# INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

**The quality of this reproduction is dependent upon the quality of the copy submitted.** Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

# Stochastic Decentralized Systems

by

Timothy David Barfoot

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Institute for Aerospace Studies
University of Toronto

# Abstract


**STOCHASTIC DECENTRALIZED SYSTEMS**

Timothy David Barfoot
<tim.barfoot@utoronto.ca>

Doctor of Philosophy
Graduate Department of Institute for Aerospace Studies
University of Toronto
2002

Fundamental aspects of decentralized systems are considered from a control perspective. The stochastic framework afforded by Markov systems is presented as a formal setting in which to study decentralized systems. A stochastic algebra is introduced which allows Markov systems to be considered in matrix format but also strikes an important connection to the classic linear system originally studied by Kalman [1960]. The process of decentralization is shown to impose constraints on observability and controllability of a system. However, it is argued that communicating decentralized controllers can implement any control law possible with a centralized controller. Communication is shown to serve a dual role, both enabling sensor data to be shared and actions to be coordinated. The viabilities of these two types of communication are tested on a real network of mobile robots where they are found to be successful at a variety of tasks. Action coordination is reframed as a decentralized decision making process whereupon stochastic cellular automata (SCA) are introduced as a model. Through studies of SCA it is found that coordination in a group of arbitrarily and sparsely connected agents is possible using simple rules. The resulting stochastic mechanism may be immediately used as a practical decentralized decision making tool (it is tested on a group of mobile robots) but it furthermore provides insight into the general features of *self-organizing* systems.

# Acknowledgements

First and foremost I must thank Professor Gabriele D'Eleuterio who always encouraged me to think in the most general terms. His free-thinking style of supervision allowed me to study topics normally not accessible to an engineering student. Although I had been thinking about stochastic matrices for Markov control for awhile he immediately saw the possibility for an algebra whence Chapter 2 and the nice connection to classic linear systems. His constant optimism and enthusiasm for this work kept me motivated during the long PhD process. Thanks for everything, Gabe.

I would also like to greatly thank Ernest Earon whose partnership in the construction of the RISE system made the addition of experimental results to this thesis possible. His work on the rovers was highly professional and very much appreciated; Chapter 4 could not exist without him. Lawrence Lok helped in the initial stages of the rovers and I would like to thank him for continuing to support this project. Thanks also to T Cherpillod for design and construction of the rover electronics. The electro-mechanical parts of the rovers were diligently duplicated by two outstanding summer students, K Niewiadomska and S Chan. Thank you all!

Thank you very much to NSERC (Natural Sciences and Engineering Research Council) and CSA (Canadian Space Agency) for personal financial support throughout my time as a grad student. Thanks also to CRESTech (Center for Research in Earth and Space Technology), CSA, MDR (Macdonald-Dettwiler Robotics), and Dynacon Ltd. for financial and in kind support of the RISE project.

Thanks to my thesis committee for reading the thesis and providing helpful comments. Finally, thanks in advance to anybody else who ends up reading it. Do not hesitate to contact me if you wish to discuss it or find it useful in your work.

# CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF SYMBOLS

| Symbol | Description | Section | Page |
|---|---|---|---|
| ${}^m\mathbb{R}^n$ | The set of real matrices with $m$ rows and $n$ columns. | 2.1 | 9 |
| $\mathsf{A}, \mathsf{B}, \mathsf{x}, \mathsf{y}$ | Examples of the typeface used for real matrices, columns. | | |
| ${}^m\mathbb{S}^n$ | The set of stochastic matrices with $m$ rows and $n$ columns. | 2.1 | 9 |
| $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{x}, \boldsymbol{y}$ | Examples of the typeface used for stochastic matrices, columns. | | |
| ${}^m\mathbb{D}^n$ | The set of deterministic matrices with $m$ rows and $n$ columns. | 2.1 | 10 |
| $\mathbf{A}, \mathbf{B}, \mathbf{x}, \mathbf{y}$ | Examples of the typeface used for deterministic matrices, columns. | | |
| ${}^m\mathbb{F}^n$ | The set of stochastic functions that map ${}^n\mathbb{S}$ to ${}^m\mathbb{S}$. | 2.12 | 29 |
| $f, g$ | Examples of the typeface used for stochastic functions. | | |
| $\mathcal{A}, \mathcal{B}$ | Examples of the typeface used for subspaces. | 2.7 | 20 |
| $0$ | The usual zero matrix (all entries are zero). | | |
| $\boldsymbol{\Omega}, \boldsymbol{\omega}$ | The stochastic zero matrix (a.k.a., the uniform matrix), the stochastic zero column. | 2.1 | 10 |
| $\mathbf{1}$ | The usual identity matrix. | 2.1 | 10 |
| $\boldsymbol{\Xi}$ | The stochastic identity matrix (a.k.a., the exponential identity matrix). | 2.4 | 15 |
| $\downarrow(\cdot)$ | The stochastic normalization operator. | 2.1 | 10 |
| $^\intercal$ | The stochastic unbiased inverse operator. | 2.1 | 11 |
| $^\iota$ | The stochastic biased inverse operator. | 2.15 | 34 |

