related to its complexity after cancelling the effect of nuisance factors. Nuisance factors, such as clutter, are the cause of much of the complexity in data. This leads us to the notion of information representation, which is associated with the concept of *Kolmogorov complexity*. Kolmogorov's theory states that the length of an optimal (non-redundant) statement (code) that defines a category is a measure of its complexity Li and Vitanyi (2008). Kolmogorov complexity is closely related to the intuitive notion of conceptual difficulty Feldman (2000), Sigman (2004).

Gibson was also one of Shannon's critics; he had a different view on information. Gibson's notion is that information consists of invariants underlying change

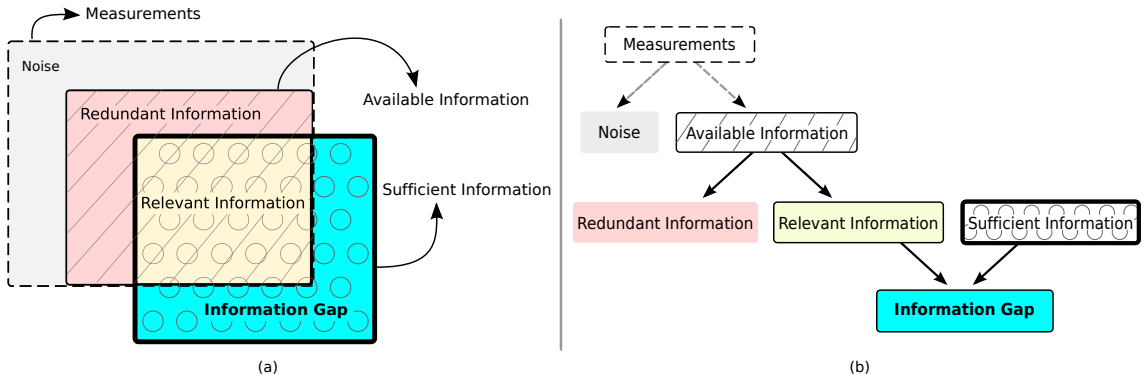


Figure 2.1: Schematic illustration of the information gap: (a) In this graph, the dashed-line square on the top indicates the noisy measurements, from which the available information is extracted. Dashed-line is used to emphasize the fact that measurements may not be in the same space as available information is. In other words, available information is extracted from the measurements by the perception process, and the relationship between them is not necessarily set-inclusion. The square at the other corner of the graph demonstrates the sufficient information, which has an overlap with available information. That part is the relevant information, shaded in yellow. The rest of available information, shaded in pink, is therefore redundant information, since it is not relevant to the task at hand. Finally, part of sufficient information, which is left out of available information and shaded in blue, is the information gap. (b) This diagram illustrates the explained concepts in a tree format. The arrows indicate extraction at the top level, inclusion at the middle level, and subtraction at the bottom level.

Gibson (1976). Extraction of invariants relates to the explanation of how an observer perceives a true phenomenon of interest, despite uncertain sensory inputs on which the perceptions rely Gibson (1986). Inspired by Gibson's work, Soatto called an operational notion of information, *actionable information* Soatto (2009). Since the question of representation is not quite valid without bringing the task into the equation, he addressed the issue of representation, taking account of decision-making and control. Hence, actionable information is a measure for the portion of data that is relevant to the task after removing complexity in the data due to nuisances. In other words, actionable information is defined as the complexity (coding length) of a maximal statistic that is invariant to the nuisances associated with a given task. A statistic (or feature) is invariant if its value does not depend on the nuisance. Maximal invariant is the largest among all invariant statistics in the sense of inclusion of σ-algebras generated by the statistics [1].

The two attributes of relevance and complexity bring us to the concept of *information bottleneck*, which is closely related to Soatto's approach Tishby (1999). Information Bottleneck is aimed at finding a compressed, non-parametric, and model-independent representation T of a random variable Y that is as relevant and informative as possible to another random variable X. In this framework, the mutual information[2] between T and Y, $I(Y;T)$, is a measure of complexity, which should be minimized and the mutual information between T and X, denoted by $I(X;T)$, is a measure of informativeness, which should be maximized. Hence, finding the desired representation T can be formulated as an optimization problem in which the trade-off

[1] A nonempty subset of the power set of a nonempty set is a σ-algebra if it includes the empty set and it is closed under complementation and countable unions Shreve (2004).

[2] Mutual information between two random variables is a measure of the amount of information that one contains about the other. It can also be interpreted as the reduction in the uncertainty about one random variable due to knowledge about the other one Cover and Thomas (2006).

between complexity and informativeness can be controlled by a lagrange multiplier. For parametric distributions, minimal sufficient statistics minimize the mutual information $I(Y;T)$ Cover and Thomas (2006). We may therefore view information bottleneck as a generalization of the classic notion of minimal sufficient statistics Shamir *et al.* (2010).

In general, invariant and sufficient statistics may form two different sets; the difference between these two sets leads us to the concept of *information gap*. In order to be able to bridge the information gap, the system must be able to control the perception process Soatto (2009). Thus, perception and control are quite intertwined with the emphasis on dependence of perception as a thoughtful activity on the capacity for action. Soatto's approach is tailored for active vision, which deals with a specific type of sensors; the approach is an important step towards a general theory of *controlled sensing* Soatto (2009).

The concept of controlled sensing is well described by Noë in his book on "*Action in Perception*" Noë (2004):

> "*What we perceive* is determined by *what we do* (or what we know how to do); it is determined by *what we are ready to do*. ... [To be] precise, we *enact* our perceptual experience, we act it out."

Regarding the critical role of information, complex systems can significantly benefit from a mechanism that controls the directed flow of information in a way to decrease a properly-defined task-specific information gap. Decreasing the information gap will reduce the risk involved in achieving a satisfactory level of performance. In order to find an appropriate name for such a control mechanism, we may look to the neuroscience literature, in the context of which the term *cognitive control* sounds

appealing.

Building on the terminology presented so far, Fig. 2.1 summarizes the concept of the information gap in a way that is relevant to our context. As illustrated,

- *available information* is extracted from noisy *measurements*, which also includes mapping from measurement space to information space.
- Regarding the task at hand, available information can be partitioned into *relevant* and *redundant* information.
- We define *sufficient information* as the required information for performing the task at hand with minimal risk. The mentioned relevant information is therefore the intersection between available information and sufficient information.
- Finally, the difference between sufficient information and available information forms the *information gap*.

In the following sections, we first look at psychology and neuroscience to pave the way how cognitive control can be defined and then, we present a systematic way of implementing cognitive control for managing the information gap.

2.4 How Do We Define Cognitive Control?

The term *cognitive control* was first used by psychologists and neuropsychologists. For example, Gardner et al. Gardner *et al.* (1959) explain six control principles (levelling-sharpening, tolerance for unrealistic experiences, equivalence range, focusing, constricted-flexible control, and field dependence-independence) and 14 experimental tasks to measure them.

Then, Hammond and Summers proposed that Hammond and Summers (1972):

> "Performance in cognitive tasks involves two distinct processes: acquisition of knowledge and cognitive control over knowledge already acquired."

They asserted that acquisition and application of knowledge are independent components of learning in cognitive tasks as well as psychomotor tasks, and then tried to introduce the concept of cognitive control theoretically, and illustrate its empirical significance in studies of human learning, judgment, and interpersonal behavior Hammond and Summers (1972). They also emphasized the role of task-related feedback as opposed to response-oriented feedback and tried to develop a multiple-cue probability learning theory. Some years later, the following definition was proposed in Brass *et al.* (2005):

> "Cognitive control processes refer to our ability to coordinate thoughts and actions in accordance with internal goals."

A similar definition can be found in Kouneiher *et al.* (2009) as well. Yet, another relevant definition presented in Alexander and Brown (2011) is as follows:

> "Cognitive control at the neural level is seen as a result of evaluating the probable and actual outcomes of one's actions."

The work done by Feldman and Friston Feldman and Friston (2010) directly relates the neuropsychological ideas to the probabilistic view of an environment. For example, they explain that through *attention*, the brain optimizes its probabilistic representation of the environment. In information theory's terminology, that might be understood to mean a probabilistic representation with minimum entropy due to

the fact that entropy is a measure of uncertainty about a random variable Shannon (1948); Cover and Thomas (2006).

Both in the human brain Rao and Ballard (1997, 1999) and in cognitive dynamic systems Haykin (2012a,b), a *perception* process is performed on sensory measurements. The role of perception is to extract the available information out of *noisy* sensory measurements. In response to information extracted through the perceptor, the human brain performs actions in order to continually enhance this information in subsequent cycles. These actions could be called *cognitive actions*.

For example, say you are in an almost dark room. You might not recognize all the objects clearly. So, the brain will enlarge the pupil size to increase the light entering into the eyes (i.e., to increase information). Suppose the room is too dark so that changing the pupil size does not help. In such a situation, you may perform an external action such as turning on the light. These actions are not being applied to change the state of the environment (for example, the place of objects in the room), but to mitigate the level of uncertainty.

Cognitive Control from an Engineering Perspective

Thus far, the definitions of cognitive control that we have cited, have been drawn from psychology, neuropsychology, and neuroscience. In this paper, we borrow the term cognitive control from neuroscience, and propose the following definition from an engineering perspective with emphasis on controlling the directed flow of information:

> "Given a probabilistic dynamic system that at least has the perception-action cycle, and ideally mimics the human brain, the function of *cognitive control* is to *adapt the directed flow of information* from the perceptual

> part of the system to its executive part so as to reduce the information gap, which is equivalent to reducing the properly defined risk functional for the task at hand, the reduction being with a probability close to one."

As mentioned before, risk is defined as the expected loss associated with a decision Berger (1985). As a result, there is a requirement for a metric to quantify the information gap. This necessity leads us to the notion of a new type of state to be controlled. This idea is explained in the following section.

2.5 The Two-state Model

At a specific point in time, the state of a dynamic system represents the minimal information that defines the actual condition of the system at that time. By the same token, any change in the state over time (state trajectory) represents the behavior of the system. However, the state is accessible only through noisy measurements, which, in turn, calls for a perception process to provide a *posterior distribution*. As explained in Section 2.3, the difference between the maximal useful information available in the posterior and the sufficient statistics for the given task results in the information gap. This new quantity is thereby defined as the entropic state. The rationale behind choosing this name is that a first-hand candidate for this metric is Shannon's entropy of the posterior due to the fact that entropy can be considered as a global measure of the behavior of the corresponding probability distribution function. This discussion can be summarized in the following two notions of state:

- *system's state*, which is invariant with respect to the measurement process, and
- *entropic state*, which is a metric for quantifying the information gap.

Due to uncertainties both in modelling and in measurements, we have to model the state of the system by random variables, and the result of perception will be the posterior distribution, as explained before. The notion of the two mentioned states naturally results in thinking in terms of a *two-state model* of a cognitive control system, composed as follows:

- *State-space model*, which includes the corresponding mappings from input to state and from state to output. This model also describes evolution of the system's state over time.

- *Entropic-state model* for quantifying the information gap, given the posterior computed by the perception, which depends on environmental uncertainties and disturbances in addition to the sensors' own limitations and modeling errors, as well as the sufficient statistics, which depend on the problem under consideration.

Both models may vary from one cycle of the perception-action process to the next in accordance with statistical variations of the environment. Moreover, the feedback information passed on to the cognitive controller is simply the entropic state. As a result, in practice, cognitive control is the paradigm of reducing the entropic state. In the following section, we first present a short review of *reinforcement learning* (RL) and the fact that it is practiced in mammalian brains, then we explain that RL is naturally the tool for cognitive control.

2.6 Reinforcement Learning

Reinforcement learning (RL) is the mathematical paradigm of learning the best possible action on the sole basis of environmental rewards and punishments (positive and negative rewards). In RL, the goal is to maximize some form of rewards accumulated over the course of time, which are the consequences of a selected action at the current time instance. In neuroscience and computational neuroscience, there are now evidences that support the existence of RL in mammalian brains. This belief has been strongly supported through electrophysiological recordings in behaving animals and functional imaging (fMRI) of human decision-making process Niv (2009); Dayan and Niv (2008).

The history of the existence of RL in mammalian brains starts with behavioral studies and goes way back to Pavlovian (classical) conditioning, which involves conditionally learned predictions Yerkes and Morgulis (1909). Pavlov observed that dogs can be conditioned to predict serving food by a non-relevant stimulus (conditioning stimulus) such as ringing a bell before they really get served. The dogs then salivate to the ringing of the bell even if there would be no food at all. After the classical conditioning comes the instrumental conditioning, which is learning actions that increase the probability of rewarding events and decrease the probability of adverse events. In other words, instrumental conditioning is a form of learning, in which the behavior is modified by the consequences of actions that result in the behavior. As Y. Niv asserts Niv (2009):

> "The study of instrumental conditioning is an inquiry into perhaps the most fundamental form of rational decision-making. This capacity to select actions that influence the environment to one's subjective benefit is

> the mark of intelligent organisms. [Choosing actions] that will maximize rewards and minimize punishments in an uncertain, often changing, and computationally complex world is by no means a trivial task."

In addition to those behavioral research efforts and perhaps above them, more recent studies have revealed strong neuro-cellular/molecular evidences of RL. The dopaminergic neurons in the midbrain are now evidently known as the means of performing RL in the brain Schultz (1998); Niv *et al.* (2005); Niv (2009); Dayan and Niv (2008); Surmeier *et al.* (2009). Along the same line of thinking, one of the most important findings, which proves the existence of RL in mammalian brains, is the discovery of a key RL signal in the brain that is understood as the temporal-difference reward-prediction error Montague *et al.* (1993); Barto (1995); Montague *et al.* (1995, 1996). Additionally, using linear regression, it has been shown that the previously experienced rewards has a part to the dopaminergic response to the current reward, which is exactly according to an exponentially-weighted average of past experience, as is implied by the TD learning rule Bayer and Glimcher (2005); Bayer *et al.* (2007); Niv (2009).

Computationally, RL theory has been formalized in two parallel but distinct lines of research. In the first line, inspired by Pavlovian (classical) and instrumental conditioning and with the aim of artificial intelligence and agent-based learning, Sutton and Barto shaped the core concepts and algorithms of what is now extensively known as the theory of reinforcement learning Sutton (1978); Barto *et al.* (1983); Sutton (1984); Sutton and Barto (1990, 1998). In the second line, based on optimal control and Bellman's dynamic programming Bellman (1956, 1957, 1966), Bertsekas and Tsitsiklis developed a group of stochastic approximations, which have been known

as neuro-dynamic programming and approximate dynamic programming Bertsekas (2005); Bertsekas and Tsitsiklis (1996). However, it should be noted that, aside form the notations, the difference is mostly due to the definition of reward (in Sutton and Barto's paradigm) and cost (in Bertsekas and Tsitsiklis' paradigm); the former should be maximized, while the latter should be minimized.

There are, on the other hand, several occasions that these (mostly) mathematical theories in the machine-learning literature give insight to computational neuroscientists. For example, inspired by artificial neural-networks, Barto and his associates showed that the credit assignment problem[3] can be effectively solved by a learning system, which consists of two neuron-like blocks Barto *et al.* (1983). One block, called the "adaptive critic element (ACE)," evaluates different states of the environment, using a temporal-difference-like learning rule (from which the TD learning rule was later developed Niv (2009)). The other block, called the "associative-search element (ASE)," then learns to select the best action by means of a trial-and-error process, using the evaluation provided by the first block. Notably Niv (2009); Dayan and Niv (2008):

> "These two blocks are the precursors of the modern-day Actor/Critic framework for model-free action selection which has been closely associated with reinforcement learning and action selection in the brain."

In fact, the one central idea in the RL literature is temporal-difference (TD) learning, which is the combination of Monte-Carlo methods and dynamic programming ideas Sutton and Barto (1998). Although TD learning has its early roots in animal

[3]The credit assignment problem is the problem of assigning credit or blame for overal outcomes to each of the internal decisions made by the hidden computational units of the distributed learning system Haykin (2009).

psychology and artificial intelligence Samuel (1959); Klopf (1972), the first algorithm, called TD(0), and the famous example of *random walk* was created by Sutton Sutton (1988) (likewise the term temporal-difference). Since then, the machine-learning literature has proposed various versions and complementary ideas of the TD learning signal, associated with slightly different model-free RL methods Dayan and Niv (2008); Sutton and Barto (1998). The two important ideas among them are Q-learning and Sarsa (State-Action-Reward-State-Action). Q-learning is an off-policy[4] control algorithm, which was first introduced by Watkins in his Ph.D. dissertation Watkins (1989), and the convergence proof was later made rigorous by Watkins and Dayan Watkins and Dayan (1992). On the other hand, the Sarsa algorithm, which is an on-policy algorithm, was first explored by Rummery and Niranjan Rummery and Niranjan (1994), although they called it *modified Q-learning*, and the name Sarsa was latter introduced by Sutton Sutton (1996). Recent evidence looking primarily at one dopaminergic nucleus seems to support Sarsa Morris *et al.* (2006), whereas evidence from a rodent study of the other major dopaminergic nucleus favors Q-learning Roesch *et al.* (2007), Dayan and Niv (2008). Whether Q-learning is performed in the brain or Sarsa (or a combination of them), the fact is that the TD idea is now accepted to be a part of brain's mechanism for selecting the best action.

In what follows, reinforcement learning and planning will be discussed form a more formal point of view. We will also explain how RL is rationally an intrinsic part of cognitive control and briefly review its attributes. A mathematical treatment of RL in cognitive control has been presented in Fatemi and Haykin (2013).

[4]In off-policy algorithms, the policy used for learning is different form the one used for selecting control actions, whereas, in on-policy algorithms, both learning and control share the same policy.

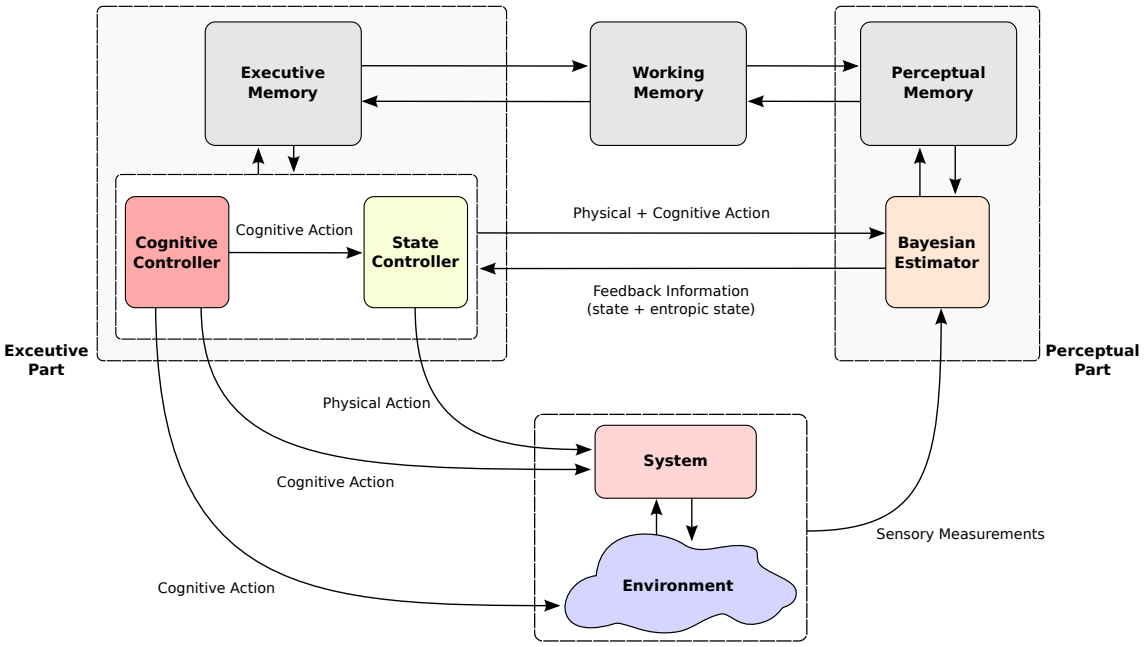


Figure 2.2: Schematic structure of a cognitive control system integrated inside a perception-action cycle of a CDS, and next to a state controller. It is worth noting that in real-world applications, not all the cognitive action links, shown in the diagram, might necessarily exist. Similarly, in case that the CDS acts as an observer (e.g. cognitive radar systems), the state controller will not be included.

2.7 Compositional Structure of Cognitive Control

In this section, we take a closer look at cognitive control in order to provide formal tools for its engineering design. Having the goal of decreasing the information gap and the fact that entropic-state, by definition, quantifies the information gap, cognitive control addresses two sub-problems:

1. optimal estimation of entropic state, and

2. optimal control of entropic state.

In so doing, the perception process is carried out on the sensory measurements, which results in the *posterior* of the system's state. Then, the entropic state is to be calculated from the posterior, and finally it should be controlled in an optimal (or sub-optimal) manner. For example, the perception process might be performed by a Bayesian estimator Ho and Lee (1964), which calculates the posterior of the system's state in each perception-action cycle. When the environment is linear with additive white Gaussian noise, the Bayesian filter simplifies to the Kalman filter as a special case Kalman (1960). However, when the environment is nonlinear and/or non-Gaussian, the usual procedure is to seek some form of approximation to the Bayesian filter; this approximation may take the form of an *extended Kalman filter* Bar-Shalom *et al.* (2001), *unscented Kalman filter* Julier *et al.* (2000), or *Cubature Kalman Filter* Arasaratnam and Haykin (2009) for nonlinear but Gaussian environments, or a *particle filter* Ristic *et al.* (2004); Gordon *et al.* (1993) for general nonlinear and non-Gaussian cases. Then, the entropic state is logically Shannon's entropy Cover and Thomas (2006) of the resulting posterior. Before going further and explaining the configuration of the cognitive controller as an RL agent, let us first take a look at the

entire structure of a cognitive dynamic system (CDS).

2.7.1 Cognitive Control Integrated Inside CDS

Building on Fuster's paradigm, Fig. 2.2 describes the functional block diagram of a cognitive dynamic system integrating within it, the cognitive controller. In this figure, we readily see that the perception-action cycle and memory occupy *physical spaces* of their own. On the other hand, attention and intelligence manifest themselves in the form of *algorithmic mechanisms*, distributed throughout the system.

1. *The Perception-action cycle (PAC)*: Following the terminology of neuroscience, the *perceptual* part of the CDS resides in the right-hand side of the figure, whereas its *executive* counterpart resides in the left-hand side. In effect, the perceptual part of the system, called the *perceptor*, observes the system and the environment directly, whereas the executive part, called the *controller*, observes them indirectly through the "eyes" of the perceptor. This indirect observation of the system and the environment is made feasible by virtue of the *feedback link* that connects the perceptor to the controller.

2. *Memory*: It builds on the perception-action cycle, as depicted in Fig. 2.2. Specifically, we have:

    - Perceptual memory, which is desirably of a hierarchical structure that consists of multiple layers of information processing. The motivation of this hierarchical structure is that of *perceptual abstraction* of the incoming measurements.

- Executive memory, which performs a dual function to the perceptual memory, as shown in Fig. 2.2; the executive memory has a hierarchical structure of its own.
- Working memory, the function of which is to reciprocally couple the perceptor and controller together, thereby constituting an integrated memory system. This reciprocal coupling makes the cognitive controller operate in a *synchronous fashion* from one PAC to the next.

3. *Attention*: It manifests itself in an algorithmic manner as perceptual attention in the perceptor and as executive attention in the controller. While perceptual attention deals with the information overflow problem, executive attention implements a version of the *principle of minimum disturbance* Widrow and Lehr (1990); Haykin (2001).

4. *Intelligence*: It builds on the PAC, memory, and attention, an integrated combination that makes intelligence the most powerful of all the cognitive processes and the most difficult one to define. Similar to attention, intelligence does not occupy a physical place within the CDS, rather its influence is distributed throughout the whole system, and thereby it derives its information-processing power by exploiting all the feedback loops within the CDS, be they global and therefore embodying the environment, or local being confined within the CDS. In short, we may say that the global and local feedbacks are the *facilitator* of computational intelligence in a CDS.

As illustrated in Fig. 2, cognitive actions can influence different parts of the CDS:

- Cognitive actions might be applied to the environment in order to indirectly

affect the perception process. An example of this type is turning on the light in a dark room. Here, the physical state includes the position of objects, which is not affected by the light.

- Cognitive actions might also be applied to the system itself in order to reconfigure the sensors and/or actuators, an example of which is changing the pupil size of our eyes according to different light intensities. Another example is changing the transmitted waveforms of a cognitive radar system.

- Additionally, cognitive actions might also be applied as a part of state-control actions (physical actions). In such a case, a physical action is applied to the system, but with the goal of decreasing the information gap (with or without other goals). For instance, consider a quadratic optimal controller with a cost function of the form

$$J = (\mathbf{x} - \mathbf{x}_d)^{\mathrm{T}} \mathbf{Q} (\mathbf{x} - \mathbf{x}_d) + \mathbf{u}^{\mathrm{T}} \mathbf{R} \mathbf{u} \tag{2.1}$$

to be minimized, where $\mathbf{x}$ and $\mathbf{x}_d$ are the system's state and its corresponding desired-value vectors, $\mathbf{u}$ is the physical control vector, the matrices $\mathbf{Q}$ and $\mathbf{R}$ apply the desired weights for system's state and control respectively, and the superscript T denotes matrix transposition. To include a cognitive goal, we may add another term to (2.1) to take care of the information gap as well. The resulting cost function may now be formulated as

$$J = (\mathbf{x} - \mathbf{x}_d)^{\mathrm{T}} \mathbf{Q} (\mathbf{x} - \mathbf{x}_d) + \mathbf{u}^{\mathrm{T}} \mathbf{R} \mathbf{u} + \beta H, \tag{2.2}$$

where H is the entropic state and the scalar β is an importance factor ($\mathbf{Q}$, $\mathbf{R}$, and β are design parameters).

Nevertheless, all these different types of cognitive actions do not necessarily exist in a given problem. In actual fact, a real-world problem might include only one of the above mentioned types of cognitive actions, even without the system's state controller. An example is a cognitive radar system, which only estimates the state of the target without being able to physically control it. In this paper, we mostly focus on cognitive actions, which are directly applied to the system (or to the environment). Such actions call for the implementation of RL and planning in the cognitive control agent, as discussed next.

2.7.2 RL in Cognitive Control

Let us denote the entropic state by $H_{k|k}$[5], when it reaches the cognitive controller in Fig. 2 at cycle k, after it perceived (estimated) given all the information up to and including cycle k. For optimal control of the entropic state, note that $H_{k|k}$ cannot be controlled directly, even if it gets estimated optimally. Therefore, the primary goal of decreasing $H_{k|k}$ *cannot* be achieved via direct-goal-oriented control techniques, such as full-state feedback control techniques. Additionally, and perhaps more importantly, entropic state is required to be minimized, not just for the next cycle but rather over some lookahead time horizon. In more formal terms, the cognitive-control action at cycle k should optimally minimize the entropic state at cycle $k+1$ and all the cycles thereafter, based upon knowledge of the environment at cycle k. These two issues

[5]Here, $X_{m|n}$ denotes the value of X at time (or at cycle) m, given the information up to and including time (or cycle) n.

naturally form the cognitive-control paradigm as a reinforcement-learning problem. Indeed, RL is at the very heart of a cognitive controller.

In RL, the most basic concept is that of finding a *policy* is facilitated "only" by *rewards*, which are provided by the environment. Based on the Markov assumption, in RL we refer to a model by knowing $\mathcal{P}^a_{ss'}$ and $\mathcal{R}^a_{ss'}$, defined as follows, respectively:

$$\mathcal{P}^a_{ss'} = P[s_{k+1} = s' | s_k = s,\ a_k = a],\ and \tag{2.3}$$

$$\mathcal{R}^a_{ss'} = E[r_k | s_{k+1} = s',\ s_k = s,\ a_k = a], \tag{2.4}$$

where, s is the state supposed to be controlled and a is the action that the control agent can apply to the environment in order to control s. Note that s is not necessarily the physical state of the system. Indeed, in cognitive control, it is the entropic state. $\mathcal{P}^a_{ss'}$ can be found directly from the (stochastic) model that defines the evolution of s over time; however, to find $\mathcal{R}^a_{ss'}$, we need to introduce another equation to model the behavior of the reward function at cycle k (i.e., r_k).

In cognitive control, because the cognitive controller's aim is to decrease the entropic state, a rational reward should include the entropic-state decrement between two subsequent cycles. It is therefore called the *entropic reward*:

$$r_k = g_k(H_{k-1|k-1} - H_{k|k}) \tag{2.5}$$

where, $g_k(.)$ is, in general, an arbitrary function[6]. Then, the RL framework ensures decreasing the entropic state not only in the immediate cycle but in the look ahead horizon.

[6]However, $g(.)$ should be invertible, as discussed in Fatemi and Haykin (2013).

To calculate the entropic reward, assuming that the noise distributions in the state-space model can be predicted (or if they are given), then the entropic state can be predicted using a Bayesian filter. Therefore, we might benefit from the prediction of future rewards for *planning*.

In RL, there are two distinct but similar concepts as follows Sutton and Barto (1998):

- *Learning* uses actual values of the reward.
- *Planning* uses predicted values of the reward.

Planning requires a model of the environment to *simulate* future rewards; however, both learning and planning can use the same algorithm, since they conceptually perform the same task Sutton and Barto (1998). The important point to note here is that learning can be done only once in each perception-action cycle and only for the selected action (since it is based on the actual reward), whereas, in each cycle, planning can be performed for any number of simulated future cycles and any number of actions. Planning and learning can be integrated to achieve the best result. To this end, Sutton and Barto suggest a simple structure called Dyna Sutton and Barto (1998). This paradigm can be extended to include the cognitive-control concepts inside the perception-action cycle. Details of implementation of reinforcement learning and planning in cognitive control, however, are beyond the scope of this paper; see Fatemi and Haykin (2013).

2.8 Computational Experiment on Cognitive Control

In this section, a target-tracking example is presented to demonstrate cognitive control. We consider the tracking of a falling object with a radar with ten measurements per second, based on the benchmark example presented in Haykin *et al.* (2011). Here, the cognitive actions are "changing" the radar transmitter's waveform-parameters in order to mitigate the uncertainty (recall the darkroom example). The target state (i.e., system's state) is $\mathbf{x} = [x_1 \ x_2 \ x_3]^T$, where x_1, x_2 and x_3 are the altitude, velocity and ballistic coefficient that depends on the targets mass, shape, cross-sectional area, and air density, respectively. In the perceptor, a cubature Kalman filter (CKF) Arasaratnam and Haykin (2009) has been used, which provides the estimated state covariance matrix $P_{k|k}$ at cycle k. Having assumed that the true value of the target's state is required, the information gap will then be a measure that shows how inaccurate the CKF is at each cycle. The entropic state is defined as the Shannon entropy corresponding to the CKF output, and calculated by $H_{k|k} = \det\{P_{k|k}\}$. For the entropic-reward function, we used the following:

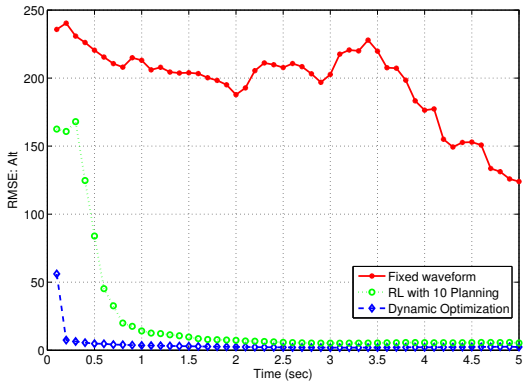
$$r_k = |\log(|H_{k-1|k-1} - H_{k|k}|)|.\text{sgn}(H_{k-1|k-1} - H_{k|k}) \tag{2.6}$$

where sgn(.) shows the standard signum function. We have used the logarithm to decrease the intensity of large differences; however, it should be noted that (as it can be seen in the results of the next experiment) the difference $|H_{k-1|k-1} - H_{k|k}|$ is never close to zero, so that we have incorrect rewards. In any case, if such events can occur, then $|H_{k-1|k-1} - H_{k|k}|$ should be used instead. This entropic reward also includes a

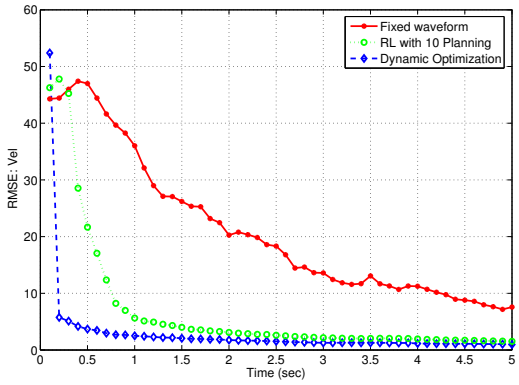
proper sign, which is needed to guide the controller correctly. In the controller side (which is the radar transmitter in this example), there is the possibility of changing the waveform properties in each cycle, which results in 764 cognitive-control actions (i.e., 764 different combinations for the waveform). Applying each action will affect the measurement noise covariance matrix. Finally, Q-Learning Sutton and Barto (1998) was chosen as the method of RL for both learning and planning. The emphasis here is on the use of RL and the integration of learning and planning. Therefore, it is assumed that system noise covariance matrix is given and there exists a model for the measurement covariance matrix as a function of control actions Haykin *et al.* (2011) (i.e., we do not have entropic-state estimation in this example); details of the implementation have been presented in Fatemi and Haykin (2013). All the simulations are performed over 250 Monte Carlo runs to mitigate the effect of randomness. The experiment takes five seconds, therefore, we have 50 perception-action cycles.

2.8.1 Experiment 1: Sub-optimality for reduced computational complexity

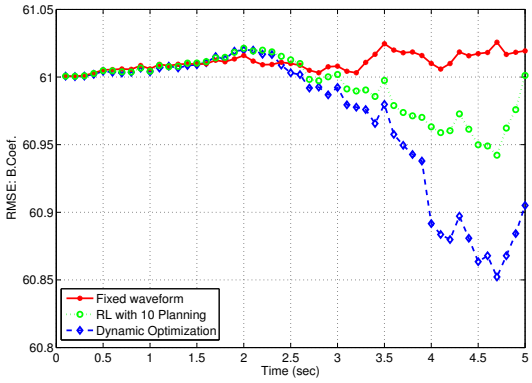
In this case study, the functionality of three different radars is compared in terms of their root mean-squared error (RMSE). We used the actual value of target state in order to compute RMSE, and be able to have the comparison between three different radar configurations. Figure 2.3-(a), (b), and (c) illustrate the RMSE of the three target state variables, namely altitude, velocity and ballistic coefficient, respectively, all of which are plotted versus time. The method of dynamic optimization Haykin *et al.* (2011, 2012b) has been used as a frame of reference, although it may not be used in real-time due to its heavy computational load. Additionally, dynamic



(a)



(b)



(c)

Figure 2.3: Results of computational experiment of Case Study 1. Figures (a), (b), and (c) illustrate the RMSE for the three state variables correspondingly.

optimization does not include infinite look-ahead horizon in the sense of Bellman equation Bertsekas (2005); Sutton and Barto (1998). For the cognitive control, here we have Q-learning plus 10 actions selected randomly for planning at each PAC. The red bulleted line on the top of the graphs is the radar with no controller (only CKF). The green circled line and blue diamond lines are RL with 10 planning and dynamic optimization methods, respectively. The RL method (with 10 actions used for planning in each cycle) is almost two orders of magnitude faster than the method using dynamic optimization; hence, RL significantly improves computational complexity at the expense of optimality.

2.8.2 Experiment 2: Information-processing power of planning

In Figure 2.4, we have illustrated the entropic-state decrement over an increasing number of perception-action cycles. The dot magenta line on the top (which almost sticks to the squared-line beneath it) is the fixed-waveform radar, where there are no cognitive control actions at all. Nevertheless, because CKF is used in the perceptor, the entropic-state still decreases (almost two orders of magnitude over the entire 50 perception-action cycles). Following that, the blue squared line is for cognitive control only with learning. Since the total number of perception-action cycles is far less than the entire number of possible actions (50 vs. 764), this method performs on average, no better than the fixed-waveform method (because Q-learning could not converge to any meaningful policy). Then, we retain RL but this time, we have also added planning. This method is repeated for three different number of random actions, which are selected for planning at each cycle: (a) only one random action (red circled

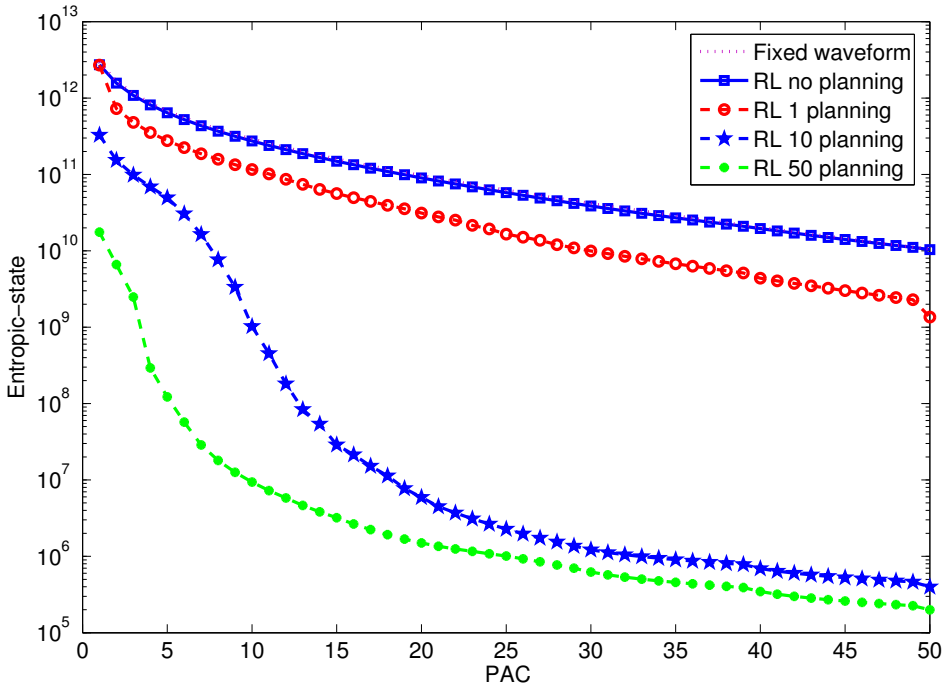


Figure 2.4: Decreasing the entropic-state in a target tracking example using cognitive control. Note that "Fixed waveform" and "RL no planning" lines almost coincide on each other on the top of the graph.

line), (b) 10 random actions (stared blue line), and (c) 50 random actions (asterisk green line). In the first case that only one action is selected for planning, although one planning is still much less than the entire number of actions, yet it is enough to demonstrate an obvious improvement. As for the other two cases, they both show more than four orders of magnitude improvement in the entropic-state reduction.

2.9 Concluding Remarks

2.9.1 Summarizing Highlights of the Paper

1. Control of directed information flow in CDS is summed up in the information gap, which is defined as the difference between relevant information (useful part of what is extracted from the measurements) and sufficient information (i.e., the information needed to perform a task of interest with minimal risk).

2. Cognitive control is itself defined as the process of adapting the directed flow of information form the perceptual part of a dynamic system to its executive part, such that the information gap is reduced by an amount equivalent to a reduction in the properly defined risk functional, with a probability close to one.

3. Two-state model, one being the system's state and the other being the entropic state that quantifies the information gap.

4. Reinforcement learning, exemplified by Q-learning, the employment of which in a cognitive controller is assured by means of the entropic state being computed in the perceptor and passed directly to the controller as feedback information.

5. Planning, an integral part of reinforcement learning, requires a model of the environment to simulate future rewards.

6. Lessons learned from the computational experiment:

    - The use of reinforcement learning in a cognitive controller results in a significant reduction in computational resources in exchange for a suboptimal

performance.

- The incorporation of planning into reinforcement learning enhances the information-processing power of a cognitive controller.

2.9.2 Comparison of Cognitive Control versus Adaptive Control and Neurocontrol

1. *Cognitive Control versus Adaptive Control*: Adaptation is an integral part of cognition. We therefore expect that whatever task is performed by an adaptive controller, the cognitive controller does it better. To elaborate, it can be argued that an adaptive controller could accommodate three of the basic functions of cognition, namely the perception-action cycle (PAC), attention, and intelligence. In other words, an adaptive controller lacks memory, whereas memory (and therefore learning) is an integral part of a cognitive controller, hence the ability to outperform an adaptive controller at the expense of increased system complexity.

2. *Cognitive Control versus Neurocontrol*: For a neurocontroller, to acquire artificial intelligence and therefore be able to learn from its environment, the traditional approach is to build a neural network into its design. In direct contrast, a cognitive controller looks to neuroscience for guidance. In specific terms, the PAC, memory, and attention are built into the cognitive controller's design; thereby, the controller acquires intelligence, which is the most powerful among all the functions that define cognition. Moreover, the intelligence is distributed throughout the dynamic system via local and global loops. While the

neurocontroller works as a whole or it does not work at all Hrycej (1997), cognitive processes (i.e., PAC, memory, attention, and intelligence) can be built into the system in an orderly fashion. We therefore expect a cognitive controller to outperform a neurocontroller for a given task, again at the expense of increased complexity.

Most importantly, it should also be emphasized that cognitive control has an intrinsic difference compared to adaptive control and neurocontrol in that the goal of cognitive control is to reduce the information gap. Indeed, as illustrated in Fig. 2, a cognitive control agent may exist next to or be independent of any other physical controller. In other words, cognitive control is not a *replacement*, but is an *addition* to a system design paradigm.

To sum up, cognitive control is a new way of thinking about control inspired by the human brain. Over and above the improved utilization of computational resources, yet be able to deliver a good performance through the incorporation of planning in reinforcement learning, it is in *risk management*, where cognitive control will make its biggest difference to the control literature.

Acknowledgement

The authors of this paper gratefully acknowledge the financial support provided by the National Science and Engineering Research Council (NSERC) of Canada for the work reported in this paper. Just as importantly, the detailed feedback notes on different aspects of the paper by four reviewers have not only reshaped the paper into its present form but also made the paper the best it could be.

The following chapter is a reproduction of an IEEE copyrighted, published paper*:

Fatemi M., Haykin, S.

Cognitive Control: Theory and Application, *IEEE Access*, **2**, 698–710, June 2014.

---

* At the time of submitting this thesis (October 2014), the paper has been marked as **top-listed** among the most popular papers of the IEEE Access Journal for three months in a row since its publication.

Chapter 3

Cognitive Control: Theory and Application

3.1 Abstract

From an engineering point-of-view, cognitive control is inspired by the prefrontal cortex of the human brain; cognitive control may therefore be viewed as the overarching function of a cognitive dynamic system. In this paper, we describe a new way of thinking about cognitive control that embodies two basic components: learning and planning, both of which are based on two notions: 1) two-state model of the environment and the perceptor, and 2) perception-action cycle, which is a distinctive characteristic of the cognitive dynamic system. Most importantly, it is shown that the cognitive control learning algorithm is a special form of Bellman's dynamic programming. Distinctive properties of the new algorithm include the following: a) Optimality of performance, b) algorithmic convergence to optimal policy, and c) linear law of complexity measured in terms of the number of actions taken by the cognitive

controller on the environment.

To validate these intrinsic properties of the algorithm, a computational experiment is presented, which involves a cognitive tracking radar that is known to closely mimic the visual brain. The experiment illustrates two different scenarios: a) the impact of planning on learning curves of the new cognitive controller, and b) comparison of the learning curves of three different controllers, based on dynamic optimization, traditional Q-learning, and the new algorithm. The latter two algorithms are based on the two-state model, and they both involve the use of planning.

3.2 Introduction

Cognition is a distinctive characteristic of the human brain, which distinguishes itself from all other mammalian species. It is therefore not surprising that when we speak of cognitive control, we naturally think of cognitive control in the brain Miller and Cohen (2001). Most importantly, cognitive control resides in the executive part of the brain, reciprocally coupled to its perceptual part via the working memory Fuster (2003). The net result of this three-fold combination is the perception-action cycle that embodies the environment, thereby constituting a closed-loop feedback system of a global kind.

In a point-of-view article published in the Proceedings of the IEEE on the integrative field of Cognitive Dynamic Systems viewed from an engineering perspective, it was first described in the literature Haykin (2006a). This new way of thinking was motivated by two classic papers: "Cognitive Radio: Brain-empowered Wireless Communications" Haykin (2005), and "Cognitive Radar: A Way of the Future" Haykin (2006b). However, it was a few years later that the second author became aware of

Fuster's basic principles of cognition, namely, perception-action cycle, memory, attention, and intelligence. It was that particular awareness that prompted the engineering need for bringing cognitive control into the specific formalism of cognitive dynamic systems.

During the past few years, cognitive control viewed from an engineering perspective, has featured in two journal papers, as summarized here:

1. In Haykin *et al.* (2011), a control-theoretic approach was described using *dynamic optimization*, representing a simplified version of Bellman's dynamic programming. It was in this paper that for the first time, we faced the imperfect state information problem, so called due to the fact that the controller does not have the provision to sense the environment in a direct manner. Although it is feasible to mitigate this problem algorithmically as formulated in Bertsekas (2005), the incurred cost of computational complexity is so expensive that we had to limit the dynamic programming algorithm with no provision in looking into the future; thereby the name dynamic optimization.

2. The *two-state model*, proposed in Haykin (2012c), provides the most effective notion to bypass the imperfect state information problem; more will be said on this notion later in the paper. For the present, it suffices to say that practical validity of this new way of thinking about cognitive control was demonstrated in Haykin *et al.* (2012a) through the use of Q-learning that represents an approximate form of dynamic programming.

It was these two early contributions to cognitive control that set the stage for a novel cognitive controller presented in the current paper. Unlike the two previous procedures for implementing cognitive control, the new cognitive controller is optimal,

in that it is well and truly a special case of Bellman's dynamic programming. Most importantly, unlike dynamic programming, the new cognitive controller follows a *linear law* of computational complexity measured in terms of actions taken on the environment. The other desirable attribute of this cognitive controller is the use of planning.

The rest of the paper is organized as follows:

- With cognitive control being the primary objective of the paper, Section II discusses two underpinnings of cognitive control, namely, learning and planning, each of which is based on two notions:

  1. The two-state model, which embodies target state of the environment and entropic state of the perceptor.
  2. The cyclic directed information flow, which follows from the global perception-action cycle: the first principle of cognition.

- Next, mathematical formalism of the learning process in cognitive control is presented in Section III, resulting in a state-free cognitive control learning algorithm, where computational complexity follows the linear law.

- Section IV goes one step further: the cognitive control learning algorithm is shown to be a special case of the celebrated Bellman's dynamic programming; hence, convergence and optimality of the new algorithm.

- Section V briefly discusses how to balance optimality of the learning process versus the convergence rate of the cognitive control learning algorithm, thereby setting the stage for both planning and the explore/exploit tradeoff, which are discussed in Sections VI and VII, respectively.

- At this point in the paper, we are ready to address structural composition of the cognitive controller in Section VIII.
- Then, Section IX validates an engineering application of the cognitive controller by presenting a computational experiment involving a cognitive tracking radar.
- Finally, Section X concludes the paper.

3.3 Cognitive Control

From a cognitive neuroscience perspective, cognitive control plays a key role in the prefrontal cortex in the brain; most importantly, cognitive control involves two important processes: learning, and planning. And, so it is in a cognitive dynamic system, inspired by the brain. The learning process is discussed in Section 3.4, followed by the planning process, which is discussed in Section 3.7. Both processes are dependant on the two-state model as well as the cyclic directed information flow, which are discussed in what follows.

3.3.1 The Two-state Model

As mentioned in the introduction, the two-state model is an essential element in deriving the cognitive control algorithm. By definition, the two-state model embodies two distinct states, one of which is called the *target state*, pertaining to a target of interest in the environment. The second one is called the *entropic state* of the perceptor[1], the source of which is attributed to the unavoidable presence of uncertainties in the environment as well as imperfections in the perceptor itself.

[1]The terms *cognitive perceptor* and *perceptor* are used interchangeably in the paper.

Insofar as cognitive control is concerned, the two-state model is described in two steps as follows:

1. State-space model of the environment, which embodies the following pair of equations:

$$\begin{cases} \text{Process equation:} & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{v}_k \\ \text{Measurement equation:} & \mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k \end{cases} \tag{3.1}$$

   where $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{z}_k \in \mathbb{R}^m$ are the state and measurement (observable) vectors at cycle k, respectively; $\mathbf{f}$ is a vector-valued transition function, and $\mathbf{h}$ is another vector-valued function that maps the target state-space to the measurement space[2]; $\mathbf{v}_k$ denotes an additive process noise that acts as the driving force, evolving state $\mathbf{x}_k$ at cycle k to the updated state $\mathbf{x}_{k+1}$ at cycle $k+1$; finally $\mathbf{w}_k$ is the additive measurement noise.