| Symbol | Description | Section | Page |
|---|---|---|---|
| $\cdot$ | The stochastic scalar multiplication operator. | 2.1 | 11 |
| $\bigoplus, \oplus$ | The stochastic addition operator. | 2.1 | 11 |
| $\ominus$ | The stochastic subtraction/negation operator. | 2.2 | 12 |
| $\otimes$ | The stochastic vector product operator. | 2.5 | 16 |
| $\langle\ ,\ \rangle$ | The stochastic inner product operator. | 2.3 | 13 |
| $\rangle\langle$ | The stochastic outer product operator. | 2.6 | 19 |
| $\|\|\ \|\|$ | The norm of a stochastic vector. | 2.18 | 37 |

The progress of science requires the growth of understanding in both directions, downward from the whole to the parts and upward from the parts to the whole.
—Freeman Dyson
THE SCIENTIST AS REBEL, 1995

# Chapter 1

# INTRODUCTION

STOCHASTIC DECENTRALIZED SYSTEMS are important to anybody who wonders how many simple agents can interact to produce a coherent group behaviour. Perhaps the most telling examples from nature are social insects (e.g., ants, termites, wasps). These deceivingly simple creatures live in colonies whose membership can be in the millions and larger. Darwin [1839] was impressed with the capabilities of ants:

> A person, on first entering a tropical forest, is astonished at the labours of the ants: well-beaten paths branch off in every direction, on which an army of never-failing foragers may be seen going forth, and others returning, burdened with pieces of green leaf, often larger than their own bodies.

Ants harvest vegetation to fertilize underground fungus gardens, enslave other insect species as cattle, and build intricate systems of roads and bridges. They are able to navigate using such landmarks as trees and stars but their complex chemical communications are perhaps their most inspiring feature. Through chemicals called *pheromones*, ants are able to communicate a great deal of information to their nest mates, thus enabling the entire colony to function as a *superorganism* [Hoyt, 1996]. E. O. Wilson[1], the first myrmecologist to show the important role of pheromones in ant societies, remarks

> There are ... many ways in which ants and human beings are alike. Both are resounding success stories of evolution ... and both have achieved their success through their ability to form social groups, to communicate, and to manipulate their environment with great dexterity.

From the point of view of designing an artificial system (e.g., a robotics system), it would be most desirable to be able to mimic the abilities of successful sociobiological systems. To do so we must

[1] From a 1990 interview [Hoyt, 1996].

understand what makes decentralized systems work. What are the general principles, if any? Can we reduce the essence of how a colony of ants is able to cooperate down to a few simple concepts? Do these same concepts apply to all such systems? Must we construct such a system to prove that it will work? It is hoped that answering these questions will not only provide building blocks with which engineers may construct artificial systems but will at once help scientists to better understand the undeniable force of biological life that permeates our world.

The age-old issue of scientific reductionism is paramount to any discussion of decentralized systems. If there is one clear thing when it comes to *reductionism* it is that the term means many things to many people. Here we mean replacing the study of a more complex system by a simpler, more fundamental one. It is hoped the simpler system will convey the essence of the more complex one. For example, aspects of chaos theory were found in highly complicated systems as well as the simple logistic equation. Period-doubling cascades and intermittency are two well known routes to chaos found in all types of chaotic system, from the very simple to the very complex, from the mathematical to the real [Hao, 1990]. Perhaps the same general similarities are to be found in decentralized systems.

Physicists such as Newton and Einstein sought to describe the universe in terms of a few simple rules with unparalleled success. But Prigogine and Stengers [1984] remind us that

> Classical dynamics, the science of eternal, reversible trajectories, was alien to the problems facing the nineteenth century, which was dominated by the concept of evolution ... Boltzmann's interpretation [of evolution] implies the forgetting of initial conditions, the "destruction" of initial structures, while Darwinian evolution is associated with self-organization, ever-increasing complexity.

Schrödinger [1956] believed that biological life could not be completely accounted for by the current physical laws.

> While living matter does not elude the known laws of physics, the hereditary mechanism does seem to obey certain laws of physics as yet unknown. The laws of physics as we know them are statistical laws based on the tendency of things to go into disorder. There are known qualifications to this – quantum theory being one (particularly at low temperatures). Life appears to be another. It seems to be an orderly behaviour of matter not based exclusively on the tendency towards disorder.