2. Entropic state model of the perceptor, which is formally defined by the following equation:

$$\text{Entropic-state equation:} \quad H_k = \phi(p(\mathbf{x}_k|\mathbf{z}_k)) \tag{3.2}$$

   The H_k is the entropic state at cycle k in accordance with the state posterior $p(\mathbf{x}_k|\mathbf{z}_k)$ in the Bayesian sense, which is computed in the perceptor[3]. As such, H_k is the state of the perceptor, and ϕ is a quantitative measure such as Shannon's

[2]In order to guarantee the existence and uniqueness of the solution to (4.1), both $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are assumed to be Lipschitz continuous Rudin (1976); i.e., there exists $\lambda > 0$ such that $||\mathbf{f}(x_2) - \mathbf{f}(x_1)|| \leq \lambda||x_2 - x_1||$, for all x_1 and x_2, with $||.||$ denoting the Euclidian norm and likewise for $\mathbf{h}(\cdot)$.

[3]To emphasize the cycles, in which the state and the measurement are taken, in this paper, we may also use the notation $H_{k|k}$, in accordance with the subscripts in the posterior, $p(\mathbf{x}_k|\mathbf{z}_k)$.

entropy[4].

It is important to note here that, in general, Shannon's entropy could assume the value zero; however, in cognitive control, the entropic state H_k will always have a non-zero, positive value due to the fact that the environment always involves uncertainty and we can never reach perfect target-state reconstruction with 100% accuracy.

By definition Haykin *et al.* (2012a), the function of cognitive control is defined as follows:

> To control the entropic state (i.e., state of the perceptor), such that the target's estimated state continues to be reliable across time.

Cognitive control therefore requires the entropic state, which is computed in the perceptor and then passed to the cognitive controller as *feedback information*.

3.3.2 Cyclic Directed Information Flow

The *global perception-action cycle*, depicted in Fig. 4.1, plays a key role in a cognitive dynamic system; it is said to be global, in that it embodies the perceptor in the right-hand side of the figure, the cognitive controller in the left-hand side of the figure, and the surrounding environment, thereby constituting a closed-loop feedback system. In descriptive terms, the global perception-action cycle operates on the observables

[4]Shannon's entropy for a random variable X, having the probability density function $p_X(x)$ in the sample space Ω, it is defined asCover and Thomas (2006):

$$H = \int_{\Omega} p_X(x) \log \frac{1}{p_X(x)} dx$$

Correspondingly, Shannon's entropy of target state $\mathbf{x}_k$ with the posterior $p(\mathbf{x}_k|\mathbf{z}_k)$ is defined as:

$$H_k = \int_{\mathbb{R}^n} p(\mathbf{x}_k|\mathbf{z}_k) \log \frac{1}{p(\mathbf{x}_k|\mathbf{z}_k)} d\mathbf{x}_k.$$

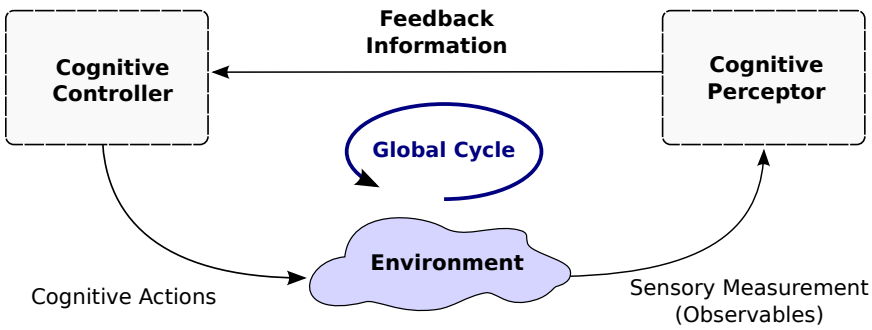


Figure 3.1: Block diagram of the global perception-action cycle in a cognitive dynamic system.

(measurements) of the environment, so as to separate *relevant information* about the environment from irrelevant information that is not needed. The lack of sufficient relevant information extracted from the observables is attributed to the unavoidable uncertainties in the environment as well as design imperfections in the perceptor. The entropic state introduced in sub-section A is indeed a measure of the lack of sufficient information. The entropic state supplies the *feedback information*, which is sent to the cognitive controller by the perceptor. With this feedback information at hand, the cognitive controller acts on the environment, producing a change in the observables. Correspondingly, this change affects the amount of relevant information about the environment, which is extracted from the new observables. A change is thereby produced in the feedback information and with it, a new action is taken on the environment by the cognitive controller in the next perception-action cycle. Continuing in this manner from one cycle of perception-action to the next, the cognitive dynamic system experiences a *cyclic directed information flow*, as illustrated in Fig. 4.1.

In addition to feedback information directed from the perceptor to the cognitive

controller, there is also a feedforward information link from the cognitive controller to the perceptor. In other words, the perceptor and the cognitive controller are reciprocally coupled. This important link is illustrated in Fig. 3.3, and will be discussed later in Section 3.7.

3.4 Formalism of The Learning Process in Cognitive Control

Previously in Section 3.3, we introduced learning and planning as the two important processes in the execution of cognitive control. In actual fact, the aims of both learning and planning processes are to improve an entity called *cognitive policy*. By definition, cognitive policy is the probability distribution of cognitive actions at the perception-action cycle k, which includes the influence of action taken in the preceding cycle, $k-1$. Let $\pi_k(c, c')$ denote the cognitive policy at cycle k, defined as follows:

$$\pi_k(c, c') = \mathbb{P}[c_{k+1} = c' | c_k = c]; \quad \text{with } c, c' \in \mathcal{C},$$

where $\mathcal{C}$ is the cognitive action-space, c and c' are two cognitive actions, and $\mathbb{P}$ is a probability measure.

The cognitive policy should pertain to the long-term value of cognitive actions. In order to formalize a long-term value for each cognitive action, an immediate *reward* has to be defined. To this end, the *incremental deviation* in the entropic state from

one cycle to the next, denoted by $\Delta_1 H_k$, is defined by

$$\Delta_1 H_k = H_{k-1} - H_k. \tag{3.3}$$

where H_{k-1} and H_k are the entropic states at the preceding and current cycles $k-1$ and k, respectively. Note that $\Delta_1 H_k$ could assume a positive or negative value, depending on conditional changes in the environment. The *entropic reward* for cognitive control at cycle k, denoted by r_k, is now defined as an arbitrary function of two entities: the entropic-state's value, $H_{k|k}$, and the incremental deviation $\Delta_1 H_k$, as shown by:

$$r_k = g_k(H_k, \Delta_1 H_k) \tag{3.4}$$

where, g_k is an arbitrary scalar-valued operator. For example, the entropic reward in (3.4) may take the following form:

$$r_k = \frac{\Delta_1 H_k}{H_k} \tag{3.5}$$

where the entropic states H_k always assumes a positive value.

Remark 1. *Computation of the entropic reward r_k requires knowledge of the incremental deviation $\Delta_1 H$, defined in* (3.3)*. To satisfy* (3.4)*, it follows therefore that we need a short-term memory that accounts for the preceding entropic-state H_{k-1}.*

As a result, after taking a cognitive action, a positive r_{k+1} indicates a decreasing deviation that can be considered as an immediate reward for the taken action. Conversely, a negative r_{k+1} demonstrates a cost against the selected action.

We may now define the following *value-to-go function* for the cognitive controller:

$$J(c) = \mathbb{E}^{\pi}[r_{k+1} + \gamma r_{k+2} + \gamma^2 r_{k+3} + ... \mid c_k = c] \tag{3.6}$$

where $\gamma \in [0, 1)$ denotes a *discount factor* that decreases the effect of future actions, and $\mathbb{E}$ denotes the expected value operator for which the expected value is calculated using the policy distribution π_k.

Lemma 1. *$J(c)$ satisfies the following recursion:*

$$J(c) = \mathcal{R}(c) + \gamma \Sigma_{c'} \pi_k(c, c') J(c') \tag{3.7}$$

where $\mathcal{R}(c) = \mathbb{E}^{\pi}[r_{k+1}|c_k = c]$ denotes the expected immediate reward at cycle $k+1$ of the currently selected action c at cycle k.

Proof. Using the linear property of the expected value operator Bertsekas and Tsitsiklis (2008), we may expand (3.6) as follows:

$$\begin{aligned} J(c) &= \mathbb{E}^{\pi}[r_{k+1} + \gamma r_{k+2} + \gamma^2 r_{k+3} + ... \mid c_k = c] \\ &= \mathbb{E}^{\pi}[r_{k+1}|c_k = c] + \gamma \mathbb{E}^{\pi}[\Sigma_{j=0}^{\infty} \gamma^j r_{k+j+2}|c_k = c] \end{aligned}$$

In the second line of the equation, the first term is the expected immediate reward $\mathcal{R}(c)$. The second term lacks c_{k+1} in the condition to be the action-value of one-step

future action. Therefore, using the *total probability theorem*[5], we may write:

$$\begin{aligned} J(c) &= \mathcal{R}(c) + \gamma \mathbb{E}^{\pi}[\Sigma_{j=0}^{\infty} \gamma^j r_{k+j+2} | c_k = c] \\ &= \mathcal{R}(c) + \gamma \Sigma_{a'} \mathbb{P}[c_{k+1} = c' | c_k = c] \times \mathbb{E}^{\pi}[\Sigma_{j=0}^{\infty} \gamma^j r_{k+j+2} | c_k = c, c_{k+1} = c'] \\ &= \mathcal{R}(c) + \gamma \Sigma_{c'} \pi_k(c, c') J(c') \end{aligned}$$

□

It is noteworthy that (3.7) has the flavor of Bellman's equation for dynamic programming, on which more will be said in the next section. In order to have a *recursive* algorithm, we may express the recursion in the following form:

$$J(c) \leftarrow \mathcal{R}(c) + \gamma \Sigma_{c'} \pi_k(c, c') J(c') \tag{3.8}$$

With recursion in mind and for the sake of flexibility, on every cycle of the recursion, (3.8) becomes more of practical value in an algorithmic sense by having $J(c)$ plus a weighted incremental update, as shown by

$$J(c) \leftarrow J(c) + \alpha[\mathcal{R}(c) + \gamma \Sigma_{c'} \pi_k(c, c') J(c') - J(c)] \tag{3.9}$$

where $\alpha > 0$ is a *learning parameter*. On the basis of the recursion described in (3.9), we may formulate Algorithm 1, which updates the value-to-go function from one cycle

[5]For random variables X, Y and Z defined in Ω_X, Ω_Y, and Ω_Z, respectively, the total probability theorem Bertsekas and Tsitsiklis (2008) says:

$$\mathbb{E}[X|Y = y] = \sum_{z \in \Omega_Z} \mathbb{P}[Z = z|Y = y] \mathbb{E}[X|Y = y, Z = z]$$

Algorithm 1: A value-to-go updating algorithm under Lemma 1.

1 **Varables:**
2 $J :=$ value-to-go function
3 $\gamma :=$ discount factor, $\gamma \in [0, 1)$
4 $\alpha :=$ learning parameter, $\alpha > 0$
5 **Inputs:**
6 $\mathcal{R}(c) :=$ expected reward of action c
7 $\pi :=$ learning policy
8 **Updating:**
9 **for** *all cognitive actions* $c \in \mathcal{C}$ **do**
10 $\quad J(c) \leftarrow J(c) + \alpha[\mathcal{R}(c) + \gamma \Sigma_{c' \in \mathcal{C}} \pi_k(c, c') J(c') - J(c)]$
11 **end**

of perception-action to the next.

From an implementation perspective, the term $\Sigma_{c' \in \mathcal{C}} \pi_k(c, c') J(c')$ in line 10 of Algorithm 1 may be substituted by $mean\{J(c)\}$, simply by considering $\pi_k(c, c')$ to be a uniform distribution here[6]. This method is called *off-policy* and is known also to be convergent to the optimal policy Sutton and Barto (1998).

Hereafter, the recursive algorithm based on (3.9) is referred to as the *cognitive control learning algorithm*. This algorithm has been derived by exploiting the cyclic information flow that is a characteristic of the global perception-action cycle. With $mean\{J(c)\}$ substituted in line 10, examination of Algorithm 1 immediately reveals that this algorithm follows a linear law of computational complexity with respect to the number of actions taken by the cognitive controller, which is the cardinality of the cognitive action-space $\mathcal{C}$.

[6]With $mean\{J\}$ substituted in (3.9), the learning rule becomes similar to the traditional Q-learning algorithm Watkins (1989); Watkins and Dayan (1992), yet it differs from it in two basic fronts: First, Q-learning uses $max\{J\}$ as an approximation, and second, (3.9) is calculated for all the cognitive actions, whereas in Q-learning, the update is only for the current state and action. In Section IX, we have chosen traditional Q-learning as a frame of reference for comparison in our computational experiment.

From a practical perspective, linearity of the algorithm by itself is not adequate. To be more precise, convergence as well as optimality of the algorithm would have to be justified theoretically. With this objective in mind, we propose to shift gear from the cognitive perspective and appeal to Bellman's dynamic programming, which is known to be both convergent and optimal Bellman (1957, 1961).

3.5 Cognitive Control Learning Algorithm Viewed as a Special Case of Bellman's Dynamic Programming

Bellman's celebrated dynamic programming algorithm Bellman (1957, 1961) was first described in the literature about fifty five years ago; yet it remains to occupy an important place in the study of optimal control. The optimality manifests itself in terms of maximizing a long-term value-to-go function; it is formally defined over time by means of immediate rewards. In its basic form, Bellman's dynamic programming deals with finite-horizon problems. However, from an analytic perspective, the preferred mathematical approach is to deal with infinite-horizon problems, where the rewards are considered over an infinite number of cycles.

In dynamic programming, a system is defined by its set of states $\mathcal{S}$ and set of actions $\mathcal{A}$. On a cycle-by-cycle basis, the system has a transition from state $s \in \mathcal{S}$ at cycle k to $s' \in \mathcal{S}$ at cycle $k+1$ as a result of action $a \in \mathcal{A}$. This transition results in an immediate reward $r_{k+1} \in \mathbb{R}$. The state-action-based *value-to-go* function is then

defined by the formula:

$$\tilde{J}(s,a) = \mathbb{E}^{\tilde{\pi}}[r_{k+1} + \gamma r_{k+2} + \gamma^2 r_{k+3} + ...|s_k = s, a_k = a],$$

for which, $\tilde{\pi}_k(s,a) = \mathbb{P}[a_{k+1} = a|s_k = s]$ is the state-based *policy* when the system is in state s; the tilde in $\tilde{\pi}_k(s,a)$ is intended to differentiate it from the policy $\pi(c,c')$ used in the previous section. As mentioned previously, $\mathbb{P}$ denotes a probability measure and $\mathbb{E}^{\tilde{\pi}}$ denotes the expected value operator with respect to the policy $\tilde{\pi}$. In Appendix A, it is shown that $\tilde{J}(s,a)$ obeys *Bellman's equation* for dynamic programming as follows:

$$\tilde{J}(s,a) = \sum_{s' \in \mathcal{S}} T^a_{ss'}[R^a_{ss'} + \gamma \sum_{a' \in \mathcal{A}} \tilde{\pi}_k(s,a')\ \tilde{J}(s',a')] \tag{3.10}$$

where the transition probability $T^a_{ss'}$ and the immediate expected reward $R^a_{ss'}$ are respectively defined by the following pair of equations:

$$\begin{cases} T^a_{ss'} = \mathbb{P}[s_{k+1} = s'|s_k = s,\ a_k = a], \\ R^a_{ss'} = \mathbb{E}^{\tilde{\pi}}[r_{k+1}|s_{k+1} = s', s_k = s, a_k = a] \end{cases} \tag{3.11}$$

The optimal value-to-go function, denoted by $\tilde{J}^*$, is obtained by maximizing the sum of all the terms in (3.10) with respect to action a. Unfortunately, the end result of this maximization is an exponential growth in computational complexity, known as the *curse of dimensionality* Bellman (1961). Nevertheless, the algorithm is known to be convergent as well as optimal Sutton and Barto (1998). Algorithm 2 describes a dynamic programming algorithm corresponding to (3.10). Inclusion of the two *nested for-loops* in Algorithm 2 (lines 10 and 11) is indeed the root of the curse of

Algorithm 2: A value-to-go updating algorithm for a generic dynamic programming.

Varables:
$\tilde{J}$:= value-to-go function
γ := discount factor, $\gamma \in [0, 1)$
α := learning parameter, $\alpha > 0$
Inputs:
$T^a_{ss'}$:= transition probability
$R^a_{ss'}$:= expected reward
$\tilde{\pi}$:= learning policy
Updating:
for *all states* $s \in \mathcal{S}$ **do**
  for *all actions* $a \in \mathcal{A}$ **do**
    $\tilde{J}(s,a) \leftarrow \tilde{J}(s,a) + \alpha[\sum_{s' \in \mathcal{S}} T^a_{ss'}[R^a_{ss'} + \gamma \sum_{a' \in \mathcal{A}} \tilde{\pi}_k(s,a')\ \tilde{J} - \tilde{J}(s,a)]$
  end
end

dimensionality problem.

The cognitive control learning algorithm, described in Section 3.4, is indeed *state-free*. On the other hand, in light of the fact that Bellman's dynamic programming is *state-dependant*, the question to be addressed is:

> How do we make Bellman's dynamic programming to be on par with the cognitive control learning algorithm, such that both of them are state-free?

To this end, consider the two models depicted graphically in Fig. 3.2. In a generic sense, part (a) of the figure illustrates the transition from state $s_k = s$ at time k to a new state $s_{k+1} = s'$ at time $k + 1$ under the influence of action $a_k = a \in \mathcal{A}$, as it would be in Bellman's dynamic programming. On the other hand, part (b) of the figure depicts a "special" transition that involves a single state s and therefore a graphical representation of the following model:

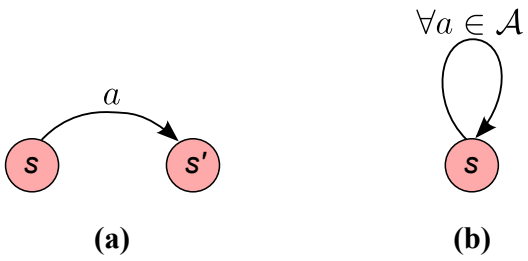


Figure 3.2: Graphical illustration of state transition in dynamic programming: (a) generic model, and (b) special case of Model 1.

Model 1:

- State-space contains only one state, that is

  $\mathcal{S} = \{s\}$,

- There exists a self-loop for s, including all the actions in the action space, i.e.,

  $\mathbb{P}[s_{k+1} = s | s_k = s, a_k = a] = 1, \ \forall a \in \mathcal{A}$.

Model 1 is a valid model that lends itself to the application of Bellman's dynamic programming; moreover, the application of dynamic programming to Model 1 will not affect the properties of optimality and convergence, which are basic to dynamic programming Bertsekas (2005). The idea behind using Model 1 is to remove dependence of the dynamic programming algorithm on the states, as it would be in the cognitive control learning algorithm.

We next show that the following lemma holds:

Lemma 2. *Dynamic programming of Model 1 is equivalent to the cognitive control learning algorithm.*

Proof. It suffices to show that Bellman's equation for Model 1 is identical to the recursion equation in Lemma 1. Assume that the action-space of Model 1 is the same

as the cognitive action-space in the previous section, that is, $\mathcal{A} = \mathcal{C}$. Because Model 1 has only one state, the outer summation in Bellman's equation (3.10) has only one term with the transition probability $T^a_{ss'}$ being one (due to the second property of Model 1). Additionally, the current and next states s_k and s_{k+1} in the condition of $R^a_{ss'}$ are always equal to s; hence, they add no additional information to $R^a_{ss'}$, and they are therefore redundant in $R^a_{ss'}$. We may thus formally write:

$$\begin{aligned} R^a_{ss'} &= \mathbb{E}[r_{k+1}|s_{k+1} = s, s_k = s, a_k = a] \\ &= \mathbb{E}[r_{k+1}|a_k = a] \\ &= \mathcal{R}(a) \end{aligned}$$

Similarly, since in Bellman's dynamic programming, current actions are independent of previous actions, we may express the corresponding policy:

$$\begin{aligned} \tilde{\pi}_k(s, a') &= \mathbb{P}[a_{k+1} = a'|s_k = s] \\ &= \mathbb{P}[a_{k+1} = a'|s_k = s, a_k = a] \\ &= \mathbb{P}[a_{k+1} = a'|a_k = a] \\ &= \pi_k(a, a') \end{aligned}$$

Substituting $R^a_{ss'}$ and $\tilde{\pi}_k$ in (3.10) will then prove the lemma. □

On the basis of Lemma 2, we may now state that the cognitive control learning algorithm is indeed a special case of of dynamic programming. Accordingly, the cognitive control learning algorithm inherits the basic properties of dynamic programming, namely, convergence and optimality. We may now conclude the section with

the following statement:

> The cognitive control learning algorithm is not only linear, but also convergent to the optimal policy.

3.6 Optimality vs. Convergence-rate in Online Implementation

Thus far, we have addressed optimality and convergence of the cognitive control learning algorithm. However, there are two other practical issues relating to the convergence rate of the learning process, which are described as follows:

1. To implement the *for-loop* in Algorithm 1, the expected immediate rewards should be known for *all* the actions in the action space $\mathcal{C}$. In reality, the immediate reward is available only for the currently selected action, which can replace its expected value. Hence, there would be $M = |\mathcal{C}|$ perception-action cycles required to collect information about all the actions. To overcome this first issue we propose to use *planning*, which is to be described in Section 3.7.

2. If we were to explore all the M cognitive actions in the action space $\mathcal{C}$, we would end up with a cognitive controller of poor performance in the exploration period. To overcome this second issue, we propose to use the ϵ-greedy strategy, which is to be discussed in Section 3.8.

Thus, through the use of planning and ϵ-greedy strategy, an efficient convergence rate with optimal performance for *on-line* applications is assured.

3.7 Formalism of the Planning Process in Cognitive Control

Planning is defined as the process of using *predicted* future rewards in order to improve our knowledge of the value-to-go function $J(c)$. Hence, the planning process plays a key role in speeding up the convergence rate of the cognitive controller. To this end, predicted values of entropic rewards are therefore required.

Referring to (4.1), pertaining to the state-space model of the environment, we may infer the following points:

1. If the probability density function of the noise terms in (4.1) is known, then the entropic state can be predicted one cycle into the future by using the Bayesian filtering framework of the perceptor.

2. The predicted entropic reward in the cognitive controller is then computed for the next hypothesized cycle.

In what follows next, this two-step procedure is illustrated in an example involving a Gaussian environment. This example will then be used in our computational experiment.

Predicting the Entropic Reward in a Gaussian Environment:

Consider a target with arbitrary dynamics in a Gaussian environment, with the state and measurement vectors denoted by $\mathbf{x}$ and $\mathbf{z}$, respectively. Since the noise terms in (4.1) are both Gaussian, the posterior $p(\mathbf{x}_k|\mathbf{z}_k)$ at each cycle is simply reduced to its mean value and covariance matrix. Let the entropic state be expressed by Shannon's

entropy of the Gaussian posterior Cover and Thomas (2006), namely:

$$H_{k|k} = \frac{1}{2}\log(\det\{(2\pi e)\mathbf{P}_{k|k}\}) \tag{3.12}$$

where $\det\{.\}$ denotes the determinant operator, and the matrix $\mathbf{P}_{k|k}$ is the covariance matrix of the posterior at cycle k, given the measurement also at cycle k. Since the logarithm is a monotonic function, (3.12) may be simplified to express the entropic state as follows:

$$H_{k|k} = \det\{\mathbf{P}_{k|k}\} \tag{3.13}$$

Based on this definition, a one-step *predicted* entropic state $H_{k+1|k} = \det(\mathbf{P}_{k+1|k})$ is found if we know the predicted covariance $\mathbf{P}_{k+1|k}$. To that end, the Kalman filter[7], operating as the perceptor, provides $\mathbf{P}_{k+1|k}$ simply by knowing the system noise covariance matrix $\mathbf{Q}_k$ and measurement noise covariance matrix $\mathbf{R}_{k+1}$ Bar-Shalom *et al.* (2001). Assuming that these two covariance matrices are given, we may compute the predicted entropic state of the perceptor. This process may be repeated to achieve further stages of prediction into the future, namely $H_{k+j|k}$, $j = 1, ..., l$, for l-step look-ahead horizon in time. Having all the $H_{k+j|k}$, predicted future rewards can then be calculated using equation (3.4), and we may therefore benefit from a planning process as well.

□

[7]In this context, if the process and/or measurement dynamics are nonlinear, then the Kalman filter may be replaced by a nonlinear version such as the extended Kalman filter (EKF), unscented Kalman filter (UKF), or cubature Kalman filter (CKF); the CKF will be employed in our computational experiment in Section IX.

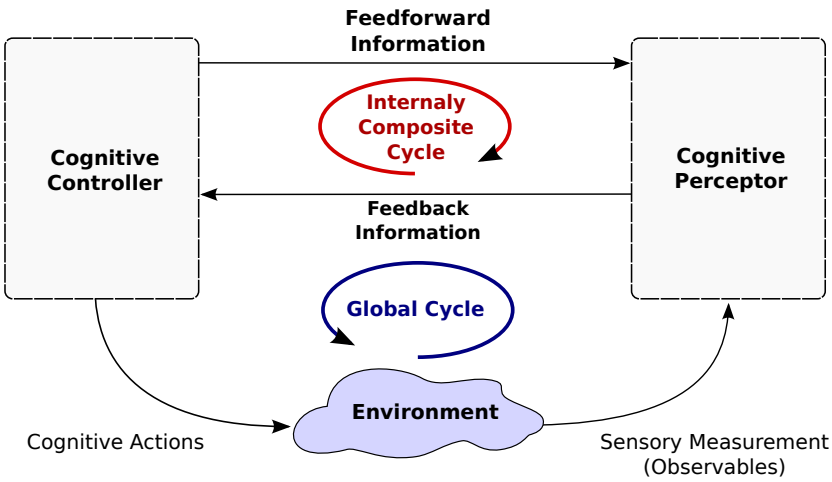


Figure 3.3: Block-diagram illustrating the combined presence of feedback information as well as feedforward information links.

The issue that emphasizes the need for planning is the time required for having actual rewards. In a cognitive dynamic system, we need to wait for one cycle to the next in order to access new rewards, and thereby proceed with the cognitive control learning algorithm, cycle by cycle. Unfortunately, Fig 4.1 lacks a *feedforward* link from the controller to the perceptor. In such a scenario with an action library involving M possible actions (i.e., $|\mathcal{C}| = M$), there would have to be M global perception-action cycles for exploring the complete action library. If the time T seconds are taken for each global perception-action cycle, then there would have to be MT seconds needed to cover the entire action library. In order to mitigate such a long-windowed exploration phase, we propose to introduce a *feedforward* link, which connects the controller to the perceptor, as depicted in Fig. 3.3. The feedforward information is a hypothesized future action, which is to be selected for a planning stage. In so doing, a new so-called *internally composite cycle* Haykin and Fuster (2014) is therefore created, which completely bypasses the environment. Accordingly, the duration τ taken by

such a cycle will be small compared to that of the global perception-action cycle, T. The practical benefit of introducing the internally composite cycle in Fig. 3.3 is the fact that the perceptor and the cognitive controller are now reciprocally coupled with each other, resulting in an exploration phase that is considerably shorter than in Fig. 4.1 by the factor T/τ.

Building on the scenario illustrated in Fig. 3.3, the two distinct but similar phases of learning and planning may now be implemented together, as follows:

1. **Learning**, which is based on actual values of the pair of entropic rewards at cycles k and $k-1$ as in (3.3) and (3.4), reproduced here for convenience of presentation:

$$g(|H_{k|k}|, \Delta_1 H), \quad \Delta_1 H = H_{k-1|k-1} - H_{k|k}$$

2. **Planning**, which is based on predicted values of the entropic reward; for example, at cycle $k+1$ and the actual reward at the current cycle k, we have the predicted reward defined by:

$$g(|H_{k+1|k}|, \Delta_2 H), \quad \Delta_2 H = H_{k|k} - H_{k+1|k}$$

Recall that learning is based on Lemma 1; equally, this lemma also applies to planning because conceptually speaking, both learning and planning perform the same required task. Note, however, learning is processed *only once* in each global perception-action cycle, which involves a single selected cognitive action; that is because learning is based on actual reward. On the other hand, in Fig. 3.3, planning is performed for any number of internally composite cycles and any number of hypothesized future actions in each of such cycles. Hence, specially in problems with very

large number of possible actions (compared to the number of global perception-action cycles), a cognitive controller with learning only and therefore no planning may not perform on average much better than random action-selection. It follows therefore that planning is an essential requirement for policy convergence.

3.8 Explore/exploit Tradeoff for Cognitive Control

As discussed previously in Section 3.6, in order to collect information about all the cognitive actions, the cognitive controller has to invest several global cycles, especially at the beginning of the experiment. During this phase, which is complete exploration of the cognitive action-space, the selected cognitive action in each cycle may result in completely poor performance. In particular, for problems with large set of cognitive actions, the resulting efficiency of the learning algorithm may remain unacceptable for a long period of time. Planning helps to mitigate this issue considerably, yet there is another auxiliary approach to smoothen the exploration process as much as possible, as discussed next.

In the cognitive control learning algorithm, and generally in dynamic programming, two different steps exist:

1. Updating the value-to-go function, J,

2. Updating the policy, π.

Note that updating J requires the knowledge of π, and vice versa. Hence, different approaches may be taken to update J and π, one after the other. When designing an

algorithm to shape the cognitive policy on a cyclic basis, the following two extreme approaches may then be taken:

- In the first approach, the cognitive controller will explore the entire action-space uniformly *without regard to* the value-to-go function as guidance. This strategy is called *pure explore*.

- In direct contrast, at each cycle, the cognitive controller may select an action that maximizes the value-to-go function J. This strategy is called *pure exploit*.

These two pure strategies are both extreme and clearly in conflict with each other. In reality, a *mixed strategy* is therefore desirable; namely, it is most of the time optimal in terms of value-to-go maximization, while at the same time, the strategy also involves exploration of other actions.

A commonly used mixed strategy as a compromise between the two mentioned pure strategies is called ϵ-greedy strategy Powell (2011), as follows:

- With the probability of ϵ (e.g., 5%), the cognitive controller selects action randomly (pure explore),

- With the probability of $1 - \epsilon$ (e.g., 95%), the cognitive controller selects action based on the maximum value criterion (pure exploit). In this case, the action selection is completely aligned with the value-to-go function, hence the term *greedy*.

Furthermore, in cognitive control, the explore/exploit tradeoff may be performed separately in two stages:

1. For the cognitive policy, we use an ϵ-greedy strategy, in which all the cognitive actions have the chance of being selected at least with a small but nonzero probability ϵ; this means most of the time, the policy is greedy but not always.

2. In the planning phase, instead of selecting m (out of $M = |\mathcal{C}|$) "random" cognitive actions, which is complete exploration, we may select the m cognitive actions based on some prior knowledge. In such a case, the selection of m cognitive actions is driven by some selection prior probability distribution based on the policy.

Deployment of the explore/exploit tradeoff in cognitive control may be viewed as a facilitator of *attention* as one of the basic principles of cognition. Therefore, a cognitive controller empowered with the explore/exploit tradeoff tries to allocate computational resources in such a way that it remains focused on the knowledge gained about the environment, but the controller does not fall into local optimal actions and thereby miss the big picture.

3.9 Structural Composition of the Cognitive Controller

Having the three constituents of perception, feedback information, and control, we may incorporate all three of them to propose a framework for cognitive control in a state-space modelled environment, as described next.

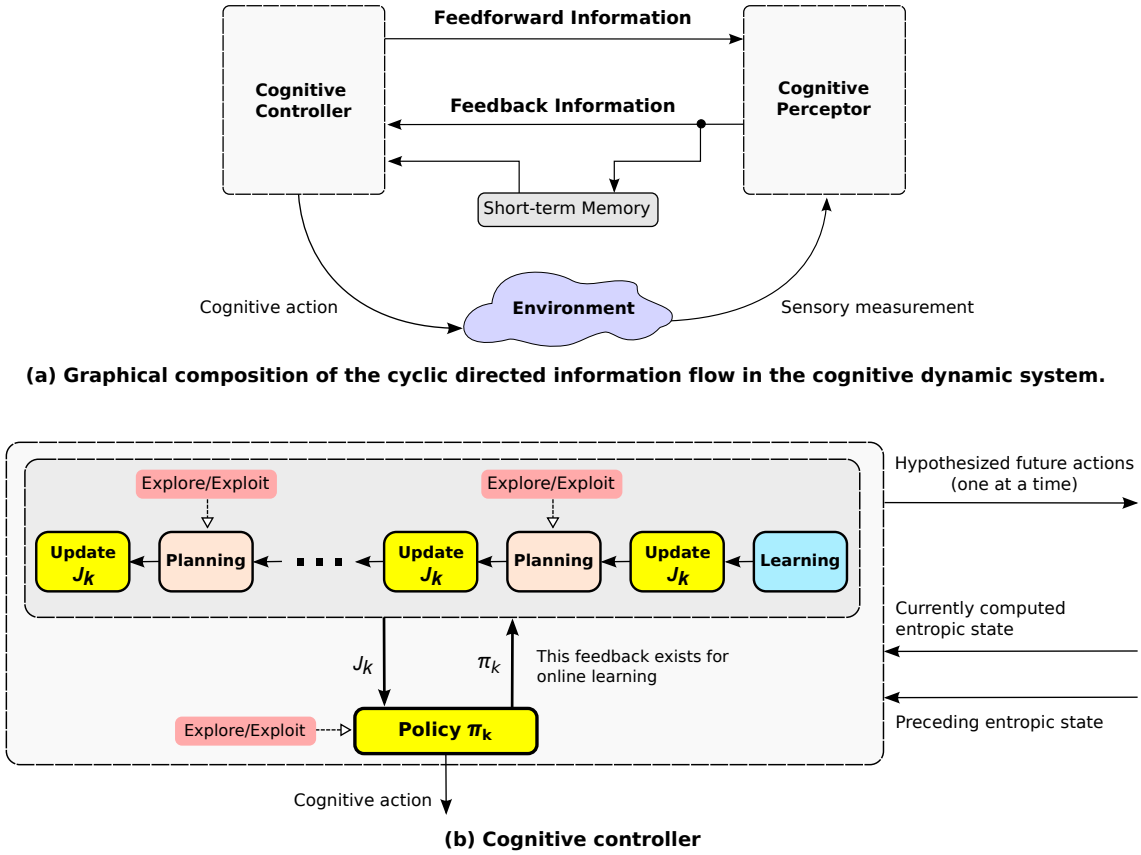


Figure 3.4: Block diagrammatic description of cognitive control: (a) cyclic directed information flow, and (b) illustration of algorithmic process in the cognitive controller.

Structure

To incorporate planning and learning, a basic and simple structure is suggested by Sutton and Barto, called Dyna Sutton and Barto (1998). However, Dyna lacks state-space modelling and the inclusion of Bayesian perception with cyclic directed information flow, required in cognitive control. Thus, inspired by Dyna and having cognitive control in mind, we propose a new structure depicted in Fig. 3.4. This structure consists of two parts: (a) and (b) for ease of understanding. A global perception-action cycle is initiated in the perceptor at the right-hand side of Fig. 3.4-a, where Bayesian perception is performed. The feedback information to be controlled will then be the entropic-state $H_{k|k}$, which is passed to the cognitive controller at the left-hand side of Fig. 3.4-a. At the same time, as explained in Remark 1, $H_{k|k}$ is also preserved in a short-term memory for the next cycle; it is short-term because in each cycle, the previous value will be overwritten. Then, in the cognitive controller, learning and planning are performed in the manner depicted in Fig. 3.4-b. It is noteworthy that in Fig. 3.4-b, the processes of learning and planning are performed in a serial manner[8]. To be specific, learning is performed, the result of which is an updated value-to-go function $J(c)$ for the preceding action. Then, we have a number of planning stages, each of which gives rise to a particular value-to-go update. In practice, the number of planning stages is dependant on the application of interest.

The explore/exploit tradeoff, explained in Section 3.8, is carried out in two different places: one place pertains to planning, and the other one pertains to policy-making. At the end, a cognitive action is selected from the derived policy and applied

[8]In the human brain, we have a similar scenario to that described in Fig. 3.4-b. Learning and planning use the same resources in the prefrontal cortex; where both learning and planning require organization in the time domain, with learning being current and planning being predictive Fuster and Haykin (2014).

to the environment; and with it, the next global perception-action cycle is initiated. This framework is indeed the underlying structure for implementing cognitive control.

Complete Algorithm

Algorithm 3 defines implementation of the cognitive controller, as described above, under Structure. "Updates" in lines 22 and 32 of the algorithm refer to the implementation of equation (3.7) for the currently selected cognitive action in learning and the hypothesized predictive action in planning, respectively. Also, line 30 in Algorithm 3 is implemented using the state-space model, as explained previously in Section II-A. Finally, the explore/exploit tradeoff is applied both in line 26 of the algorithm, where attention is deployed over some specific cognitive actions, namely the set $\mathcal{C}_1$, and the point where the cognitive policy π is shaped as ϵ-greedy in line 36 of the algorithm.

3.10 Computational Experiment: Cognitive Tracking Radar

In what follows, we will demonstrate the information-processing power of the cognitive controller applied to a cognitive radar system, where the emphasis is on tracking performance. To be specific, we consider the tracking of a falling object is space, using a radar with 10 measurements per second, based on the benchmark example presented in Haykin *et al.* (2011) and Haykin *et al.* (2012c). Here, the cognitive actions "change" the radar transmitter's waveform parameters on a cycle-by-cycle basis in order to correspondingly control noise in the receiver via the environment.

The target state is $\mathbf{x} = [x_1,\ x_2,\ x_3]^T$, where x_1, x_2 and x_3 denote the altitude,

velocity and ballistic coefficient, respectively; the ballistic coefficient depends on the target's mass, shape, cross-sectional area, and air density. The measurement vector $\mathbf{z} = [r,\ \dot{r}]^T$, consists of radar's range and range-rate. The extended state-space model is then defined by the following set of equations, involving both the state-space model as well as the entropic-state model:

$$\begin{cases} \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{v}_k \\ \mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k(\theta_{k-1}) \\ H_k = \det\{\mathbf{P}_{k|k}\} \end{cases}$$

where the vector θ_{k-1} refers to the waveform transmitted at the previous cycle, $k-1$. For details of the functions $\mathbf{f}(.)$ and $\mathbf{h}(.)$, the reader is referred to Haykin *et al.* (2011). Both noise terms, $\mathbf{v}_k$ and $\mathbf{w}_k$, are assumed to be white and zero-mean Gaussian. The system noise has the following covariance matrix Haykin *et al.* (2011):

$$\mathbf{Q} = \begin{bmatrix} q_1\frac{\delta^3}{3} & q_1\frac{\delta^2}{2} & 0 \\ q_1\frac{\delta^2}{2} & q_1\delta & 0 \\ 0 & 0 & q_2\delta \end{bmatrix}$$

where $q_1 = 0.01$, $q_2 = 0.01$, and $\delta = 1$. To model the measurement noise covariance matrix $\mathbf{R}$ as a function of waveform parameters, we use the model developed by Kershaw and Evans Kershaw and Evans (1994). There, it is shown that for the transmit waveform, combining linear frequency modulation with Gaussian amplitude

modulation, the measurement noise covariance matrix is defined by

$$\mathbf{R}(\theta_{k-1}) = \begin{bmatrix} \frac{c^2\lambda^2}{2\eta} & -\frac{c^2 b\lambda^2}{2\pi f_c \eta} \\ -\frac{c^2 b\lambda^2}{2\pi f_c \eta} & \frac{c^2}{(2\pi f_c)^2 \eta}(\frac{1}{2\lambda^2} + 2b^2\lambda^2) \end{bmatrix}$$

where, the constants f_c and η are the carrier frequency and the received signal-to-noise ratio (SNR), respectively, and $c = 2.9979 \times 10^8\ m/s$ is the speed of light. Finally, $\theta = [\lambda,\ b]^T$ is the waveform-parameter vector, which is adjustable by the cognitive controller for matching the transmitted waveform to the environment as closely as possible.

For the Bayesian filter in the perceptor, a cubature Kalman filter (CKF) Arasaratnam and Haykin (2009) has been used, which provides the estimated state covariance matrix $\mathbf{P}_{k|k}$ at cycle k. The entropic-state is then determined by $H_{k|k} = \det\{\mathbf{P}_{k|k}\}$, as in (13). For the entropic reward function, $r_k = |\log(|\Delta H|)|.\text{sgn}(\Delta H)$, with $\Delta H = H_{k-1|k-1} - H_{k|k}$ has been used, where $\text{sgn}(\cdot)$ denotes the standard signum function. This entropic reward also includes the right algebraic sign, which is required to guide the controller correctly. In the cognitive controller (i.e., radar transmitter), θ is changed at each perception-action cycle, which gives rise to 382 possible cognitive actions (382 is the number of different combinations for the transmit-waveform library). On each cycle, the cognitive action taken by the cognitive controller will affect the measurement noise covariance matrix. The time allowed for the experiment is five seconds for scenario 1 and 25 seconds for scenario 2; we therefore have to consider 50 and 250 perception-action cycles, respectively. All the simulations are performed over 1000 Monte Carlo runs to minimize the effect of randomness. It is also noteworthy that Algorithm 3, just like any other learning algorithm, is sensitive to the design

parameters; as such, it is important to fine-tune the parameters for a given problem of interest.

In what follows, we describe two different experimental scenarios, one dealing with planning and the other comparing three different controllers.

Scenario 1: The Impact of Planning on Cognitive Control

In this experiment, we conduct three distinct case-studies:

1. *Absence of cognitive control*, that is, there is no feedback information form the receiver to the transmitter. In effect, in so far as the receiver is concerned, the CKF acts entirely on its own. As illustrated in Fig. 3.5, the green diamond-line at the top of the figure refers to the fixed-waveform radar, where there is no cognitive action at all. Nevertheless, because the CKF is an integral part of the perceptor, the learning curve decreases almost two orders of magnitude in the course of 50 cycles.

2. *Cognitive learning with no planning*, in which the recursive algorithm of (9) operates on its own in the cognitive controller. As explained in Section 3.7, since the total number of cycles is far less than the entire number of possible cognitive actions (50 vs. 382), the red bar-line in Fig. 3.5 is not that much better than the case study involving the fixed transmit waveform.

3. *Cognitive learning with planning*, for which we retain learning, but this time we also add planning. Implementing *explore*-only in the planning phase (see Section 3.8), this third case-study is repeated for three different choices of $|\mathcal{C}_1|$ (see line 26 of Algorithm 1): (i) only one random cognitive action (blue triangle-line),

(ii) two random cognitive actions (black circle-line), and (iii) three random cognitive actions (cyan square-line). In the case of $|\mathcal{C}_1| = 1$, although one planning is still much less than the entire number of cognitive actions, it is enough to demonstrate a considerable improvement compared to the case with learning only. As for the other two cases, they both show more than four orders of magnitude improvement in the entropic-state reduction compared to the radar with fixed waveform.

Scenario 2: Comparison of Learning Curves of Three Different Cognitive Controllers

We refer back to the two different cognitive controller described in the Introduction, and compare them experimentally with the new cognitive controller described in this paper. Thus, the study involves the following three different configurations with the same cubature Kalman filter for the cognitive radar receiver (perceptor):

1. *Cognitive controller using dynamic optimization:* This optimization algorithm is a simplified version of Bellman's dynamic programming Haykin *et al.* (2011), in that it does not account for the future impact of the currently selected action. The reason is that at each perception-action cycle, we must compute the change in the entropic state for *all* the actions in the action-space. Therefore, from a practical perspective, the computational throughput of dynamic optimization is extremely high. To account for this practical difficulty, the depth of horizon is reduced to unity; in other words, there is no provision in looking into the future. Even so, the computational throughput is too heavy and therefore of limited

practical applications[9]. The learning curve of this first cognitive controller is depicted by the blue line in Fig. 3.6.

2. *Cognitive controller, using Q-learning as well as planning:* To be specific, the learning process in the cognitive controller is performed using the traditional Q-learning algorithm Watkins (1989); Watkins and Dayan (1992), which is made possible by exploiting the two-state model described in Section II-A. Moreover, the controller embodies planning with cardinality $|\mathcal{C}_1| = 3$. The learning curve of this second cognitive controller is depicted in Fig. 3.6 by the green line.

3. *The new cognitive controller,* which follows Algorithm 3. Specifically, it combines the use of the cognitive control learning algorithm as well as planning. The third and final learning curve (red line) in Fig. 3.6 accounts for the new cognitive controller. The planning part of this cognitive controller is also set to $|\mathcal{C}_1| = 3$. What is truly remarkable is the fact that the learning curve for the cognitive controller based on Algorithm 3 outperforms those of both Q-learning and dynamic optimization.

In this second scenario, the number of perception-action cycles has been set to 250 for the simple reason to allow for convergence to optimality.

It is important to note here that the numbers of floating-point operations (FLOPS) required for Algorithm 3 and Q-learning (both equipped with planning of $|\mathcal{C}_1| = 3$) are almost two orders of magnitude less than that of the method of dynamic optimization. Moreover, in the method of dynamic optimization, the computational load is unchangeable. In direct contrast, through the use of planning in Algorithm

[9]Appendix B shows that the method of dynamic optimization may indeed be derived as a special case of the proposed algorithm in this paper.

3 (involving the selection of the planning set $\mathcal{C}_1$), we have complete design flexibility. Specifically, we may move anywhere from learning-only (least optimal, most computationally efficient), to any desirable number of planning stages that remains computationally efficient. This significant practical property of the new cognitive controller provides an information processing power to match the engineering design of the cognitive controller to any problem of interest, where levels of optimality and available computational resources are both specified.

3.11 Conclusion

3.11.1 Cognitive Processing of Information

The new cognitive controller in a cognitive dynamic system is inspired by the brain on two fundamental accounts: learning and planning:

A.1 The learning process in cognitive control is based on two basic ideas:

- The *entropic state* of the perceptor, which makes it possible to bypass the imperfect-state information problem that arises in the brain and other cognitive dynamic systems, such as the cognitive radar Haykin *et al.* (2012c); Haykin and Fuster (2014).

- The *cyclic directed information flow*, which is attributed to the global perception-action cycle that defines the first principle of cognition Fuster (2003); Haykin (2012a).

A.2 The planning process in cognitive control: This second process is inspired by

the prefrontal cortex in the brain Fuster (2014); Haykin and Fuster (2014). Specifically, the cognitive controller in one side of the system is *reciprocally coupled* to the cognitive preceptor in the other side of the system. This reciprocal coupling, attributed to the combined use of *feedback information* from the perceptor to the controller as well as *feedforward information* from the controller to the perceptor, is the essence of the *shunt form* of perception-action cycle that completely bypasses the environment. In this paper we refer to this cycle as the internally composite cycle Haykin and Fuster (2014); most importantly, it is this particular form of the perception-action cycle that accommodates the use of planning in the cognitive controller.

3.11.2 Linearity, Convergence, and Optimality

These three intrinsic properties of the cognitive control learning algorithm are accounted for as follows:

- The linear law of computational complexity, measured in terms of actions taken on the environment, follows directly from the learning algorithm.
- Convergence and optimality of the learning algorithm follow from the proof that this algorithm is indeed a special case of the classic Bellman's dynamic programming.

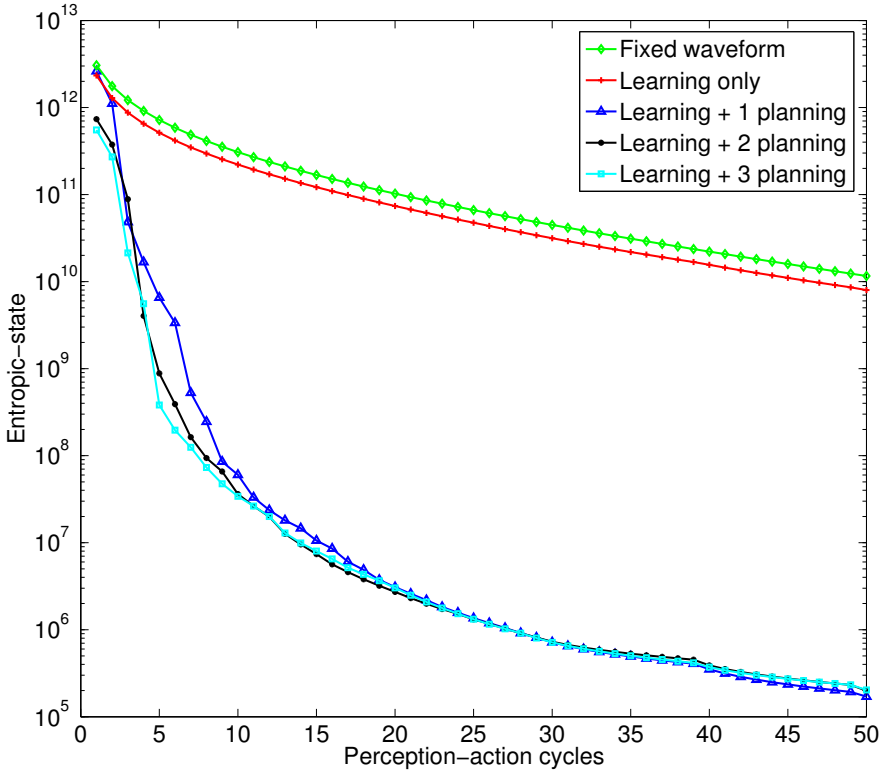


Figure 3.5: The impact of planning on cognitive control in Scenario 1.