In 1900, Hilbert attempted to reduce all of mathematics to formal statements using a finite number of axioms and an alphabet of symbols. He was unsuccessful. As Dyson [1995], whose balanced attitude towards reductionism is reflected in the opening quote of this chapter, puts it,

> Gödel took Hilbert's formalized axioms of mathematics as his building blocks and built out of them a lofty structure of ideas into which he could finally insert his undecidable arithmetical statement as the keystone of the arch. The proof is a great work of art. It is a construction, not a reduction. It destroyed Hilbert's dream of reducing all mathematics to a few equations, and replaced it with a greater dream of mathematics as an endlessly growing realm of ideas. Gödel proved that in mathematics the whole is always greater than the sum of the parts.

However, to deny categorically the merits of reductionism could be naive. This would mean that there are no fundamental similarities between different decentralized systems. The study of each system would be completely unique. Certainly each system has some unique details, but are there also certain aspects of such systems that are common and which may be exploited in the design of a decentralized system?

Artificial intelligence (AI), and the more general field of artificial life, is concerned with producing artificial systems which, on some level, are able to mimic "biological intelligence". Just what is meant

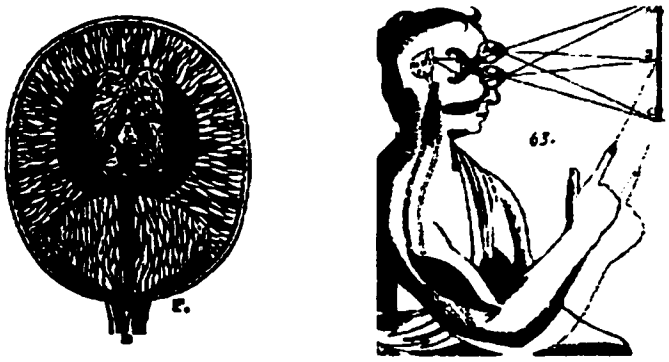


Figure 1.1: [Descartes, 1644]'s schematic drawings of the brain. For him, the teardrop shaped *pineal gland* or *epiphysis* was the locus of interaction between the mind and body.

by biological intelligence is vague at best. In fact, there are two main schools of thought when it comes to artificial intelligence. The major issue separating the two is that of *centralization*. One relies on it while the other denies it. The source of this conflict may be traced to a much older philosophical debate between *dualism* and *materialism*. The most famous dualist was Descartes who imagined a *res cogitans*[2] as a system whose bodily organs sensed the world and informed the mind (a wholly separate entity to the brain) which thought things over and directed the body to perform appropriate actions. For him the locus of interaction between the mind and body was the *pineal gland*, often referred to as the *Cartesian theater*. Descartes [1637] [3] wrote

> ...I recognized that I was a substance whose whole essence or nature is to think and whose being requires no place and depends on no material thing.

Materialists take the sharp opposing view that the mind *is* the brain, that all mental phenomena can be explained by quantifiable, physical phenomena. Schrödinger [1956] advocates a materialist approach in understanding life.

> How can the events that take place within a living organism be accounted for by physics and chemistry? Although present-day physics and chemistry is unable to account for such events, there is no reason for doubting that they can be accounted for by those sciences. Essentially, the problem for physicists is that the structures they deal with are much simpler than those dealt with by biologists. The laws discovered for such simple systems do not apply easily to the more complex. I propose, however, to initially approach the subject of biology from the standpoint of a naive physicist attempting to apply what he knows to an unfamiliar field.

Modern neuroscience has as yet failed to locate the Cartesian theater and in philosophical circles, dualism has fallen into disrepute leaving materialism as the prevailing wisdom of the day. However, the proponents of dualism are still lurking about, particularly in the field of traditional artificial intelligence[4], founded on the *system symbol hypothesis* [Simon, 1969] which states that intelligence operates on a system of symbols. As Brooks [1990] explains (and asks),

> The implicit idea is that perception and motor interfaces are sets of symbols on which the central intelligence system operates. Thus, the central system, or reasoning engine, operates in a domain

[2] Latin, a "thinking thing".
[3] Written in *Discourse on Method*, taken from [Dennett, 1991].
[4] Also called strong AI, classical AI, old AI.

> independent way on the symbols ... [It] must be fed symbols by the perception system. But what is the correct symbolic description of the world around the intelligence system? Surely that description must be task dependent.

This centralized symbolic approach has caused many AI researchers to focus huge amounts of attention on specific subsystems of the perceive-think-act process (e.g., traditional vision). Dennett [1991], an outspoken materialist, points out that

> ... the exclusive attention to specific subsystems of the mind/brain often causes a sort of theoretical myopia that prevents theorists from seeing that their models still presuppose that somewhere, conveniently hidden in the the obscure "center" of the mind/brain, there is a Cartesian Theater, a place where "it all comes together" and consciousness happens. This may seem like a good idea, an inevitable idea, but until we see, in some detail, why it is not, the Cartesian Theater will continue to attract crowds of theorists transfixed by an illusion.