3.11.3 Engineering Application

Practical validity of the new cognitive controller has been demonstrated experimentally in a cognitive tracking radar benchmark example. Specifically, the new cognitive controller has been compared against two other different sub-optimal cognitive controllers: One controller involves dynamic optimization that is computationally expensive; the other controller involves the use of traditional Q-learning that is computationally tractable, but inefficient in performance.

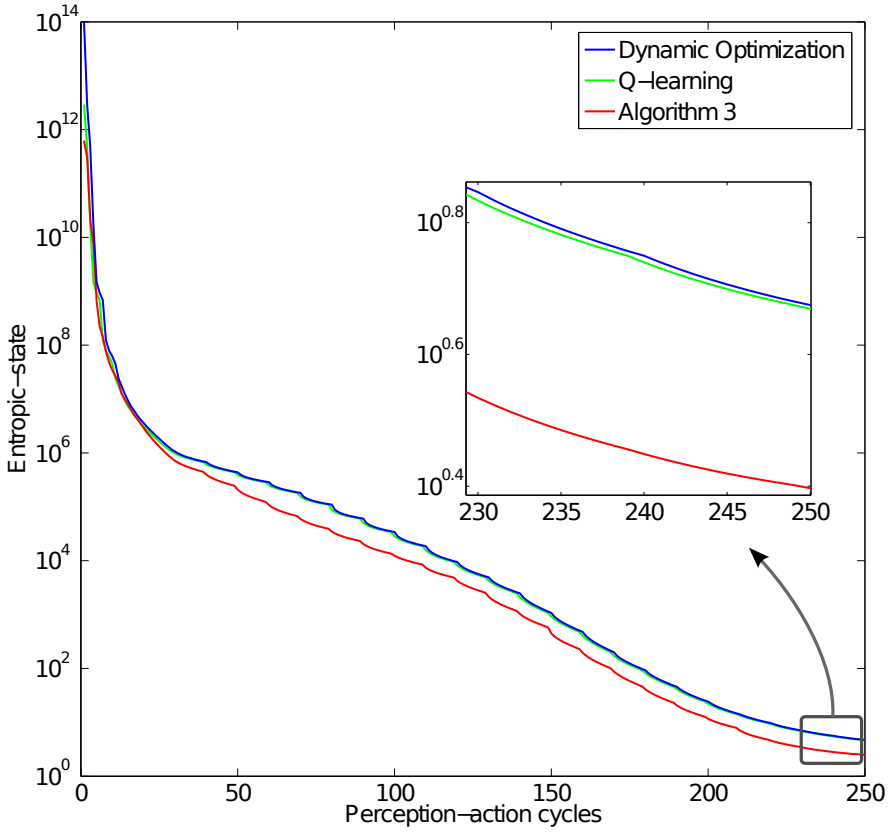


Figure 3.6: Comparative performance evaluation of three different cognitive control algorithms in Scenario 2.

Acknowledgment

The two authors of this paper deeply appreciate the encouraging statements made by the two anonymous reviewers of the *IEEE Access* journal on novelty of the cognitive controller described herein for the first time.

3.12 Appendix A

In this appendix, we derive Bellman's equation (3.10). The proof is along the same line as the proof of Lemma 1.

Using the linear property of the expected value operator as well as the total probability theorem Bertsekas and Tsitsiklis (2008), we may expand $\tilde{J}$ as follows:

$$
\begin{aligned}
\tilde{J}(s,a) &= \mathbb{E}^{\pi}[r_{k+1} + \gamma r_{k+2} + \gamma^2 r_{k+3} + ... \mid s_k = s, a_k = a] \\
&= \mathbb{E}^{\pi}[\Sigma_{j=0}^{\infty} \gamma^j r_{k+j+1} \mid s_k = s, a_k = a] \\
&= \sum_{s' \in \mathcal{S}} \mathbb{P}[s_{k+1} = s' | s_k = s, a_k = a] \times \mathbb{E}^{\pi}[\Sigma_{j=0}^{\infty} \gamma^j r_{k+j+1} | s_k = s, a_k = a, s_{k+1} = s'] \\
&= \sum_{s' \in \mathcal{S}} T^a_{ss'} \times \mathbb{E}^{\pi}[\Sigma_{j=0}^{\infty} \gamma^j r_{k+j+1} | s_k = s, a_k = a, s_{k+1} = s'] \\
&= \sum_{s' \in \mathcal{S}} T^a_{ss'} \times \{\mathbb{E}^{\pi}[r_{k+1} | s_k = s, a_k = a, s_{k+1} = s'] + \\
&\qquad\qquad \gamma \mathbb{E}^{\pi}[\Sigma_{j=0}^{\infty} \gamma^j r_{k+j+2} | s_k = s, a_k = a, s_{k+1} = s']\} \\
&= \sum_{s' \in \mathcal{S}} T^a_{ss'} \{R^a_{ss'} + \gamma \mathbb{E}^{\pi}[\Sigma_{j=0}^{\infty} \gamma^j r_{k+j+2} | s_k = s, a_k = a, s_{k+1} = s']\} \\
&= \sum_{s' \in \mathcal{S}} T^a_{ss'} \{R^a_{ss'} + \gamma \sum_{a' \in \mathcal{A}} \mathbb{P}[a_{k+1} = a' | s_k = s, a_k = a, s_{k+1} = s'] \times \\
&\qquad\qquad \mathbb{E}^{\pi}[\Sigma_{j=0}^{\infty} \gamma^j r_{k+j+2} | s_k = s, a_k = a, s_{k+1} = s', a_{k+1} = a']\} \\
&= \sum_{s' \in \mathcal{S}} T^a_{ss'} \{R^a_{ss'} + \gamma \sum_{a' \in \mathcal{A}} \tilde{\pi}_k(s,a') \tilde{J}(s',a')\}
\end{aligned}
$$

□

3.13 Appendix B

In this appendix, we show that dynamic optimization used in the cognitive radar Haykin *et al.* (2011), it may be considered as a special case of the cognitive control learning algorithm, introduced in this paper.

At each perception-action cycle, the cost-function in the dynamic optimization algorithm is equivalent to the entropic state in this paper, and it is predicted for all the actions in the action library, using the Kershaw and Evans model Kershaw and Evans (1994). The action that has the minimum cost is then selected as the optimal action.

Turning back to cognitive control, recall the learning update in Algorithm 1 (line 11):

$$J(c) \leftarrow J(c) + \alpha[\mathcal{R}(c) + \gamma \Sigma_{c' \in \mathcal{C}} \pi_k(c, c') J(c') - J(c)]$$

Substituting $\alpha = 1$ and $\gamma = 0$ yields the following:

$$J(c) \leftarrow \mathcal{R}(c) \tag{3.14}$$

which implies that under the assumptions of $\alpha = 1$ and $\gamma = 0$, the value-to-go function in cognitive control turns into the immediate reward.

Next, consider the substitution of $\mathcal{C}_1 \leftarrow \mathcal{C}$, in algorithm 3 (line 26). This case is equivalent to having *complete* planning at each perception-action cycle, which is clearly a possible choice.

Combining the learning and planning processes, discussed above, we have then exactly the same algorithm as dynamic optimization. To be more specific, in the

case of having complete planning with unitary learning factor and no future inclusion (zero-discount), the new cognitive controller is reduced to dynamic optimization, and therefore the new cognitive controller embodies features that do not exist in dynamic optimization. Hence, it is not surprising that in Scenario 2 of Section IX, the learning curve for dynamic optimization deviates from that of the new cognitive controller.

Algorithm 3: A complete algorithm to implement Lemma 2, which embodies both learning and planning.

Varables:
$\mathcal{C}$:= set of all cognitive actions
$\mathcal{C}_1$:= set of selected cognitive actions for planning
J := value-to-go function
π := control policy
$memLearning$:= short-term memory for learning
$memPlanning$:= short-term memory for planning
c := selected cognitive action
r := computed reward
k := time step
Initialization:
$k \leftarrow 0$;
$memLearning \leftarrow H_0$;
$c \leftarrow$ a random cognitive action;
Apply c to the environment;
repeat
    $k \leftarrow k + 1$;
    $H_{k|k} \leftarrow Input$(entropic_state) from Perceptor;

    Learning:
    $r \leftarrow g_k(H_{k|k},\ (memLearning - H_{k|k}))$;
    Update J;
    $memLearning \leftarrow H_{k|k}$;

    Planning:
    Select $\mathcal{C}_1 \subseteq \mathcal{C}$;
    for *all cognitive actions* $c \in \mathcal{C}_1$ **do**
        for *i=1 to num_prediction_steps* **do**
            $memPlanning \leftarrow H_{k+i-1|k}$;
            compute $H_{k+i|k}$ using c;
            $r \leftarrow g_k(H_{k+i|k},\ (memPlanning - H_{k+i|k}))$;
            Update J;
        end
    end

    Update π by J;
    Select c based on π;
    Apply c to the environment;
until *perception-action cycles are finished*;

The following chapter is a reproduction of a paper submitted to an IEEE journal for publication:

Fatemi M., Setoodeh, P., and Haykin, S.
Improving Observability of Stochastic Complex Networks under the Supervision of Cognitive Dynamic Systems, *IEEE Transactions on Network Science and Engineering*, paper submitted, October 25, 2014.

Chapter 4

Improving Observability of Stochastic Complex Networks under the Supervision of Cognitive Dynamic Systems

4.1 Abstract

Much has been said about observability in system theory and control; however, it has been recently that observability in complex networks has seriously attracted the attention of researchers. This paper examines the state-of-the-art and discusses some issues raised due to "complexity" and "stochasticity". These unresolved issues call for a new practical methodology. For stochastic systems, a degree of observability may be defined and the observability problem is not a binary (i.e., yes-no) question

anymore. Here, we propose to employ a goal-seeking system to play a supervisory role in the network. Hence, improving the degree of observability would be a valid objective for the supervisory system. Towards this goal, the supervisor dynamically optimizes the observation process by reconfiguring the sensory parts in the network. A cognitive dynamic system is suggested as a proper choice for the supervisory system. In this framework, the network itself is viewed as the environment with which the cognitive dynamic system interacts. Computer experiments confirm the potential of the proposed approach for addressing some of the issues raised in networks due to complexity and stochasticity.

4.2 Introduction

In 1977, Herbert Simon wrote Simon (1977), p. 258:

> "To a Platonic mind, everything in the world is connected to everything else—and perhaps it is. Everything is connected but some things are more connected than others."

The point he is emphasizing is *connectivity*, which is at the heart of complex networks. Indeed, the complexity of networks manifests itself in how dense and with what kind of structure, the edges are distributed in a network with arbitrary large number of nodes.

In the realm of network science, an extremely important issue to be addressed in many of real-world applications is how to acquire sufficient information about a network with minimal computational effort. In other words, the problem of interest is to understand why a complex network with high connectivity behaves in a certain

way by accessing only a small subset of nodes. This problem is subsumed under the broader problem of network *observability*. Knowing whether or not a network is observable would be critical because in large networks, it is often impractical or even impossible to monitor all nodes' states. On the other hand, in many real-world applications, all the states are not necessarily accessible to the outside world. Therefore, there is a need to reconstruct (i.e., estimate) those states on the sole basis of observing other variables, which are related to those states as well as accessible for measurements.

Although it is a classic and well-known problem in system theory and control, observability in network science is relatively new, mostly started with the prominent work reported in Liu *et al.* (2013). For deterministic networks, the proposed algorithm in Liu *et al.* (2013) yields the minimum number of nodes in the network that should be monitored in order to satisfy the requirement for observability. It also provides subsets of nodes from which the monitor nodes should be selected. However, extending the results to stochastic networks aiming at estimating the state of the network does not seem to be that straightforward. As a matter of fact, having the proposed framework of Liu *et al.* (2013) in mind, some of the simulation results obtained for complex stochastic networks (especially those with dense structures) may seem counterintuitive. Hence, for estimating a network's state in face of model uncertainties and imperfect measurements, additional steps must be taken. We distinguish our paper from Liu *et al.* (2013) in two accounts: a) complexity in terms of edge density, and b) stochasticity.

Here, we propose to implement a *controlled-sensing* mechanism in the network. By taking this approach, a supervisory system would be responsible for *reconfiguration*

of the sensory parts in the network in order to dynamically *optimize the observation* process. A *cognitive dynamic system* (CDS) in the sense described in Haykin (2012a) will be able to perfectly play the role of the supervisory system, where the stochastic network of interest is viewed as the environment with which the CDS interacts. A CDS is built on Fuster's paradigm of cognition, which suggests five pillars for a cognitive system: perception-action cycle, memory, attention, intelligence, and language Fuster (2003). Perceptual and executive parts of the perception-action cycle as well as memory are physical entities, attention is algorithmic, and intelligence emerges due to the interactions among the former three pillars. Language will play a key role, when we have a network of cognitive systems.

Following this new way of thinking, the CDS, which acts as a supervisor over a given network of interest, tries to reconstruct the hidden states of the network based on the information it gathers from monitor nodes (i.e., a selected subset of nodes whose outputs are accessible to the CDS). The perceptual part of the CDS employs Bayesian filtering for reconstruction of entire state of the network. Furthermore, through the use of *cognitive control* Haykin *et al.* (2012a); Fatemi and Haykin (2014), the executive part of the CDS tries to improve accuracy of the reconstructed state from each global cycle of perception-action to the next. To this end, a quantitative measure for the lack of information in the state posterior is also computed in the cognitive perceptor, which is passed on to the cognitive controller as the feedback information. The cognitive controller will then use this information to reconfigure the sensory parts of the network by *rearranging* the monitor nodes in such a way that the available information to the perceptor is maximized in the following cycles. In addition to rearrangement of monitors, the CDS may also have to increase the number

of monitor nodes or remove redundant ones (i.e., nodes with minimal contribution in acquiring information).

In the proposed approach, learning and planning stages involved in cognitive control provide enough flexibility to handle different situations that may occur in the network of interest. In this regard, a few points are worth mentioning:

- Cognitive control can directly incorporate any practical constraints such as limitation on the number of nodes that can be monitored (i.e., number of deployed sensors).

- Due to design parameters such as learning and discount factors as well as size and depth of the planning stage, the methodology can be adapted for different practical applications.

- The required computations for implementing the proposed methodology can be performed either online or partially offline:

    i) In the online implementation, cognitive controller and Bayesian filter find the best set of monitor nodes taking some prescribed constraints into account. Moreover, the selection process happens in a cyclic manner from each global cycle of perception-action to the next.

    ii) In the partially offline implementation, the proposed methodology is used as the basis for Monte Carlo simulations, which provide clues about the best set of monitor nodes considering the practical constrains. Here, the preferred sets of monitor nodes for different working conditions are found and stored beforehand. This way, an appropriate set of monitor nodes for current operational conditions will be recalled from the stored data and

therefore the amount of computation that must be performed on the fly will significantly decrease.

The rest of the paper is organized as follows: Section 4.3 reviews some basic concepts from network science as the required background for the following sections. Next, in Section 4.4, the problem of stochastic observability is discussed in detail with emphasis on network observability. Section 4.5 explains how complex networks can be viewed as the environment with which a cognitive dynamic system interacts. This way, the CDS plays the role of a supervisor that is responsible for improving network observability. Advantages of the proposed approach are shown through a set of computationl experiments in Section 4.6 for both linear and nonlinear case studies. Finally, Section 4.7 concludes the paper by highlighting the key results and drawing lines for future research.

4.3 Brief Account on Network Science

Regarding the critical role that networks play in shaping and sustaining our modern societies, the study of *complex networks* has been expanding across diverse scientific disciplines over the last two decades Cohen and Havlin (2010). This relatively new branch of science has began to be referred to as *network science*.

4.3.1 Networks with Stochastic Dynamics

A number of entities that have interactions with each other (i.e., linked in a physical and/or mathematical sense) form a *network*. The underlaying *topology* of a network is mathematically described by a graph, where each node represents one entity and

edges show the interactions between the nodes they connect. Moreover, each entity in the network (i.e., each node in the graph) attributes to a *state*. In reality, each state is a realization of a physical quantity, such as the electrical load on a power station, the density of a chemical compound in a biomedical receiver, or the amount of an item in a warehouse.

In order to be mathematically precise, the following set of definitions are recalled from graph theory Cohen and Havlin (2010):

Definition (digraph). A *digraph* (directed graph) $G(N, L)$ is determined by a pair of sets:

1. A set of nodes, N with $|N| = n$, where n is called the graph size.

2. A set of directed edges:

$$L = \{(i, j) \text{ iff there exists an edge from } i \text{ to } j \text{ for } i, j \in N\}.$$

Definition (incident matrix). The *weighted incident matrix* (or simply incident matrix) of digraph $G(N, L)$ is a square matrix, $\mathbf{A} \in \mathbb{R}^{n \times n}$, which has a row and a column for each node. If there is a link from node i to node j in the digraph, the corresponding element of the incident matrix $\mathbf{A}_{ij}$, which represents the dependency *weight* of node j on i, will be nonzero. Otherwise, the entry $\mathbf{A}_{ij}$ will be zero. Hence, in general, the incident matrix of a digraph is asymmetric. If there are edges in the network that connect some nodes to themselves (i.e., if self-loops exist in the digraph), the corresponding diagonal elements of A will be nonzero.

A network may represent a linear stochastic system that satisfies the Markovian assumption. In this case, the following pair of process and monitor equations provide

a dynamic model for the network:

$$\begin{cases} \text{Process equation:} & \mathbf{x}_{k+1} = \mathbf{A}_k^T \mathbf{x}_k + \mathbf{v}_k \\ \text{Monitor equation:} & \mathbf{z}_k = \mathbf{C}_k \mathbf{x}_k + \mathbf{w}_k \end{cases} \tag{4.1}$$

where $\mathbf{A}_k$ is the incident matrix of the corresponding digraph G at cycle k that may vary in the course of time and T denotes the transpositional operator. In this model, state vectors of the digraph's nodes, $\mathbf{x}_k^{(i)}$, are concatenated to form the augmented vector $\mathbf{x}_k = \{\mathbf{x}_k^{(i)}\}_{i=1}^n$ that represents the whole network state. The evolution of the network state in the course of time is governed by the above process equation in which the process noise, $\mathbf{v}_k$, takes account of model uncertainties. It is assumed that a subset of nodes, $M \subseteq N$ with $|M| = m \leq n$, is available for monitoring, from which $q \leq m$ nodes are chosen as monitor nodes. Therefore, there are $\binom{m}{q} = \frac{m!}{q!(m-q)!}$ different options for choosing q monitor nodes from m accessible nodes. Similarly, the observed outputs of the monitored nodes, $\mathbf{z}_k^{(j)}$, are concatenated to form the augmented measurement vector $\mathbf{z}_k = \{\mathbf{z}_k^{(j)}\}_{j=1}^q$. The above monitor equation [1] describes the relationship between the state and measurement vectors, where the measurement noise, $\mathbf{w}_k$, takes account of measurement uncertainties. Matrix $\mathbf{C}_k$, has a row associated with every output of every monitor node at cycle k.

For the sake of brevity, in this paper, we assume that the random processes $\mathbf{v}$ and $\mathbf{w}$ are both zero-mean, white and mutually independent. Also, we solely focus on Gaussian environments, i.e., $v_k \sim \mathcal{N}(0, \mathbf{Q}_k)$ and $w_k \sim \mathcal{N}(0, \mathbf{R}_k)$, where $\mathbf{Q}_k$ and $\mathbf{R}_k$ denote the *covariance* matrices of process and monitor noises, respectively. However,

[1] In the control literature, the second equation in (4.1) is called measurement or output equation. In the network context, the measurements (i.e. observables) are provided by nodes that are chosen to be monitored (i.e. monitor nodes). Hence, the term monitor equation was adopted.

the application of cognitive control is not limited to Gaussian models.

The same modelling philosophy can be equally applied to stochastic nonlinear systems regarding the fact that for a nonlinear system, as discussed in Liu *et al.* (2013), there exists a unique inference diagram (i.e., a digraph). In such cases, the network, which is mathematically described by the corresponding inference graph, represents the stochastic nonlinear system under study. The state-space model will then take the following form:

$$\begin{cases} \text{Process equation:} & \mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{v}_k \\ \text{Monitor equation:} & \mathbf{z}_k = \mathbf{g}_k(\mathbf{x}_k) + \mathbf{w}_k \end{cases} \tag{4.2}$$

If we have direct access to the states of nodes that are monitored, the above nonlinear monitor equation will be reduced to a linear one.

The linear and nonlinear state equations in (4.1) and (4.2) are discrete-time models. The developed framework can be equally applied for continuous-time processes. However, in such cases, a hybrid (i.e., continuous-discrete) version of the Bayesian filter would be required for network state reconstruction.

Now that we have covered the relationship between state-space models, digraphs, and networks, we need to know how *edge distribution* as well as *edge density* affect the observability of complex networks if the number of nodes does not change. Answers to these questions will help for better sensor design in real-world applications. In order to set the stage for answering these key questions, we take a look at some well-known network topologies.

4.3.2 Two Basic Network Topologies of Practical Importance:

Among different classes of networks, we consider Erdős-Rényi and scale-free random networks for their importance in modelling real-world networks Jackson (2008):

Erdős-Rényi (ER) Networks

Named after P. Erdős and A. Rényi Erdős and Rényi (1960), in the growing process of this class of networks, a connection (an edge in the graph) may be produced between each pair of nodes with equal probability p, independent of the other edges. In their seminal paper, Erdős and Rényi provided a detailed behavioural analysis for such networks for different values of p. As a result, ER networks have become the most basic class in complex-network studies.

Scale-free Networks

A scale-free network is a network whose degree distribution follows a power law, at least asymptotically. To be more precise, let the fraction of nodes in the network that have k connections to other nodes be denoted by $P(k)$. Then, for large values of k we will have $P(k) \sim k^{-\gamma}$, where γ is a parameter whose value is typically in the range $\gamma \in (2, 3)$, although occasionally it may lie outside of this interval Barabási and Albert (1999); Choromański *et al.* (2013). Many of the real-world networks are thought to be somehow scale-free. Examples include social and collaborative networks, internet networks including the World Wide Web, some financial networks, protein-protein interaction networks, and airline networks.

Next section provides a formal definition of observability in the context of stochastic networks and suggests a way for improving the observability.

4.4 Observability of Stochastic Complex Networks

Talking about complexity, it is noteworthy to distinguish between three stages of system structure: *simple*, *complicated*, and *complex* Cotsaftis (2009). Simple systems are the building blocks for both complicated and complex systems Milo *et al.* (2002). The difference between complicated and complex systems is due to the fact that in the latter, interactions between system components are fairly strong and somehow overshadow the component features. As a result, while a *reductionist* approach may work for analyzing complicated systems, for complex systems, taking a *holistic* approach is a must Cotsaftis (2009). In networks, moving from a sparse structure towards a dense structure can be interpreted as passing from a complicated network to a complex network.

When it comes to networks, in different branches of science and engineering, it is common to deal with sequential data gathered from the network. A large portion of our knowledge about a network, especially when it is large-scale and complex, cannot be presented in terms of quantities that can be measured directly. In such cases, building a model would be the logical basis for explaining the cause behind what we observe via the measurement process. This leads us to the notions of *state* and *state-space* model of a dynamic network, where the term "dynamic" may refer to time evolution of node state Liu *et al.* (2011a), edge state Nepusz and Vicsek (2012), a combination of both, or even size and topology of the network Setoodeh and Haykin (2009).

To investigate *reconstructing* (i.e. *estimating*) the state of dynamic networks from measuring the outputs of its monitor nodes, a key question is whether or not it is possible to do so using a given model of the dynamic network under study. This

critical question that must be answered before choosing a proper estimation algorithm among different candidates, leads us to the concept of *observability* Muske and Edgar (1997). For deterministic networks, observability implies that an observer would be able to distinguish between different initial states based on measurements. In other words, an observer would be able to uniquely determine observable initial states from measurements Kailath (1980).

For defining observability in the context of networks, we may need a paradigm shift from the classic state trajectories to more abstract trajectory manifolds Cotsaftis (2009). To be more precise, in estimating the state, we may settle for finding a restricted initial subspace of the original state space instead of an individual initial stateLiu *et al.* (2011b).

In Liu *et al.* (2013), Liu, Slotine, and Barabasi proposed an intuitive method, which provides possible sets of necessary monitor nodes in a "deterministic" network, according to the Jacobian-based definition of observability. Additionally, they mentioned that any of the given sets may be sufficient in some specific cases. The method, which is called LSB hereafter, is based on a graph theory concept, known as strongly connected component (SCC). An SCC is a subgraph, in which there exists a directed path from each node to every other nodes. Although it is easy to implement the LSB algorithm, here are a few points that are worth thinking about:

- LSB results in a number of sets (called root SCCs), from each of which a node should be selected as one of the monitor nodes. However, LSB does not provide any further information about which of the nodes in each root SCC would be a better monitor node.

- For most of dense and almost uniformly edge-distributed networks (e.g., Erdős-Rényi networks), which simply has one or a few SCCs, the LSB method hardly provides any practical clue about the monitor nodes. Table 4.1 elaborates on this critical problem.

- LSB is meant for deterministic networks, where the network model is completely known and the observation of monitor nodes is assumed to be perfect. It is mentioned in Liu *et al.* (2013) under the suggested future research topics that both assumptions may be violated in practice, where model uncertainties and measurement imperfections are involved. Indeed, our experiments demonstrate that for problems with modelling and measurement uncertainties, the practical monitor nodes may be different from what are suggested by LSB.

More importantly, in many practical cases, we are limited in the number of monitor nodes due to different reasons including limited computational resources. In a problem with limited number of monitor nodes, LSB provides no preference among the suggested monitor nodes and may therefore be used only as a clue for the selection of the monitor nodes.

Going one step further to address real-world problems, the issue of stochasticity deserves special attention. In different applications, it is often desirable to predict next states based on collected data up to a certain time instant. Since the future is always uncertain, it is also preferred to have a measure that shows our confidence about the predictions; a probability distribution over possible future outcomes will do the job Murphy (2002).

For stochastic systems, there is not a unique definition of observability. However, most of the proposed definitions for observability of stochastic systems have roots

Table 4.1: Number of monitor nodes based on the Liu-Slotine-Barabasi (LSB) method are compared for two basic network topologies of the same size, namely, scale-free and Erdős-Rényi (ER) random networks. Each row has roughly the same number of edges and the number of monitors are averaged over 1000 realizations and rounded up. For the scale-free networks, α, β, and γ are respectively the probabilities of adding a new node connected to an existing node chosen randomly according to the in-degree distribution, adding an edge between two existing nodes (one existing node is chosen randomly according to the in-degree distribution and the other is chosen randomly according to the out-degree distribution), and adding a new node connected to an existing node chosen randomly according to the out-degree distribution. Clearly, with the same number of nodes, the more dense the network is, the less number of monitor nodes is suggested by LSB. Similarly, LSB suggests considerably less number of necessary monitors for more uniformly-distributed networks. It is also noteworthy that for ER random networks, which are more dense than 5%, LSB provides almost no information about the monitor nodes. A similar problem happens for dense scale-free networks as well.

		Scale-free		**ER Random**	
Number of nodes	Average number of edges	Parameters	Avg. LSB monitors (± 1)	Probability for edge creation ($\in [0,1]$)	Avg. LSB monitors (± 1)
100	210	$\alpha = 0.41$ $\beta = 0.54$ $\gamma = 0.05$	74	0.021	12
100	370	$\alpha = 0.21$ $\beta = 0.74$ $\gamma = 0.05$	67	0.037	3
100	600	$\alpha = 0.41$ $\beta = 0.54$ $\gamma = 0.05$	56	0.060	1
100	1620	$\alpha = 0.05$ $\beta = 0.94$ $\gamma = 0.01$	1	0.162	1

in information theory. For instance, in Liu and Bitmead (2011), observability was defined on the basis of the concept of *mutual information*:

$$I(\mathbf{X};\mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}). \tag{4.3}$$

where $H(\mathbf{X})$ denotes *entropy* of $\mathbf{X}$ and $H(\mathbf{X}|\mathbf{Y})$ is defined as the entropy of random variable $\mathbf{X}$ (i.e. state vector) conditional on the knowledge of random variable $\mathbf{Y}$ (i.e. measurement vector), hence the term *conditional entropy*. According to Liu and Bitmead (2011), state $\mathbf{X}$ is unobservable from measurement $\mathbf{Y}$, if they are independent or equivalently $I(\mathbf{X};\mathbf{Y}) = 0$; otherwise, $\mathbf{X}$ is observable from $\mathbf{Y}$. Since mutual information is nonnegative, equation (4.3) leads to the following conclusion: if either $H(\mathbf{X}) = 0$ or $H(\mathbf{X}|\mathbf{Y}) < H(\mathbf{X})$, then $\mathbf{X}$ is observable. A deterministic system is either observable or unobservable but for stochastic systems, a *degree of observability* can be defined, which varies between 0 and 1 Kam *et al.* (1987).

Referring back to networks, in general, two sets of states can be considered for a network: *physical states* and *information states*, which are associated to physical dynamics and information dynamics, respectively Hero and Cochran (2011). In Haykin *et al.* (2012a), the notion of cognitive controller was proposed for controlling the information state as a counterpart to physical controller that controls the physical state. In the proposed framework, cognitive and physical controllers play complementary roles.

This paper proposes a systematic method for improving observability and therefore the quality of physical-state estimates in stochastic networks, based on the previously mentioned notion of stochastic observability. Cognitive controller will be able to increase the degree of observability, if

- the measure of information, which is chosen as the information state, is the entropy of the physical state (i.e. the entropic state), and
- the role of cognitive controller is defined to minimize this entropy.

In this setup, a cognitive perceptor computes the mentioned entropic state and thereby sets the stage for cognitive control. As a result, the cognitive controller operates as the information supervisor of the network to address the previously mentioned issues of concern in a cycle-by-cycle manner. To be more precise, the cognitive controller algorithmically chooses the best set of monitor nodes from one cycle of perception-action to the next in a way to reduce the conditional entropy. In case of complete observability, the entropic state will approach zero Fatemi and Haykin (2014); however, this is not the case in practice due to the ever presence of uncertainty and modelling imperfections.

4.5 Complex Networks Viewed as the Environment of Cognitive Dynamic Systems

Much has been written about the relationship between neuroscience and engineering. However, when it comes to cognitive neuroscience with emphasis on cognition, the Cognitive Dynamic System (CDS) first described in Haykin (2006a) and later expanded in Haykin (2012b), is the closest description of such a system viewed from the perspective of Fuster's principles of cognition Fuster (2003). Fuster's principles are discussed in the Introduction; however, from the perspective of this article, it is the perception-action cycle that is the center of focus.

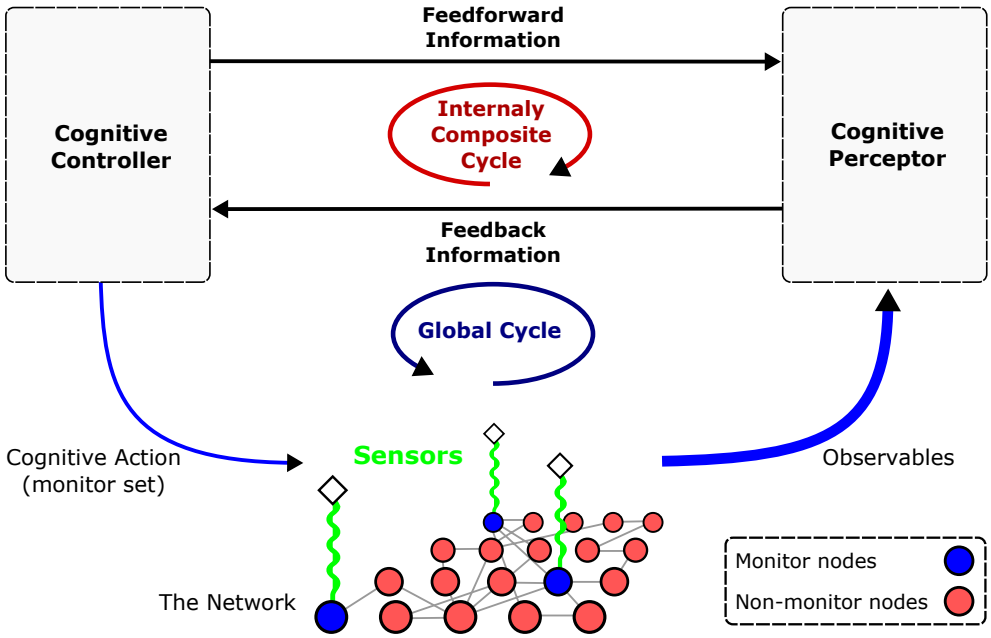


Figure 4.1: Block diagram of the global perception-action cycle over a network, where a cognitive dynamic system acts as a supervisor. The nodes in blue are the monitor nodes and the diamonds are the observables.

In the context of CDS, the environment is generic in terms of being an entity with any number of hidden states, which is seen only though the observables. As a result, it is quite natural to consider networks as the environment of a CDS with observables being the outputs of monitor nodes. In the structure depicted in Fig. 4.1, the illustrated cognitive dynamic system indeed acts as a supervisor over a given network of interest in that it reconstructs the hidden state of the network on the sole basis of observing the monitor nodes. Furthermore, through the use of cognitive control, the cognitive dynamic system guarantees the accuracy of reconstructed state from each global cycle of perception-action to the next. In the following subsection, we first describe Bayesian perception of the network, which directly results in the definition of the so-called *entropic state* that accounts for the mentioned information

state. Then, the next two subsections discuss the cyclic directed information flow, which and the algorithmic processes involved in cognitive control.

4.5.1 Bayesian Perception of Networks: The Two-state Model

We begin the global perception-action cycle by focusing on the perceptor on the right-hand side of Fig. 4.1. The function of the perceptor is to monitor the network separately from the controller, and reconstruct the network state on the sole basis of extracting information from the observables. To be more specific, we may look to *Bayesian filtering* Ho and Lee (1964) for estimating the hidden state of the network; using a state-space model (4.1) or (4.2) that consists of a pair of equations: (a) *process equation* that describes evolution of the state over time, which is contaminated by system noise, and (b) *monitor equation*, which describes dependence of the incoming observables on the state of monitor nodes, corrupted by measurement noise. Optimal solution of the state estimation problem is given by the well-known *Bayesian filter* Ho and Lee (1964), at least in conceptual terms, which includes the special but important case of Kalman filter and its nonlinear versions Kalman (1960); Bar-Shalom *et al.* (2001); Crisan and Rozovskii (2011). In a more general fashion, also for non-Gaussian environments, particle filters might be preferred to approximate the optimal Bayesian filter Gordon *et al.* (1993); Ristic *et al.* (2004); Robert and Casella (2005). If we have a continuous-time process, then the Bayesian perceptor will take the form of a hybrid filter due to the fact that the observation process is still discrete in time.

As discussed in Fatemi and Haykin (2014), the two-state model is an essential element in deriving the cognitive control algorithm. By definition, the two-state model embodies two distinct states, one of which is called the *network state*, that is

the vector of all the states attributed to the nodes (or edges or both) of the network. The second one is called the *entropic state* of the perceptor, the source of which is attributed to the unavoidable presence of uncertainties in the environment as well as imperfections in the perceptor itself. These two states exactly corresponds to the "physical" and "information" states, which were previously discussed. Insofar as cognitive control is concerned, the two-state model is described in two steps as follows:

1. State-space model of the network, which is described by (4.1) or (4.2).

2. Entropic state model of the perceptor, which is formally defined by the following equation:

$$\text{Entropic-state equation:} \quad H_k = \phi(p(\mathbf{x}_k|\mathbf{z}_k)) \tag{4.4}$$

   The H_k is the entropic state at cycle k in accordance with the state posterior $p(\mathbf{x}_k|\mathbf{z}_k)$ in the Bayesian sense, which is computed in the perceptor[2]. As such, H_k is the state of the perceptor and ϕ is a quantitative measure such as Shannon's entropy[3].

[2]To emphasize the cycles, in which the state and the measurement are taken, in this paper, we may also use the notation $H_{k|k}$, in accordance with the subscripts in the posterior, $p(\mathbf{x}_k|\mathbf{z}_k)$.

[3]Shannon's entropy for a random variable X, having the probability density function $p_X(x)$ in the sample space Ω is defined asCover and Thomas (2006):

$$H = \int_{\Omega} p_X(x) \log \frac{1}{p_X(x)} dx$$

Correspondingly, Shannon's entropy of network state $\mathbf{x}_k$ with the posterior $p(\mathbf{x}_k|\mathbf{z}_k)$ is defined as:

$$H_k = \int_{\mathbb{R}^n} p(\mathbf{x}_k|\mathbf{z}_k) \log \frac{1}{p(\mathbf{x}_k|\mathbf{z}_k)} d\mathbf{x}_k$$

This entropy can be viewed as the perceptor state.

It is important to note that in general, Shannon's entropy could assume the value zero; however, in cognitive control, the entropic state H_k will always have a non-zero, positive value due to the fact that the environment always involves uncertainty and we can never reach perfect target-state reconstruction with 100% accuracy.

By definition, the function of *cognitive control* is to minimize the entropic state (i.e., state of the perceptor) on a cycle-by-cycle manner Haykin *et al.* (2012a). Cognitive control therefore requires the entropic state, which is computed in the perceptor and then passed to the cognitive controller as *feedback information*. Needless to say, this original definition of cognitive control matches the requirement of stochastic observability, as discussed previously.

The following subsection discusses the cyclic information flow and defines the cognitive controller as an optimal supervisor for the state reconstruction process by the perceptor.

4.5.2 Cyclic Directed Information Flow

The *global perception-action cycle*, depicted in Fig. 4.1, plays a key role in a cognitive dynamic system; it is said to be global, in that it embodies the perceptor in the right-hand side of the figure, the cognitive controller in the left-hand side of the figure, and the monitored network, thereby constituting a closed-loop feedback system that includes the environment (i.e., the network in this context). The entropic state introduced in Subsection 4.5.1 is indeed a measure of the lack of sufficient information for state-reconstruction in the perceptor. Next, the entropic state supplies the *feedback information*, which is sent to the cognitive controller by the perceptor. With this feedback information at hand, the cognitive controller acts on the network, producing

a change in the monitor nodes. Correspondingly, this change affects the amount of relevant information about the network, which is extracted from the new configuration of monitor nodes. A change is thereby produced in the feedback information and with it, a new action is taken on the network by the cognitive controller in the next perception-action cycle. These actions are called "cognitive actions" due to their role in controlling the directed information flow. To summarize, we may therefore define each cognitive action to be the selection of a possible set of monitor nodes. Continuing in this manner from one cycle of perception-action to the next, the cognitive dynamic system experiences a *cyclic directed information flow*, as illustrated in Fig. 4.1.

In addition to feedback information directed from the perceptor to the cognitive controller, there is also a *feedforward information* link from the cognitive controller to the perceptor. In other words, the perceptor and the cognitive controller are reciprocally coupled. This important link provides the means for bypassing the network in order to "predict" the future global cycles for a hypothesized action. This feedforward link is the facilitator of *predictive planning*.

4.5.3 Summary of Cognitive Control

The algorithmic steps involved in cognitive control from each cycle of perception-action to the next are summarized as follows (for further information, the reader is referred to Fatemi and Haykin (2014)):

A) **Initialization**:

i) Action Library: As described in Section 4.3.1, for a network with m accessible nodes and prescribed q monitor nodes at each perception-action cycle, $\binom{m}{q}$

sets of monitor nodes will be available in total, the selection of which are considered to be cognitive actions in the action library of the CDS.

ii) Value-to-go: To each cognitive action (set of monitor nodes), a value-to-go is allocated, which is initialized to zero.

iii) Initial Action: One of the sets in the cognitive action library is then selected randomly at the very first cycle.

B) **Cyclic Process**:

i) Given the observables, reconstruct the network state using Bayesian filtering and compute the state posterior through the well-known time-update and measurement update stages of filtering.

ii) Compute the corresponding entropic state as the feedback information for cognitive control.

iii) Compute the entropic reward and update the value-to-go function.

iv) Compute the predictive planning updates using the internally composite cycle.

v) Repeat step "iv" for all hypothesized cognitive actions and lookahead predictions, as computationally permitted.

vi) Using the resulting policy, select the best set of monitor nodes for the next cycle.

A direct consequence of using cognitive control is not only that it allows for the network structure to be dynamic, but it also results in finding an exact monitor set

in each cycle as opposed to methods such as LSB, which only provide a collection of possible choices but not any exact choice.

Moreover, applying different constraints to monitor nodes are also permitted simply by defining the cognitive actions to be in accordance with the given constraints. This is another desirable feature of deploying cognitive control. The reason for this distinctive capability is that the cognitive controller finds the best cognitive action in the cognitive-action-space regardless of how this action-space has been defined. Therefore, we can define the cognitive-action-space in one form or another that best fits the design specifications of the problem at hand. For example, cognitive actions may be defined as sets of monitor nodes with prescribed cardinality or with inclusion/exclusion of a number of prescribed nodes. In the latter case, we may exclude some of network's nodes from being monitor nodes because they are *inaccessible*. On the other hand, we may force the set of monitor nodes to include some prescribed nodes by simply defining the cognitive actions to be so.

Next section provides computer experiments in order to confirm the advantages of the proposed method and validate the claims made in the previous sections. For the sake of demonstrating the power of cognitive control, we explicitly restrict the cardinality of monitor sets to some prescribed values.

4.6 Computational Experiments

In this section, we provide different examples to demonstrate the methodology just discussed. Our approach follows the one elaborated in Fatemi and Haykin (2014). The first two sets of experiments pertain to the observability of linear networks. The third experiment will then examine the observability of a nonlinear benchmark process.

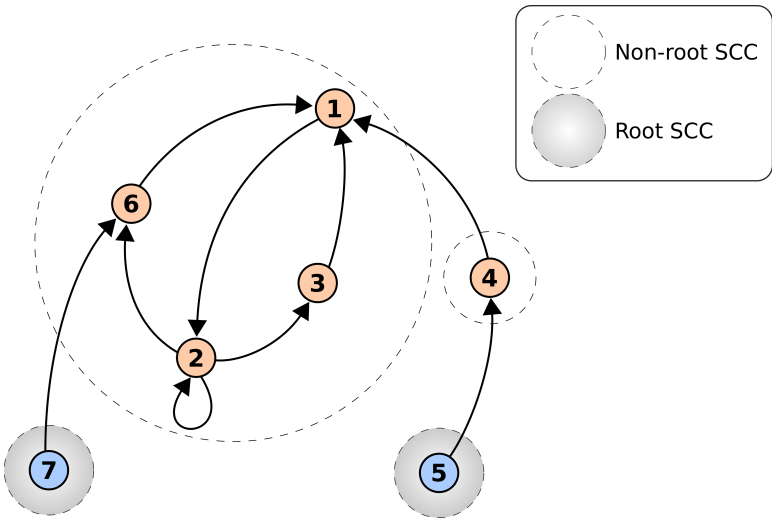


Figure 4.2: Graphical illustration of the network in example 1. The numbered circles depict the seven nodes of the network. Dashed-line circles demonstrate strongly-connected components (SCC), where the shaded ones are the root SCC's that contain no inward edges. The nodes in blue (5 and 7) are the suggested monitor nodes by the LSB method.

Example 1: A Small Linear Network

Consider a network of size n (with n number of nodes) with the adjacency matrix $\mathbf{A}$. Assume that all the nodes are accessible (i.e., $m = n$) but only $q << n$ nodes are permitted to be monitored at each perception-action cycle. The main reason for this setup is that full monitoring of a complex network is not practically/computationally tractable. We demonstrate that a cognitive controller that minimizes the entropic state, is able to successfully select monitor nodes that minimize the state-reconstruction error of the network.

For the sake of demonstration of basic concepts, in this first example, we use a network of size $n = 7$, with only one monitor node (i.e., $q = 1$), and the following

adjacency matrix:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & -0.3 & 0.9 & 0 & 0.4 & 0 \\ 1.2 & 1.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.6 & 0 & 0 & 0 & 0 & 1.7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

As illustrated in Fig. 4.2, the LSB method suggests nodes 5 and 7 as necessary monitors. Uncertainty in both state and monitor equations are modelled by additive zero-mean white Gaussian random processes. Under the Markovian assumption for state evolution, this problem therefore gives rise to the following state-space model:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}^T \mathbf{x}_k + \mathbf{v}_k \mathbf{z}_k = \mathbf{e}_j \mathbf{x}_k + w_k \quad , \end{cases}$$

where, $\mathbf{x}_k \in \mathbb{R}^7$ and $z_k \in \mathbb{R}$ are network's state and observation, respectively. Specifically, $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q})$ and $w_k \sim \mathcal{N}(0, \sigma_w^2)$ are zero-mean, white Gaussian random processes with covariance matrix $\mathbf{Q}$ for $\mathbf{v}_k$ and variance σ_w^2 for w_k. Selection vector $\mathbf{e}_j \in \mathbb{B}^7$, $\mathbb{B} = \{0, 1\}$, is a row-vector with all of its elements equal to *zero* except for the j-th element, which is *one*. Hence, the l-th node of the network will be selected to be the monitor node if and only if $j = l$. The initial state has been set to 1.00 for all the states. Moreover, $\mathbf{Q}$ has been selected as the diagonal matrix with diagonal elements of 10^{-6}, and $\sigma_w^2 = 0.005$. For Gaussian processes similar to those involved in this problem, Shannon's entorpy can be shown to be $H_k = \frac{1}{2}\log(\det\{(2\pi e)\mathbf{P}_{k|k}\})$,

with $\mathbf{P}_{k|k}$ denoting the covariance of state reconstruction error vector at cycle k given observation also at cycle k. We use $H_k = \text{trace}\{\mathbf{P}_{k|k}\}$ as the entropic state, where the *trace* operator is sum of eigenvalues as opposed to *determinant*, which is multiplication of eigenvalues. The reason is that *trace* operator gives rise to the same result as Shanon's entropy due to the fact that eigenvalues of the positive-definite matrix $\mathbf{P}_{k|k}$ are all positive values and logarithm is a monotonic function. However, *trace* results in larger values for the entropic state, which is preferred for the sake of demonstration. The experiment run-time is 10 seconds with 10 observations per second.

The goal here is to find the best state to be monitored in each cycle using cognitive control. Using deterministic observability test, it is easy to show that deterministic version of this network (i.e., without noise terms) is not fully observable having a single monitor node. However, we may be able to minimize the entropic state by monitoring a "proper" node at each cycle with some minimum sample rate. In so doing, we produce a perception-action cycle by adopting a Kalman filter as the perceptor and the cognitive control algorithm as the controller to pick a proper monitor at each cycle. Fig. 4.3 illustrates the problem *without* cognitive controller. As expected, due to lack of observability, no correct estimation exists and both the entropic state and estimation error merely fluctuates with no actual convergence.

In Fig. 4.4, cognitive control has been deployed with 30 actions for planning in each global cycle and two predictive lookahead cycles for each of the hypothesized actions for planning. As illustrated, both the entropic state and the state reconstruction error have been minimized and stay close to zero. An interesting outcome is that the cognitive controller's action almost settles for choosing a specific node after the

fourth second, and then completely stops switching after the seventh second.

Next, we plot the histogram for monitor-node selection over the run time, which is averaged over 50 Monte Carlo realizations. The histogram, illustrated in Fig. 4.5-a, provides clue about which node proving itself to be more important in terms of network observability. Finally, we repeat the same experiment, this time with two monitor nodes, i.e., $q = 2$. The resulting histogram of this experiment is depicted in Fig. 4.5-b, which nicely confirms the previous histogram.

Example 2: The Impact of Network's Topology on Observability of Complex Networks

In the previous example, we discussed how cognitive control in a dynamic fashion picks the best monitor nodes, when the number of monitor nodes is limited. In this example, we expand on the methodology that explained in detail in Example 1, and implement cognitive control for two basic classes of complex networks, namely, Erdős-Rényi (ER) and scale-free random networks. Each class is examined with a number of different case-studies from sparse to dense networks. All the experiments involve uncertainty both in modelling as well as monitoring.

The results, illustrated in Fig. 4.6, suggest that only one monitor node may still be sufficient to rapidly reach bounded and relatively small state-reconstruction error, provided that the monitor node changes dynamically over time (for a single static monitor, the Bayesian filter always crashes due to overflow of error covariance matrix). More importantly, the key result here is that in ER networks, which are uniform in terms of edge distribution, the state-reconstruction error is noticeably less (and more well-behaved) than their scale-free counterparts. This suggests that

becoming more complex in terms of distance from a uniform structure may imply the need for more monitor nodes.

Dense vs. Sparse Networks: A basic question to be addressed is how *edge* density and distribution affects the observability property of a network. Supported by Monte Carlo simulations depicted in Fig. 4.6, our next key result is that as the network becomes more complex, both the entropic state and state-reconstruction error increase even in the presence of the cognitive controller. It implies the need for more monitor nodes as the network becomes more dense in terms of number of edges with fixed number of nodes. This result is clearly counterintuitive considering the hypothesis from LSB that the number of monitor nodes decreases as the network becomes denser and has less SCCs (see Table 4.1). Simply put, although the number of necessary monitor nodes suggested by LSB decreases, the actual number of sufficient monitor nodes will increase.