Does Gödel's destruction of Hilbert's dream foreshadow the eventual downfall of traditional AI? There is another AI movement referred to as new AI[5], which is more akin to the view of the materialists than the dualists. It is founded on the *physical grounding hypothesis*. Brooks [1990] contrasts this hypothesis to its traditional counterpart

> ... the *system symbol hypothesis* upon which *classical AI* is based is fundamentally flawed, and as such imposes severe limitations on the fitness of its progeny. ... the *physical grounding hypothesis* provides a different methodology for building intelligent systems than that pursued for the last thirty years. The traditional methodology bases its decomposition of intelligence into functional information processing modules whose combinations provide overall system behaviour. The new methodology bases its decomposition of intelligence into individual behaviour generating modules, whose existence and co-operation let more complex behaviours emerge.

Robust behaviour must be built up from interactions between independent behaviour generating modules rather than specified by a top-down procedure from a central decision-maker. However, this is not to say that there are not fundamental types of interactions which may be exploited in this bottom up approach. In new AI, intelligence is truly in the eye of the observer. There is no Cartesian Theater in which life struts and frets its hour upon the stage.

This thesis is mostly concerned with *decentralized control systems* but it is hoped that something may be said about *decentralized systems* in general. Decentralization has at its heart many interacting units which must be organized into a coherent whole. Physicists today are beginning to find that instability and irreversibility play a large role in so-called *self-organizing* systems [Nicolis and Prigogine, 1977]. Maxwell [1882][6] wrote

> If, therefore, those cultivators of physical science... are led in the pursuit of the arcana of science to study the singularities and instabilities, rather than the continuities and stabilities of things, the promotion of natural knowledge may tend to remove that prejudice in favor of determinism which seems to arise from assuming that the physical science of the future is a mere magnified image of the past.

Laplace imagined a demon which had perfect knowledge of position and velocity of all matter in the universe, capable of inferring its evolution both into the past and the future. Prigogine and Stengers [1984] retort that

---

[5] Also called nouvelle AI, fundamentalist AI, behaviour-based AI.

[6] From a paper of Maxwell published posthumously in L. Campbell and W. Garnett, The Life of James Clerk Maxwell, London, 1882.

> Nature speaks with a thousand voices, and we have only begun to listen. Nevertheless, for nearly two centuries Laplace's demon has plagued our imagination, bringing a nightmare in which ... the world is nothing but an immense tautology. This is the challenge of the science we have inherited from our predecessors, the spell we have to exorcise today.

Even Einstein[7], who was quite against the introduction of probability into quantum theory, noted

> [A law] is more impressive the greater the simplicity of its premises, the more different are the kinds of things it relates, and the more extended its range of applicability. Therefore, the deep impression which classical thermodynamics made on me. It is the only physical theory of universal content which I am convinced, that within the framework of applicability of its basic concepts will never be overthrown.

Motivated by this notion of irreversibility, this thesis seeks to consider decentralized systems in the stochastic framework of Markov decision problems. While the mathematics of traditional AI was Boolean logic, the mathematics of new AI is probability theory and statistics. As a result, a *stochastic algebra* has been constructed, a novel extension of *stochastic matrices*[8]. Here probability distributions may be considered as vectors on which the usual operations of linear algebra may be applied. The zero vector is the uniform probability distribution. The property of linearity is related to statistical independence. In this framework, stochastic dynamic equations may display both types of evolution described above by Prigogine and Stengers [1984]: the destruction of initial structures as the system tends towards a uniform probability distribution (stable equilibrium), and the creation of new ones through appropriate control sequences (stochastic instabilities forcing the system away from equilibrium). The stochastic algebra occupies Chapter 2 and will be used to frame the theoretical work on decentralized control and decision making. The reader may not wish to read this chapter in sequence but rather refer back to it as needed.

There is much overlap between control and AI. Many of the arguments against centralized AI apply equally to centralized control. The view of scientific reductionism in this thesis follows Dyson [1995] in that there is much to learn by studying things in both directions, "downward from the whole to the parts and upward from the parts to the whole". The bulk of work in control engineering implicitly assumes a *central locus of control* to which all sensor inputs arrive and from which all actuator outputs depart. Chapter 3 examines the need for this assumption by asking the question

*When can decentralized controllers perform identically to a centralized controller?*

This will be examined in the context of two systems, *Markov systems* and *linear systems*, both of which are cast in stochastic algebra. The results from the former are new. Furthermore, under certain circumstances, it will be shown that the Markov decision problem model may be transformed into a linear system thus creating a new link to the existing linear decentralized control results.