Example 3: A Benchmark Nonlinear Process

In this last example, we follow a different approach: the role of cognitive control here is to dismiss the most redundant monitor(s) in each global cycle. In other words, we start with complete monitoring of all network nodes and we remove the node that is least informative in long-term (in dynamic programming sense). We then draw the histogram of the selected monitor nodes, which is not only over the entire run-time, but is also averaged over a large number of different realizations of the experiment. This approach provides a Monte Carlo based technique to find the best monitor nodes before the actual sensors are used in a lab setting, and may be very useful for practical sensor selection.

For this example, we look into a benchmark nonlinear process, which is presented in Liu *et al.* (2013), pertaining to a chemical reaction system with 11 species (A, B, C, ..., J, K) as depicted in Fig. 4.7. All the species are involved in the following four reactions:

$$A + B + C \rightarrow D + F + J$$

$$D \leftrightarrow E$$

$$H + I \leftrightarrow G$$

$$J + K \rightarrow G + H$$

Because two of the reactions are reversible, we have six elementary reactions. Balance equations of the chemical reaction system are derived using the mass-action kinetics as the following Liu *et al.* (2013):

$$\dot{x}_1 = -k_1 x_1 x_2 x_3$$

$$\dot{x}_2 = -k_1 x_1 x_2 x_3$$

$$\dot{x}_3 = -k_1 x_1 x_2 x_3$$

$$\dot{x}_4 = +k_1 x_1 x_2 x_3 - k_2 x_4 + k_3 x_5$$

$$\dot{x}_5 = +k_2 x_4 - k_3 x_5$$

$$\dot{x}_6 = +k_1 x_1 x_2 x_3$$

$$\dot{x}_7 = +k_4 x_8 x_9 - k_5 x_7 + k_6 x_{10} x_{11}$$

$$\dot{x}_8 = -k_4 x_8 x_9 + k_5 x_7 + k_6 x_{10} x_{11}$$

$$\dot{x}_9 = -k_4 x_8 x_9 + k_5 x_7$$

$$\dot{x}_{10} = +k_1 x_1 x_2 x_3 - k_6 x_{10} x_{11}$$

$$\dot{x}_{11} = -k_6 x_{10} x_{11}$$

where x_1, x_2, ..., x_{11} denote concentrations of the 11 species, and rate constants of the six elementary reactions are given by k_1, k_2, ..., k_6, respectively. Based on the LSB method, the original deterministic system is suggested to have at least *three* monitor nodes Liu *et al.* (2013): x_6, one from $\{x_4, x_5\}$, and one from $\{x_7, x_8, x_9\}$.

We consider 1% of uncertainty in both state and monitoring equations, presented by two white Gaussian random processes, respectively, which are mutually independent. Because the process equations are continuous-time and the monitoring process occurs in the form of digital sampling, which is discrete-time, we have to employ a *hybrid* Bayesian filter Bar-Shalom *et al.* (2001). Note also that the first derivatives with respect to most variables, the second derivatives with respect to *all* variables, and all the higher order derivatives with respect to cross variables are all zero. Therefore, a hybrid extended Kalman filter (HEKF) will be the best choice Bar-Shalom *et al.* (2001), since it will be very close to optimal. Nevertheless, it is important to say that because HEKF involves the use of a Range-Kutta ODE solver, its implementation involves additional approximations. Therefore, the result will not be on par with the linear discrete-time Kalman filter in the previous two examples. The experiment runs for 20 seconds with four sampling per second.

Using fixed monitors based on the LSB method, our simulations show that in 1000 Monte Carlo realizations of the experiment ALL have been crashed due to overflow of the estimation error covariance matrix. This fact implies that the information obtained from the monitor nodes suggested by the LSB method is considerably below

sufficiency.

Next, we deploy cognitive control to dynamically rank the best candidates to be monitored. To this end, we implement cognitive control in a way that it finds the worst monitor node in each global cycle. The cognitive controller uses 20 hypothesized action for planning in each global cycle and one predictive cycle for each of them. Fig. 4.8 illustrates the resulting histogram. In the case that only one monitor node is considered as redundant, the histogram suggests that x_9 and x_{11} are respectively the worst monitors with highest probability. It also shows that x_6 is the most important one to be monitored, followed by x_7 and x_{10} with highest probability. The result is in partial agreement with the case of deterministic systems. We then repeat the experiment, but this time with dismissing the *two* most redundant monitor nodes. In this case, the histogram suggests that the tuples that contain the nodes x_9, x_{11}, x_5, and x_3 respectively, are the worst monitor sets with highest probability, which is in total agreement with the case of only one redundant monitor node.

4.7 Summary and Discussion

In this paper, we studied the problem of observability in complex stochastic networks. The reported results demonstrated the fact that extending a good deterministic approach such as the LSB algorithm to stochastic networks is not straightforward because it may suggest an improper set of monitor nodes. Hence, we suggested to implement a cognitive dynamic system over the network of interest, for which the environment is the given network. Having the CDS, we will then be able to deploy cognitive control, which provides the best set of monitor nodes in a *dynamic* manner. Regarding the proposed framework, the following points are noteworthy:

- A practical feature of deploying cognitive control is that in addition to optimizing design parameters such as the number of planning actions, ad hoc solutions can also be incorporated within the presented methodology in this paper so as to better match the dynamic monitoring process to the problem at hand. For example, the learning parameters for both learning and planning processes can be adaptively varied in the course of time. More interestingly, switching constraints may also be applied, if need be, to decrease switching between different monitor sets.

- The proposed methodology may also be used offline using the Monte Carlo method. This way, the resulting histogram (similar to those presented in Example 3), can be used to provide realistic information about the best monitor nodes for the network of interest. This information may be very helpful for some real-world problems.

- Our next key result based on the experiments is that as the network becomes more complex in terms of both edge density as well as distribution, the required number of monitor nodes increases, which is intuitively satisfying. This conclusive statement seems counterintuitive regarding the hypothesis favoured by the LSB method that decreasing the number of SCCs results in decreasing the number of necessary monitor nodes.

- Regarding the potentials of CDS, it will be logically sound to claim that CDS may play a key role in the design of next generation of systems in future. The reasons include:

  - Many, if not all, complex networks in real world involve uncertainty both in

the modelling as well as in the monitoring stages. The CDS is by definition the paradigm that deals with uncertainty through perception and control of the directed flow of information in the best manner possible.

- By means of cognitive control, information supervision is an intrinsic part of the CDS paradigm, which guarantees entropy reduction in the perception process in an optimal (or sub-optimal) manner.
- The proposed methodology of this paper will benefit from future advancements of the CDS paradigm. Two possible ways to enrich this methodology are suggested as future research topics. An active perceptual memory can be incorporated to enhance the Bayesian estimation by feeding the filter with modelling parameters. Moreover, inclusion of the so-called *pre-adaptive* mechanism Haykin and Fuster (2014) in the control side may help to cope with disturbances and intermittencies in the monitoring process.

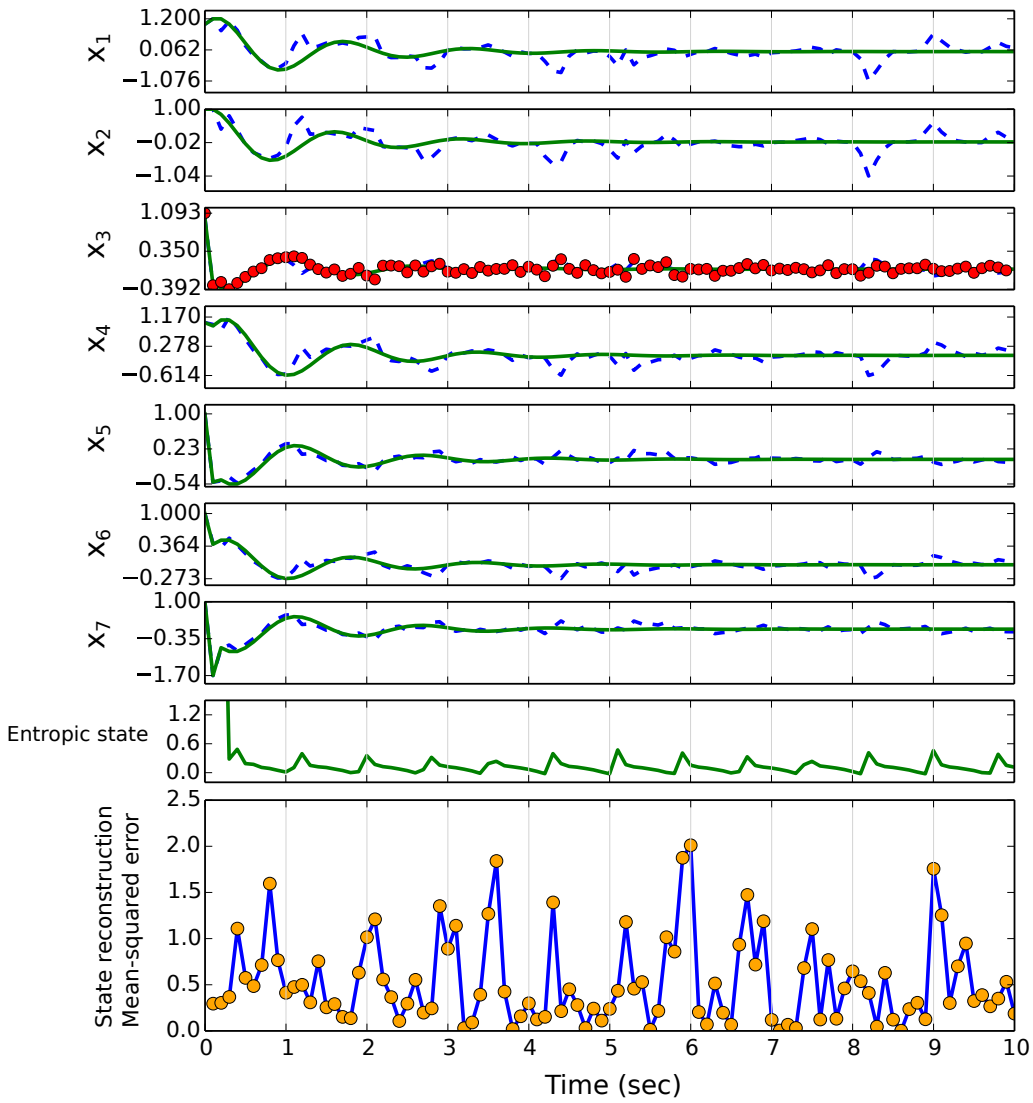


Figure 4.3: Stochastic network observability without cognitive control (example 1): Out of seven states, only one state has been randomly selected to be monitored. Solid lines show the true states, dashed blue lines illustrate the estimated states resulting from the Kalman filter, and the red dots are noisy measurements, all coming from one randomly selected state (x_3 in this illustration). The entropic state only fluctuates over time with no convergence.

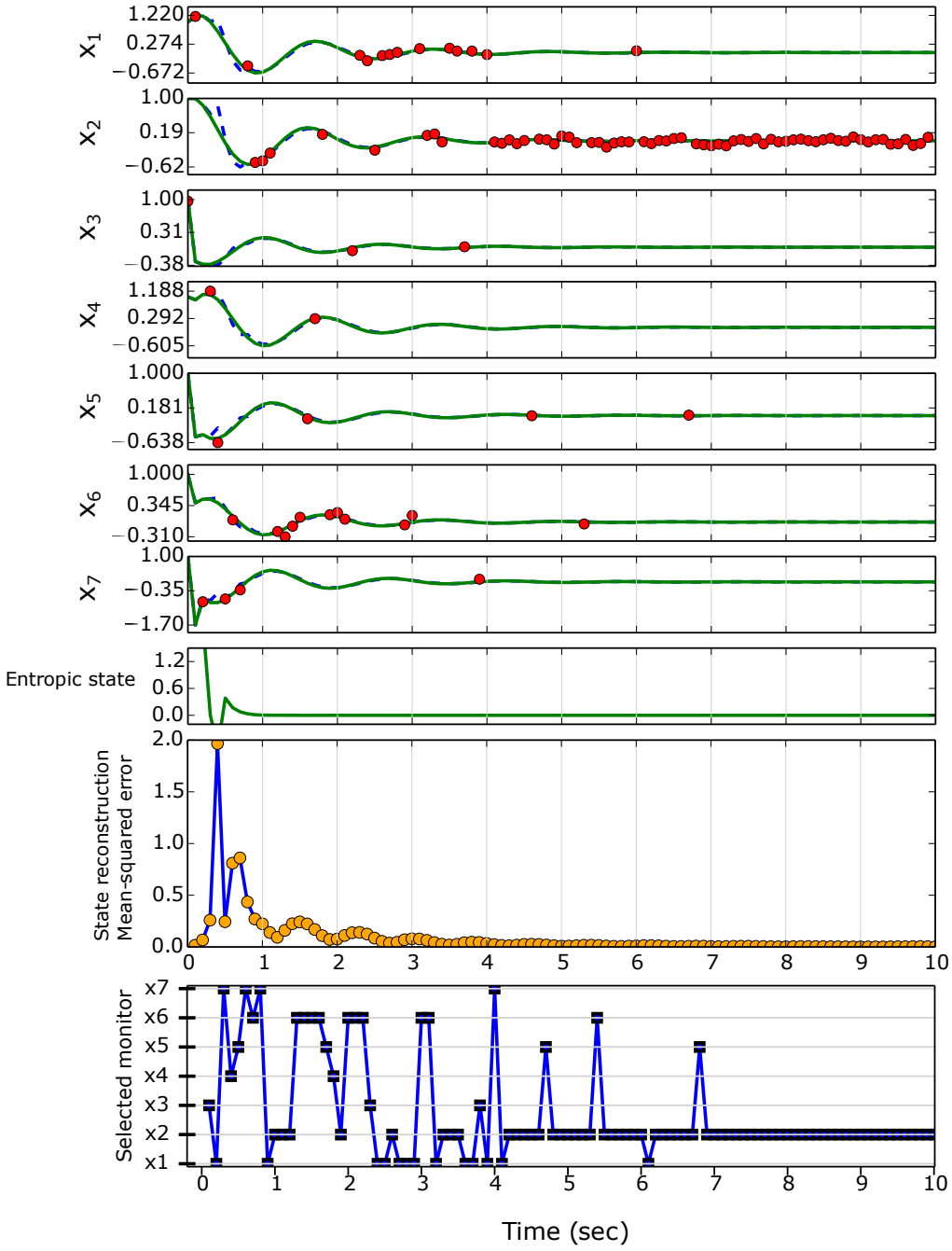


Figure 4.4: Stochastic network observability with cognitive control for the same problem as in Fig. 4.3 (example 1): As it can be seen, the entropic state becomes almost zero even only before the very first second of the experiment. It is noteworthy that the cognitive controller completely converges to a specific monitor node (x_2 in this case), which is different from the nodes suggested by the LSB method (x_5 and x_7).

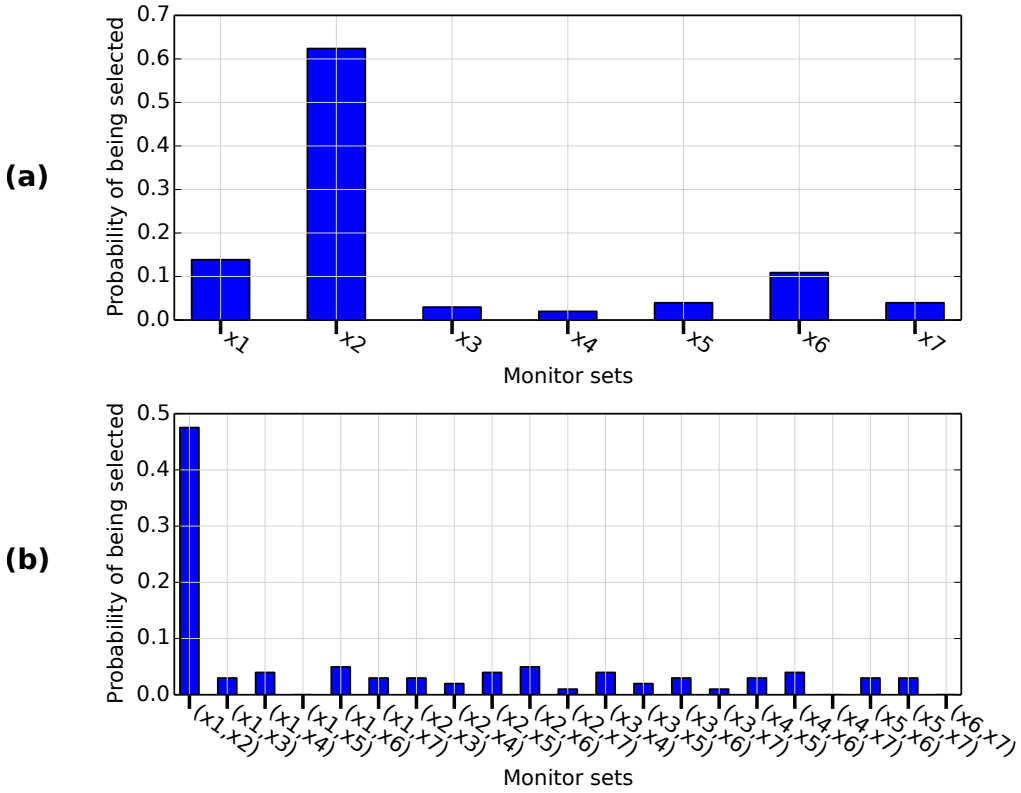


Figure 4.5: Histogram of the selected nodes in example 1, using cognitive control with 30 random actions used for planning, each of which with two cycles of lookahead prediction into the future. Simulation results are averaged over 50 realizations. (a) Only one monitor node: It suggests x_2, x_1 and x_6 as the best monitors with the highest probabilities. (b) The number of monitor nodes is selected to be two: The histogram suggests the tuples (x_1, x_2), (x_1, x_6), and (x_2, x_6) as the best monitor sets with the highest probabilities. This is consistent with the case of only one monitor node.

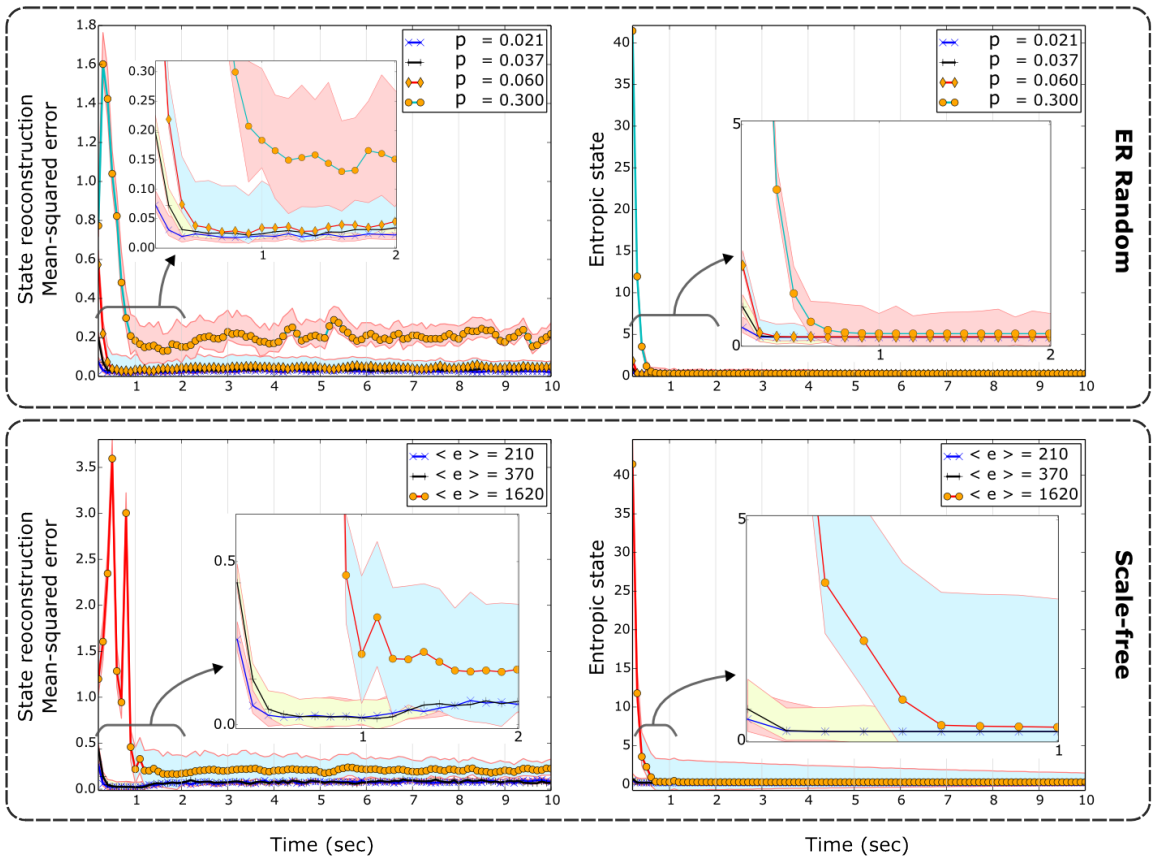


Figure 4.6: Stochastic network observability in various configurations of Erdős-Rényi (ER) and scale-free networks. All the networks have 100 nodes and they are limited to have only one monitor. The parameters p and $< e >$ denote probability for edge creation and average number of edges, respectively. For both types of networks, in addition to the entropic state, the state-reconstruction mean-squared error is also plotted to provide a measure for performance. In each global cycle, we use 150 hypothesized actions for planning and two predictive hypothesized cycles for each planning action. All the simulations are averaged over 50 realizations. For the sake of demonstration, close-ups of the confidence areas are also shown. For comparison, see also Table 4.1.

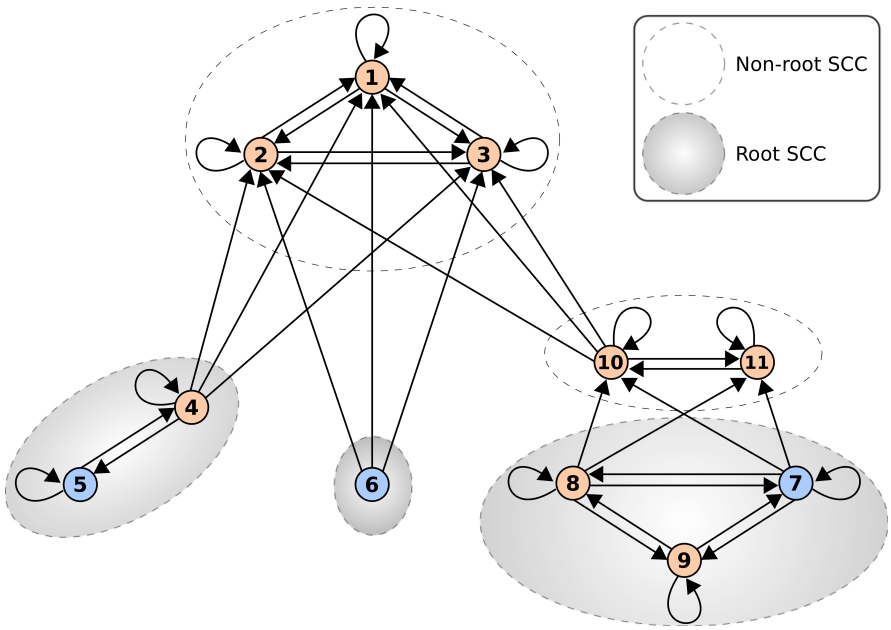


Figure 4.7: Graphical illustration of the network in example 3. The numbered circles depict the nodes of the network. Dashed-line circles demonstrate strongly-connected components (SCC), where the shaded ones are the root SCC's that contain no inward edges. The ones in blue are the suggested monitor nodes by the LSB method.