It will be argued mathematically that the assumption of centralized control, although useful in many situations, is not necessary. Decentralized systems which communicate perfectly can carry out exactly the same control as a centralized controller. This broad conclusion is not new but many of the details from Chapter 3 are novel. Thus the assumption of centralized control is equivalent to the assumption of perfect communication between decentralized controllers. As perfect communication can only occur

[7]Quoted in M.J. Klein, Thermodynamics in Einsteins Universe, in Science, 157 (1967).
[8]Non-negative real matrices whose columns sum to 1.

in the perfect world of mathematics (due to limited bandwidth), real world centralized control is an illusion, an approximation of reality. It exists only in the eye of the observer. Decentralization is the reality. As discussed above, centralization as an illusion is not a new concept. Admittedly, in a very large number of real world situations the centralized assumption is very useful. However, as systems scale to ever larger numbers of decentralized controllers, the demands on communication to maintain the illusion of centralized control may be overwhelming. A general theory of decentralized control has as a simplified case, centralized control. From a practical viewpoint, a hybrid of the two models is often used. For example, in a group of robots we might imbue each individual robot with a centralized control model (within itself) while treating the group as a decentralized system.

Chapter 4 will present an example of a real world robotics system[9] which employs communicating decentralized controllers. Decentralized approaches to robotics are important as they provide redundancy and modularity. These attributes are able to greatly enhance the autonomy of artificial systems and help to reduce the need for human supervision of robots. The question this chapter seeks to answer is

*Is decentralized control (with communication) practical for real world robotics systems?*

It will be shown that not only is it practical but, in some cases, necessary (i.e., when the communication bandwidth is too low to support the centralized assumption). Depending on the task, decentralized control can provide a much more efficient use of the communication facilities than a centralized control model. The use of well established communication protocols in a group of real mobile robots is not unique to this thesis but it certainly is not widespread. It will be shown that communication between mobile robots is a natural extension of behaviour-based control [Brooks, 1986] within a single robot. Behaviours such as clustering, flocking, and shape-forming are designed using the bottom-up philosophy and generic communication modules. Other practical advantages of decentralized control for a group of mobile robots will be discussed.

Chapter 5 returns to a more theoretical problem, namely that of decentralized decision making. In the preceding chapters the issue of coordinating controls before execution will be shown to be important. This is the question of *self-organization.* Bushev [1994] writes

> Self-organization has been generally defined as the process in which the organization of complex systems is being created, reproduced or improved.

This is a very general definition and researchers often take a narrower view in the context of their miniscule research focus (the present author included). A simple example of self-organization is considered by asking the question

*How can many communicating agents come to a common decision?*

The answer to this question becomes very interesting when the communication links between the controllers are sparse[10]. This chapter will describe a stochastic model for bottom-up, decentralized decision making based on cellular automata. This, too, will be cast in the stochastic algebra of Chapter 2. This mechanism allows an arbitrarily connected group of agents to spontaneously generate a piece of common information without the use of any centralized facility. This basic decision-making tool makes many

[9] A portion of the robotics system described here was constructed as part of this thesis. See acknowledgements.
[10] Each controller has direct communication links to only a subset of all other controllers.

different system behaviours possible. A stochastic instability will be required to do this, thus creating a conceptual link to the creation of order in nonlinear physical systems. The instability may be thought of as a feedback mechanism such that decentralized decision making is itself a problem of control. Self-organization is concerned with no less than the creation of pattern (order, unity, coherence) in a system of agents. If we prescribe to the bottom-up/materialist philosophy then the basic mechanisms of pattern creation must be understood before attempting to build something that is "alive" or "intelligent". We need to identify and understand the basic dynamic mechanisms and then enable them artificially. We must further recognize that self-organization, life, intelligence are dynamic processes, not static configurations. Observers of such dynamic processes must be careful to note on what *timescale* the system is considered to be self-organizing.

The last issue which must be mentioned is that although we require decentralized controllers to *function* independently, this does not mean that they must be *designed* independently as well. We must distinguish between the implementation of decentralized controllers and their design. The issue of decentralized design (e.g., through learning and evolution which are also stochastic decentralized processes) is beyond the scope of this thesis.

The original motivation for study in this area was the control of a small network of mobile robots. However, the study of decentralized systems has the potential to shed light on much more than this narrow application. The coordination of internet software agents[11] is becoming a major area of research. The entire biological world can be described on one level or another (e.g., cells, organisms, species) as a decentralized system. At the molecular level, physical systems (e.g., gases, solids, fluids) may be described as decentralized (e.g., interacting statistical mechanical systems). Automobile traffic is becoming a popular topic to study in the framework of decentralized systems. Understanding how many units can be organized to work together may even help explain how our brains are organized (i.e., how $10^{11}$ neurons with sparse connections can function coherently).

These are indeed lofty aspirations but this field is yet in its youth. The advent of the modern computer has only in recent years made the study of large numbers of simple units possible through simulation and visualization of artificial systems[12]. Most of the great scientific achievements of the past have been reductionist in nature. It is now the era of putting the pieces of the puzzle back together to study the interactions between them. This can be seen in many fields including that of control engineering. Here decentralized control methods are still not mainstream but they are certainly becoming more prevalent as larger and more complicated systems are being developed. It is hoped that this study of stochastic decentralized systems from the control vantage point will shed light on the immediate engineering problem at hand while also making a small contribution to understanding how natural systems function.

[11] Autonomous programs which roam the internet gathering/distributing information. Used frequently in data mining.
[12] Great progress is even being made in the areas of social science [Epstein and Axtell, 1996] through such simulations.

We find ourselves in a world in which
reversibility and determinism apply only to
limiting, simple cases, while irreversibility and
randomness are the rules.
—Ilya Prigogine
ORDER OUT OF CHAOS, 1984

# Chapter 2

# A STOCHASTIC ALGEBRA

This chapter constructs an algebra based on *stochastic matrices* which can be viewed as both probability distributions and, as we will see, vectors. A stochastic matrix is a real matrix whose columns sum to 1 and whose entries are nonnegative. The axiom of total probability is thus implicitly satisfied when using stochastic matrices as the columns may be thought of as probability distributions. Such matrices are commonly used in the study of *Markov chains*, which were named for the Russian mathematician Andrei Andreevich Markov (1856-1922), who was one of the first to study them[1]. Markov, however, used these chains to study probability theory, never applying them to the sciences. They have subsequently been used, for example, in the study of population dynamics, human speech, and economics.

Interestingly, Markov chains may be described in terms of the thermodynamic quantity, *entropy*. At the distribution level, there is a progressive uniformity of the system which may be quantified using entropy. The initial conditions are gradually forgotten as the system tends towards a stable equilibrium. This is a direct consequence of what is called the Markov property which simply stated says there are well defined probabilities, describing the transitions of the system, which are independent of the previous history. These simple mathematical models thus have embedded in them an *arrow of time*; they are irreversible at the distribution (or *ensemble*) level. At the level of single trajectories, however, the system fluctuates. In fact, if we wait long enough, it is possible recover the initial conditions (under certain conditions), but this could take a very long time. Fluctuations will become extremely important in what is to follow.

[1] Poincaré was another.

Markov chains display a gradual destruction of initial conditions, but what about the formation of new structures? If the system always progresses towards uniformity, how may new structures be formed? This will be discussed in much depth in the next chapter on control. If there are certain parameters in a Markov system which may be influenced externally, it becomes possible to control how the system behaves. Control theories for systems involving linear systems are readily available [Kalman, 1960]. For a modern treatment, see, for example, [Antsaklis and Michel, 1997]. Linear system theory presupposes the existence of a *linear algebra*, a role which is usually filled by matrix algebra, the familiar linear algebra involving *real* matrices. Unfortunately, stochastic matrices by themselves may not be added, subtracted, multiplied using the usual operations of matrix algebra. For example, the zero matrix (all zeros) of matrix algebra violates the axiom of total probability. It is not a stochastic matrix.

By redefining the algebraic operations (e.g., addition, scalar multiplication, and vector product) to be more suited to probability theory, the set of stochastic matrices may be shown to constitute a vector space, an inner product space, and an algebra. The work in this chapter began with the simple idea that the new zero matrix should be the uniform probability distribution. This was motivated by stable behaviour of Markov chains as they progress towards a state of maximum entropy. The next step was to relate vector addition to statistical independence. The rest of this lengthy chapter fell into place very naturally once the notion of an algebra was considered. This new formulation has been called *stochastic algebra* and may be used to frame the control theory (and other stochastic decentralized theory) of the subsequent chapters. For example, it will become possible to consider the familiar classic linear system in the context of Markov systems. For reference the reader may use any good text on modern linear algebra (e.g., [Greub, 1974] or [Fraleigh and Beauregard, 1989]). A basic text on calculus may also be useful.

## 2.1 Definitions

We begin with a few definitions[2].

**Definition.** STOCHASTIC MATRICES: The *closed set* of stochastic matrices ${}^m\overline{\mathbb{S}}^n$ is

$$ {}^m\overline{\mathbb{S}}^n = \left\{ \boldsymbol{A} = [a_{ij}] \in {}^m\mathbb{R}^n \;\middle|\; \sum_{i=1}^{m} a_{ij} = 1,\ a_{ij} \geq 0 \right\} $$

while the *open set* of stochastic matrices ${}^m\mathbb{S}^n$ is

$$ {}^m\mathbb{S}^n = \left\{ \boldsymbol{A} = [a_{ij}] \in {}^m\overline{\mathbb{S}}^n \;\middle|\; a_{ij} > 0 \right\} $$