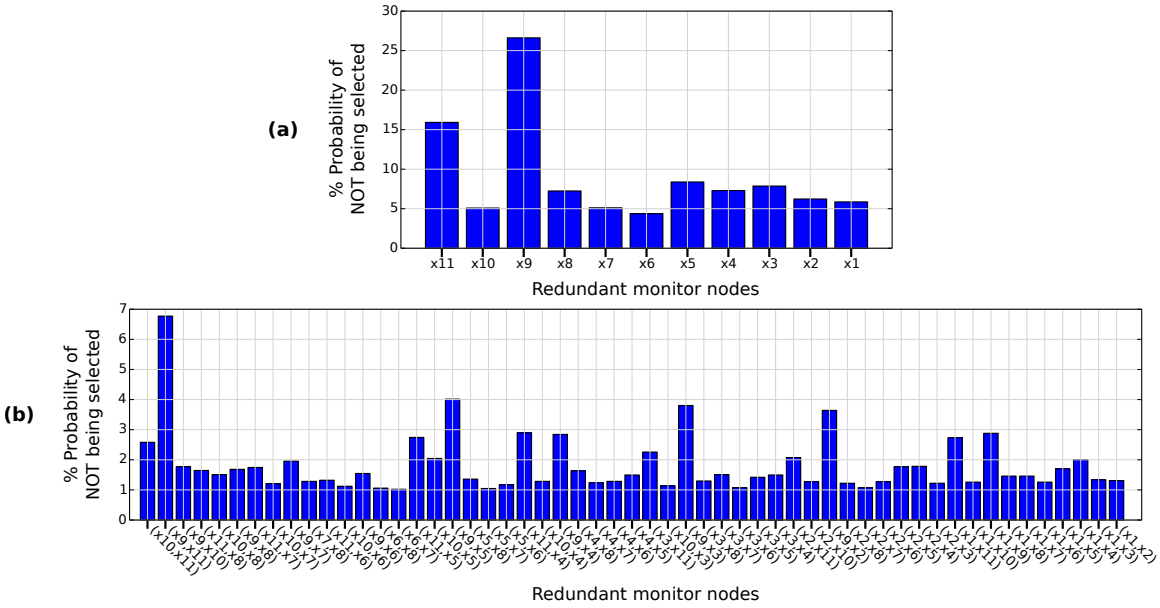


Figure 4.8: Histogram of the redundant nodes in example 3, using cognitive control with 20 random actions used for planning, each of which with one cycle of lookahead prediction into the future. The simulations are averaged over 200 realizations. (a) Only one monitor node is considered as redundant: It suggests that x_9 and x_{11} are the worst monitors with the highest probabilities. It also shows that x_6 is the most important one to be monitored, followed by x_7 and x_{10} with the highest probabilities. The result is in partial agreement with the case of deterministic systems, (b) The number of redundant monitor nodes is selected to be two: The histogram suggests that the tuples that contain the nodes x_9, x_{11}, x_5, and x_3 are the worst monitor sets with the highest probabilities, which is in total agreement with the case of only one monitor node.

Chapter 5

Conclusion

5.1 Research Summary

5.1.1 List of contributions

The contributions of this thesis are listed as follows:

1. Introducing the novel concept of *entropic state* of perceptor, and thereby the notion of a two-state model.

2. Developing the definition of cognitive control for engineering systems for the first time.

3. Mathematically formulating the learning algorithm of cognitive control on the basis of cyclic directed information flow as well as the entropic state. The novelty of this algorithm is also due to the *stateless* nature of it, which intrinsically differs cognitive control from both Bellman's dynamic programming and the traditional reinforcement learning. Most importantly, this stateless nature, as

shown in the thesis, alleviates the so-called curse of dimensionality, which is another key result.

4. Presenting mathematical proof of convergence to optimal policy for the learning algorithm.

5. Applying *planning* as the second important process in cognitive control.

6. Integrating *explore/exploit* tradeoff into cognitive control to improve the efficiency.

7. Developing two different object-oriented software testbeds for cognitive control, both of which are completely reusable.

8. Application of cognitive control in tracking radar, resulting in a new generation of cognitive radars.

9. Application of cognitive control to the information supervisory of *stochastic complex networks*: The networks are complex in terms of edge density, and they are stochastic in that they involve uncertainty in both of modelling and monitoring processes. This last contribution is not only important from the cognitive control point-of-view as a novel application, but it may also be rather influential to the network science literature.

5.1.2 Significance of the Research

In the course of the past four years, a novel paradigm has been introduced to control the directed flow of information in complex dynamic systems and networks. The

paradigm, which is called *cognitive control*, has been established conceptually in Paper I (see Chapter 2). Regarding the significance and novelty of the introduced paradigm, it is also noteworthy to mention that Paper I featured as a cover story for the Proceedings of the IEEE.

Next in Paper II (see Chapter 3), the mathematical framework with the proof of desired characteristics has been presented, which provides the algorithm to implement cognitive control. Furthermore, the theory has been illustrated though computational experiment involving the use of a cognitive tracking radar. For the first time ever, we have been able to achieve error reduction even more than the benchmark "dynamic optimization" method in less time and with considerably less computational load. The significance and novelty of Paper II has been commended by the two reviewers of IEEE Access journal, including the following quote:

> "The paper presents the study that opens the novel scientific direction in the research of cognition activity. The described results can be recognized as theoretical innovations in the cognitive control."

Moreover, the paper has been top-listed at the IEEE Access journal since July 2014.

In our third publication, Paper III herein (see Chapter 4), a novel application for cognitive control has been proposed pertaining to the observability of complex stochastic networks. This new way of thinking overcomes the shortcomings of the state-of-the-art in a practical and flexible manner. Paper III has "submitted" status at the time of writing this dissertation. The significance of Paper III is mostly due to the fact that the observability of complex networks, despite its importance, is rather new, and the involving issues in many aspects of it are still remained unsolved. Even the recently introduced LSB method (see Chapter 4) is rather simplistic. Paper

III paves the way for addressing the problem of observability in stochastic complex networks and is therefore a reasonable start-point for future research.

5.2 Future Research

In this final part of the thesis, we propose three directions for future research based on the achieved contributions. While all the three of them have the importance and impact of their own, the first highlighted future research directly impact the stance of cognitive control theory itself. The second one then involves the impact of cognitive control on risk control in the face of severe disturbances. Finally, the third future research, will more so impact the literature of network science and engineering.

5.2.1 Topic I: Hierarchical Structures

Looking to the cognitive neuroscience for inspiration, a first rationale expansion to the presented theory of cognitive control is to include a hierarchical structure. For example, the hierarchy may well be thought in terms of *labeling* a number of more primary cognitive actions to become one higher-level action at a higher level of hierarchy. As a result, the controller may have the capability of thinking and learning in different levels of abstraction. To elaborate, let us refer back to the human brain again. We, human beings, like other animals do *not* think of our primary actions once we focus on higher-level actions (higher-level in terms of a desired goal of interest). For example, when we intend to move a glass from one location on the table to somewhere else, we do not seemingly think of how to pick up the glass, or which muscles we should use and so forth. They all being done automatically, may be because they

are in a lower level of hierarchy in the control part of our brain, the end result of which is to make us capable of thinking more efficiently without concentrating on small details. Similar process may be adopted in cognitive control if need be, specifically in problems involving huge cognitive action libraries, where many of the actions are indeed too detailed for regular purposes.

5.2.2 Topic II: The Impact of Cognitive Control on Risk Control

This line of research has already been started in the Cognitive Systems Laboratory of McMaster University. The problem involves implementation of the so-called pre-adaptation mechanism Haykin and Fuster (2014) next to the cognitive controller (encompassing the executive memory) in addition to the deployment of perceptual memory. Presented in Haykin *et al.* (2014), our primary results involving severe disturbances have demonstrated considerable improvement of performance over the CDS without those new elements. Additionally, the results highlight the potential of this research for becoming the solution to the problem of risk control in face of severe and unforeseen disturbances. This problem is extremely important in real-world applications and may reasonably lead us to practically prevent catastrophic events such as the 2003 blackout.

5.2.3 Topic III: Information Supervisory of Real-world Complex Networks

As discussed in Section 5.1.2, the problem of information management in stochastic complex networks is still very recent. Moreover, many of the practical issues are still completely unattended due to lack of research. On the other hand, with the ever expansion of *connections* across many diverse entities in the real-life, the currently existing networks are becoming not only larger in size, but also more dense and thereby more complex. Needless to say, the ever presence of uncertainty is also inevitable; hence, we face stochasticity more than ever with networks becoming larger and more complex. Consequently, the research presented in Paper III (Chapter 4), may reasonably be continued in two fronts: a) by applying the introduced methodologies to real-world networks, and b) by incorporating the advancements of the cognitive dynamic systems theory, as just discussed under Topics I and II.

Bibliography

Alexander, W. H. and Brown, J. W. (2011). Medial prefrontal cortex as an action-outcome predictor. *Nature Neuroscience*, **14**(10), 1338–1344.

Arasaratnam, I. and Haykin, S. (2009). Cubature Kalman filters. *IEEE Transactions on Automatic Control*, **54**(6), 1254–1269.

Aström, K. J. and Wittenmark, B. (1995). *Adaptive Control*. Prentice Hall.

Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc.

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, **286**(5439), 509–512.

Barto, A. G. (1995). Adaptive critic and the basal ganglia. In J. C. Houk, J. L. Davis, and D. G. Beiser, editors, *Models of information processing in the basal ganglia*, pages 215–232. MIT Press.

Barto, A. G., Sutton, R. S., and Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man and Cybernetics,*, **13**, 834–846.

Bayer, H. M. and Glimcher, P. W. (2005). Midbrain dopamine neurons encode a quantitative reward prediction error signal. *Neuron*, **47**(1), 129–141.

Bayer, H. M., Lau, B., and Glimcher, P. W. (2007). Statistics of midbrain dopamine neuron spike trains in the awake primate. *Journal of Neurophysiology*, **98**(3), 1428–1439.

Bellman, R. E. (1956). Danamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America*, **40**(10), 767–769.

Bellman, R. E. (1957). *Dynamic programming*. Princeton, NJ: Princeton University Press.

Bellman, R. E. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton University Press.

Bellman, R. E. (1966). Dynamic programming. *Science*, **153**(3731), 34–37.

Berger, J. O. (1985). *Statistical decision theory and Bayesian Analysis*. Springer, 2 edition.

Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control*, volume 1, 2. Athena Scientific, third edition.

Bertsekas, D. P. and Tsitsiklis, J. (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Bertsekas, D. P. and Tsitsiklis, J. N. (2008). *Introduction to Probability*. Athena Scientific, 2nd edition.

Brass, M., Derrfuss, J., Forstmann, B., and von Cramon, D. Y. (2005). The role of the inferior frontal junction area in cognitive control. *Trends in Cognitive Sciences*, **9**(7), 314–316.

Buss, M., Hirche, S., and Samad, T. (2011). Cognitive control. In T. Samad and A. Annaswamy, editors, *The Impact of Control Technology*, pages 167–173. IEEE Control Systems Society.

Choromański, K., Matuszak, M., and MięKisz, J. (2013). Scale-free graph with preferential attachment and evolving internal vertex structure. *Journal of Statistical Physics*, **151**(6), 1175–1183.

Cohen, R. and Havlin, S. (2010). *Complex Networks: Structure, Robustness, and Function*. Cambridge University Press.

Corning, P. (2001). Control information: The missing element in Norbert Wiener's cybernetic paradigm. *Kybernetics*, **30**(9-10), 1272–1288.

Cotsaftis, M. (2009). A passage to complex systems. *in Complex Systems and Self-organization Modelling, C. Bertelle, G. H. E. Duchamp, H. Kadri-Dahmani, Editors, Springer*, pages 3–19.

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory, 2nd Edition*. John Wiley & Sons, Inc.

Crisan, D. and Rozovskii, B., editors (2011). *The Oxford Handbook of Nonlinear Filtering*. Oxford Handbooks in Mathematics. Oxford University Press.

Dayan, P. and Niv, Y. (2008). Reinforcement learning: The good, the bad and the ugly. *Current Opinion in Neurobiology*, **18**(2), 185–196.

Dupuy, J. P. (2009). *On the Origins of Cognitive Science: The Mechanization of the Mind*. MIT Press.

Erdős, P. and Rényi, A. (1960). On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, **5**, 17–61.

Fatemi, M. and Haykin, S. (2013). On reinforcement learning and planning in cognitive control. submitted to IEEE Transactions on Neural Networks and Learning Systems.

Fatemi, M. and Haykin, S. (2014). Cognitive control: Theory and application. *IEEE Access*, **2**, 698–710.

Feldman, H. and Friston, K. J. (2010). Attention, uncertainty, and free-energy. *Frontiers in Human Neuroscience*, **4**(215).

Feldman, J. (2000). Minimization of Boolean complexity in human concept learning. *Nature*, **407**, 630–633.

Fisher, R. A. (1922). On the mathematical foundation of theoretical statistics. *Philosophical Transactions of the Royal Society, London, Section A*, **222**, 309–368.

Fuster, J. M. (2003). *Cortex and Mind: Unifying Cognition*. Oxford University Press.

Fuster, J. M. (2014). The prefrontal cortex makes the brain a preadaptive system. *Proc. IEEE*, **102**(4), 417–426.

Fuster, J. M. and Haykin, S. (2014). Private communication. Private communication.

Gardner, R. W., Holzman, P. S., Klein, G. S., Linton, H. P., and Spence, D. P. (1959). Cognitive control: A study of individual consistencies in cognitive behavior. *Psychological Issues*, **1**(4), 1–186.

Gibson, J. J. (1976). The myths of passive perception. *Philosophy and phenomenological research*, **37**(2), 234–238.

Gibson, J. J. (1986). *The Ecological Approach to Visual Perception*. LEA.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, **140**(2), 107 –113.

Hammond, K. R. and Summers, D. A. (1972). Cognitive control. *Psychological Review*, **79**(1), 58–67.

Haykin, S. (2001). *Adaptive Filter Theory*. Prentice Hall, 4th edition.

Haykin, S. (2005). Cognitive radio: Brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, **23**(2), 201–220.

Haykin, S. (2006a). Cognitive dynamic systems. *Proc. IEEE, Point of View Article*, **94**(11), 1910–1911.

Haykin, S. (2006b). Cognitive radar: A way of the future. *IEEE Signal Processing Magazine*, **23**(1), 30–40.

Haykin, S. (2009). *Neural Networks and Learning Machines*. Prentice-Hall, 3 edition.

Haykin, S. (2012a). *Cognitive Dynamic Systems*. Cambridge University Press.

Haykin, S. (2012b). Cognitive dynamic systems: Radar, control, and radio. *Proc. IEEE, Point of View Article*, **100**(7), 2095–2103.

Haykin, S. (2012c). Cognitive dynamic systems: Radar, control, and radio. *Proc. IEEE, Point of View Article*.

Haykin, S. and Fuster, J. M. (2014). On cognitive dynamic systems: Cognitive neuroscience and engineering learning from each other. *Proc. IEEE*, **102**(4), 608–628.

Haykin, S., Zia, A., Xue, Y., and Arasaratnam, I. (2011). Control theoretic approach to tracking radar: First step towards cognition. *Digital Signal Processing*, **21**, 576–585.

Haykin, S., Fatemi, M., Setoodeh, P., and Xue, Y. (2012a). Cognitive control. *Proceedings of the IEEE*, **100**(12), 3156–3169.

Haykin, S., Xue, Y., and Setoodeh, P. (2012b). Cognitive radar: Step toward bridging gap between neuroscience and engineering. *Proc. IEEE*.

Haykin, S., Xue, Y., and Setoodeh, P. (2012c). Cognitive radar: Step toward bridging the gap between neuroscience and engineering. *Proc. IEEE*, **100**(11), 3102–3130.

Haykin, S., Amiri, A., and Fatemi, M. (2014). Cognitive control in cognitive dynamic systems: A new way of thinking inspired by the brain. In *ADPRL14*. IEEE.

Hero, A. O. and Cochran, D. (2011). Sensor management: Past, present, and future. *IEEE Sensors Journal*, **11**(12), 3064–3075.

Ho, Y. C. and Lee, R. C. K. (1964). A Bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, **AC-9**, 333–339.

Howard, R. A. (1966). Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, **SSC-2**(1), 22–26.

Hrycej, T. (1997). *Neurocontrol: Towards an Industrial Control Methodology*. Adaptive and Learning Systems for Signal Processing, Communications and Control Series. Wiley-Interscience.

Ioannou, P. and Sun, J. (1995). *Robust Adaptive Control*. Prentice Hall.

Jackson, M. O. (2008). *Social and Economic Networks*. Princeton University Press.

Julier, S., Uhlmann, J., and Durrant-Whyte, H. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *Automatic Control, IEEE Transactions on*, **45**(3), 477 –482.

Kailath, T. (1980). *Linear Systems*. Prentice-Hall.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, **82**(Series D), 35–45.

Kam, M., Cheng, R., and Kalata, P. (1987). An information-theoretic interpretation of stability and observability. In *Proc. American Control Conference*, pages 1957–1962.

Kershaw, D. J. and Evans, R. J. (1994). Optimal waveform selection for tracking systems. *IEEE Transactions on Information Theory*, **40**(5), 1536–1550.

Klopf, A. H. (1972). Brain function and adaptive systems: A heterostatic theory. Technical Report 133, Air Force Cambridge Research Laboratories, Bedford, MA.

Kouneiher, F., Charron, S., and Koechlin, E. (2009). Motivation and cognitive control in the human prefrontal cortex. *Nature Neuroscience*, **12**(7), 939–945.

Landau, I. D., Lozano, R., M'Saad, M., and Karimi, A. (2011). *Adaptive Control: Algorithms, Analysis and Applications*. Communications and Control Engineering. Springer, 2nd edition.

Lehmann, E. L. and Casella, G. (1998). *Theory of Point Estimation*. Springer, 2 edition.

Lewis, F. W., Jagannathan, S., and Yesildirak, A. (1998). *Neural Network Control of Robot Manipulators and Non-Linear Systems*. Systems and Control. CRC Press.

Li, M. and Vitanyi, P. M. B. (2008). *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 3 edition.

Liu, A. R. and Bitmead, R. R. (2011). Stochastic observability in network state estimation and control. *Automatica*, **47**(1), 65–78.

Liu, Y.-Y., Slotine, J.-J., and Barabasi, A.-L. (2011a). Controllability of complex networks. *Nature*, **473**, 167–173.

Liu, Y.-Y., Slotine, J.-J., and Barabasi, A.-L. (2011b). Liu et al. reply. *Nature*, **478**, E4–E5.

Liu, Y. Y., Barabasi, A.-L., and Slotine, J. J. (2013). Observability of complex systems. *Proceedings of the National Academy of Sciences*, **110**, 1–6.

Mars, R. B., Sallet, J., Rushworth, M. F. S., and Yeung, N. (2012). *Neural Basis of Motivational and Cognitive Control*. MIT Press.

Miller, E. K. and Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annu. Rev. Neurosci.*, **24**(1), 167–202.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network motifs: Simple building blocks of complex networks. *Science*, **298**(5594), 824–827.

Montague, P. R., Dayan, P., Nowlan, S. J., Pouget, A., and Sejnowski, T. J. (1993). Using aperiodic reinforcement for directed self-organization. In C. L. Giles, S. J. Hanson, and J. D. Cowan, editors, *Advances in neural information processing systems*, volume 5, pages 969–976. San Mateo, CA: Morgan Kaufmann.

Montague, P. R., Dayan, P., Person, C., and Sejnowski, T. J. (1995). Bee foraging in uncertain environments using predictive Hebbian learning. *Nature*, **377**, 725–728.

Montague, P. R., Dayan, P., and Sejnowski, T. J. (1996). A framework for mesencephalic dopamine systems based on predictive Hebbian learning. *Journal of Neuroscience*, **16**(5), 1936–1947.

Morris, G., Nevet, A., Arkadir, D., Vaadia, E., and Bergman, H. (2006). Midbrain dopamine neurons encode decisions for future action. *Nature Neuroscience*, **9**(8), 1057–1063.

Mountcastle, V. B. (1998). *Perceptual Neuroscience: The Cerebral Cortex*. Harvard University Press.

Murphy, K. P. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. Dissertation, University of California, Berkeley, USA.

Muske, K. R. and Edgar, T. F. (1997). Nonlinear State Estimation. *in Nonlinear Process Control, M. A. Henson and D. A. Seborg, Editors, Springer*, pages 311–370.

Nepusz, T. and Vicsek, T. (2012). Controlling edge dynamics in complex networks. *Nature Physics*, **8**, 568–573.

Niv, Y. (2009). Reinforcement learning in the brain. *Journal of Mathematical Psychology*, **53**(3), 139 – 154. special issue: Dynamic Decision Making.

Niv, Y., Daw, N. D., and Dayan, P. (2005). How fast to work: Response vigor, motivation and tonic dopamine. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in neural information processing systems*, volume 18, pages 1019–1026. MIT Press.

Noë, A. (2004). *Action in Perception*. MIT Press.

Nørgaard, M., Ravn, O., Poulsen, N. K., and Hansen, L. K. (2000). *Neural Networks for Modelling and Control of Dynamic Systems: A Practitioner's Handbook*. Advanced Textbooks in Control and Signal Processing. Springer.

Powell, W. B. (2011). *Approximate Dynamic Programming: Solving the curses of dimensionality*. Wiley, New York, 2nd edition.

Puskorius, G. V. and Feldkamp, L. A. (2001). Parameter-based Kalman filter training: Theory and implementation. In S. Haykin, editor, *Kalman Filtering and Neural Networks*, pages 23–67. John Wiley and Sons Inc.

Rao, R. and Ballard, D. (1997). Dynamic model of visual recognition predicts neural response properties in the visual cortex. *Neural Computation*, pages 721–763.

Rao, R. and Ballard, D. (1999). Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, pages 79–87.

Ristic, B., Arulampalam, S., and Gordon, N. (2004). *Beyond the Kalman filter: particle filters for tracking applications*. Artech House.

Robert, C. and Casella, G. (2005). *Monte Carlo Statistical Methods , 2nd Edition*. Springer.

Roesch, M. R., Calu, D. J., and Schoenbaum, G. (2007). Dopamine neurons encode the better option in rats deciding between differently delayed or sized rewards. *Nature Neuroscience*, **10**(12), 1615–1624.

Rudin, W. (1976). *Principles of Mathematical Analysis (International Series in Pure and Applied Mathematics)*. McGraw-Hill, 3rd edition.

Rummery, G. A. and Niranjan, M. (1994). On-line Q-learning using connectionist systems. Technical report, Cambridge Univeristy Engineering Department.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal on Research and Development*, **3**(3).

Sastry, S. and Bodson, M. (1989). *Adaptive Control: Stability, Convergence and Robustness*. Prentice Hall. reprinted by Dover Publications, 2011.

Schultz, W. (1998). Predictive reward signal of dopamine neurons. *Journal of Neurophysiology*, **80**(1), 1–27.

Seising, R. (2008). *The Fuzzification of Systems*. Springer.

Setoodeh, P. and Haykin, S. (2009). Robust transmit power control for cognitive radio. *Proc. IEEE*, **97**(5), 915–939.

Shamir, O., Sabato, S., and Tishby, N. (2010). Learning and generalization with the information bottleneck. *Theoretical Computer Science*, **411**(29-30), 2696–2711.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, **27**, 379–423, 623–656.

Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press. Reprinted, 1998.

Shiryayev, A. N. (1992a). *Selected Works of A.N. Kolmogorov: Volume II: Probability Theory and Mathematical Statistics*. Springer.

Shiryayev, A. N. (1992b). *Selected Works of A.N. Kolmogorov: Volume III: Information Theory and the Theory of Algorithms*. Springer.

Shreve, S. E. (2004). *Stochastic Calculus for Finance II: Continuous-Time Models*. Springer.

Sigman, M. (2004). Bridging psychology and mathematics: Can the brain understand the brain? *PLoS Biology*, **2**(9), 1265–1266.

Simon, H. A. (1977). *Models of Discovery: and Other Topics in the Methods of Science*. Boston Studies in the Philosophy and History of Science. D. Reidel.

Soatto, S. (2009). Actionable information in vision. Technical Report CSD090007, UCLA.

Surmeier, D. J., Plotkin, J., and Shen, W. (2009). Dopamine and synaptic plasticity in dorsal striatal circuits controlling action selection. *Current Opinion in Neurobiology*, **19**, 621–628.

Sutton, R. S. (1978). A unified theory of expectation in classical and instrumental conditioning. Unpublished bachelors thesis.

Sutton, R. S. (1984). *Temporal credit assignment in reinforcement learning*. Ph.D. thesis, University of Massachusetts at Amherst, Amherst, MA, USA.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, **3**(1), 9–44.

Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems 8*, pages 1038–1044. MIT Press.

Sutton, R. S. and Barto, A. G. (1990). Time-derivative models of Pavlovian reinforcement. In M. Gabriel and J. Moore, editors, *Learning and computational neuroscience: Foundations of adaptive networks*, pages 497–537. MIT Press.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning*. MIT Press.

Tishby, N. (1999). The information bottleneck method. In *The 37th Allerton Conference on Communications, Control, and Computing*, pages 368–377.

Watkins, C. J. C. H. (1989). *Learning form Delayed Rewards*. Ph.D. thesis, Cambridge University, Cambridge, UK.

Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, **8**, 279–292.

Widrow, B. and Lehr, M. A. (1990). 30 years of adaptive neural networks: Perceptron, madaline, and backpropagation. *Proceedings of IEEE*, **78**(9), 1415–1442.

Wiener, N. (1950). *The Human Use of Human Beings: Cybernetics and Society*. Houghton Mifflin.

Wiener, N. (1964). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. The MIT Press.

Wiener, N. (1965). *Cybernetics: or the Control and Communication in the Animal and the Machine*. The MIT Press, 2nd edition.

Yerkes, R. M. and Morgulis, S. (1909). The method of Pavlov in animal psychology. *Psychological Bulletin*, **6**, 257–273.