Thus the *boundary* of the set of stochastic matrices ${}^m\partial\mathbb{S}^n$ is

$$ {}^m\partial\mathbb{S}^n = {}^m\overline{\mathbb{S}}^n - {}^m\mathbb{S}^n $$

Each column of a stochastic matrix may be thought of as a probability distribution over $m$ unordered "states". Most of this chapter will proceed to construct an algebra over the open set of stochastic matrices, ${}^m\mathbb{S}^n$. We will see that the boundary of the set, ${}^m\partial\mathbb{S}^n$ is difficult to deal with in this algebra

[2] Using ${}^m\mathbb{R}^n$ to mean the set of real matrices with $m$ rows and $n$ columns.

as its members must be represented in the limit. The algebraic operators (to follow) can still be used with the boundary members but only with appropriate limit functions. Within the boundary, in the case that only one state is occupied (with probability 1), the matrix is called *deterministic.*

**Definition.** DETERMINISTIC MATRICES: The set of *deterministic matrices* ${}^m\mathbb{D}^n$ is

$$ {}^m\mathbb{D}^n = \left\{ \mathbf{A} = [a_{ij}] \in {}^m\partial\mathbb{S}^n \;\middle|\; \forall j \; \exists i, \;\; a_{ij} = 1 \right\} $$

The deterministic matrices will not per se be included in the algebraic framework (because they lie in ${}^m\partial\mathbb{S}^n$). However, some of the algebraic operations can apply to deterministic matrices as shall be noted.

When all states are equally probable we have a uniform probability distribution. When all columns are uniform this may be represented by the *uniform matrix.*

**Definition.** UNIFORM MATRIX: The *uniform matrix* ${}^m\mathbf{\Omega}^n \in {}^m\mathbb{S}^n$ is

$$ {}^m\mathbf{\Omega}^n = [\omega_{ij}], \;\; \omega_{ij} = \frac{1}{m} $$

It will often be referred to simply as $\mathbf{\Omega}$. When $n = 1$ we will use ${}^m\boldsymbol{\omega}$ or $\boldsymbol{\omega}$ instead.

**Example.** Below are examples of a stochastic matrix, $\boldsymbol{A} \in {}^3\mathbb{S}^2$, a deterministic matrix, $\mathbf{B} \in {}^3\mathbb{D}^2$, and the uniform matrix, $\mathbf{\Omega} \in {}^3\mathbb{S}^2$.

$$ \boldsymbol{A} = \begin{bmatrix} \frac{1}{2} & \frac{1}{12} \\ \frac{1}{4} & \frac{7}{12} \\ \frac{1}{4} & \frac{1}{3} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{\Omega} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \end{bmatrix} $$

□

We will also have need of the usual identity matrix.

**Definition.** IDENTITY MATRIX: The *identity matrix* ${}^m\mathbf{1}^m \in {}^m\mathbb{D}^m$ is

$$ {}^m\mathbf{1}^m = [\delta_{ij}] $$

where $\delta_{ij}$ is the *Kroenecker delta.* It will often be referred to simply as $\mathbf{1}$.

Some new operators are now introduced. These are rooted in probability theory and will be exploited frequently in what is to follow.

**Definition.** NORMALIZATION: The *normalization* operator denoted $\downarrow\boldsymbol{R}$, where $\boldsymbol{R} = [r_{ij}] \in {}^m\mathbb{R}^n$ with $r_{ij} \geq 0$ and $(\forall j) \; \sum_{i=1}^m r_{ij} \neq 0$, is

$$ \downarrow\boldsymbol{R} = \left[ \frac{r_{ij}}{\sum_{i=1}^m r_{ij}} \right] $$

This operation renders any real matrix (with no zero columns) a stochastic matrix ( $\downarrow\boldsymbol{R} \in {}^m\overline{\mathbb{S}}^n$ ).

**Definition.** : Let $\boldsymbol{A} = [a_{ij}]$, $\boldsymbol{B} = [b_{ij}] \in {}^m\overline{\mathbb{S}}^n$ . Then $\boldsymbol{A} = \boldsymbol{B}$ if and only if $a_{ij} = b_{ij} \; \forall i = 1 \ldots m, \; j = 1 \ldots n$.

**Lemma.** Let $\boldsymbol{R} = [r_{ij}], \boldsymbol{S} = [s_{ij}] \in {}^{m}\mathbb{R}^{n}$ and $r_{ij}, s_{ij} \geq 0$. Then $\downarrow\boldsymbol{R} = \downarrow\boldsymbol{S}$ if and only if $(\forall i, j),\ (\exists \lambda_j > 0),\ r_{ij} = \lambda_j s_{ij}$.

**Definition.** UNBIASED INVERSE: The *unbiased inverse*[3] operator, denoted $\boldsymbol{A}^{\tau}$, where $\boldsymbol{A} = [a_{ij}] \in {}^{m}\overline{\mathbb{S}}^{n}$, is

$$\boldsymbol{A}^{\tau} = \downarrow[a_{ji}]$$

Note, in the case that $\sum_j a_{kj} = 0$ then the $k^{th}$ column of $\boldsymbol{A}^{\tau}$ is defined to be the uniform column (this is a limiting case).

**Definition.** VECTOR ADDITION: Let $\boldsymbol{A} = [a_{ij}]$, $\boldsymbol{B} = [b_{ij}] \in {}^{m}\overline{\mathbb{S}}^{n}$. The *vector addition* of $\boldsymbol{A}$ and $\boldsymbol{B}$, denoted $\boldsymbol{A} \oplus \boldsymbol{B}$, is

$$\boldsymbol{A} \oplus \boldsymbol{B} = \downarrow[a_{ij} b_{ij}]$$

In the case that the operands are deterministic, vector addition must be computed in the limit. Appropriate limit functions must be chosen to extend vector addition to deterministic matrices.

**Example.** As an example of vector addition in the limit, consider $\mathbf{A} = [1\ 0\ 0]^{\mathrm{T}}$ and $\mathbf{B} = [0\ 1\ 0]^{\mathrm{T}}$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \lim_{\epsilon \to 0} \left( \left[ \begin{bmatrix} 1-2\epsilon \\ \epsilon \\ \epsilon \end{bmatrix} \oplus \begin{bmatrix} \epsilon \\ 1-2\epsilon \\ \epsilon \end{bmatrix} \right] \right) = \lim_{\epsilon \to 0} \left( \frac{1}{2\epsilon - 3\epsilon^2} \begin{bmatrix} \epsilon - 2\epsilon^2 \\ \epsilon - 2\epsilon^2 \\ \epsilon^2 \end{bmatrix} \right)$$

$$= \lim_{\epsilon \to 0} \left( \frac{1}{2 - 6\epsilon} \begin{bmatrix} 1-4\epsilon \\ 1-4\epsilon \\ 2\epsilon \end{bmatrix} \right) = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 0 \end{bmatrix}$$

where l'Hôpital's rule has been employed. □

**Definition.** SCALAR MULTIPLICATION: Let $\boldsymbol{A} = [a_{ij}] \in {}^{m}\overline{\mathbb{S}}^{n}$ and $\lambda \in \mathbb{R}$. The scalar multiplication of $\lambda$ with $\boldsymbol{A}$, denoted $\lambda \cdot \boldsymbol{A}$, is

$$\lambda \cdot \boldsymbol{A} = \downarrow[a_{ij}^{\lambda}]$$

In the case that $\boldsymbol{A}$ is deterministic, scalar multiplication must be computed in the limit. Again, appropriate limit functions must be chosen to allow this operation to extend to deterministic matrices.

**Example.** As an example of scalar multiplication in the limit, consider $\mathbf{A} = [1\ 0\ 0]^{\mathrm{T}}$ and $\lambda = -1$

$$(-1) \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \lim_{\epsilon \to 0} \left( \downarrow \begin{bmatrix} (1-2\epsilon)^{-1} \\ \epsilon^{-1} \\ \epsilon^{-1} \end{bmatrix} \right) = \lim_{\epsilon \to 0} \left( \frac{1}{\frac{1}{1-2\epsilon} + \frac{1}{\epsilon} + \frac{1}{\epsilon}} \begin{bmatrix} \frac{1}{1-2\epsilon} \\ \frac{1}{\epsilon} \\ \frac{1}{\epsilon} \end{bmatrix} \right)$$

$$= \lim_{\epsilon \to 0} \left( \frac{1}{\frac{\epsilon}{1-2\epsilon} + 1 + 1} \begin{bmatrix} \frac{\epsilon}{1-2\epsilon} \\ 1 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$$

where we have multiplied through by $\epsilon$ in order to compute the limit. □

[3] In the case of a deterministic matrix, $\mathbf{A} \in \mathbb{D}$, the unbiased inverse is equivalent to the Moore-Penrose pseudoinverse. This is not the case for general stochastic matrices. These will be discussed in more depth later.

## 2.2 A Vector Space

With these definitions in hand it is possible to show that the open set of stochastic matrices is a vector space.

**Proposition.** *The set ${}^m\mathbb{S}^n$ is a vector space over the field $\mathbb{R}$ under the vector addition and scalar multiplication defined above.*

**Proof.** There are eight axioms which must be satisfied. For all $\boldsymbol{A} = [a_{ij}], \boldsymbol{B} = [b_{ij}], \boldsymbol{C} = [c_{ij}] \in {}^m\mathbf{S}^n$:

**AI.** CLOSURE: $\boldsymbol{A} \oplus \boldsymbol{B} \in {}^m\mathbf{S}^n$.

Obvious by definition.

**AII.** ASSOCIATIVITY: $(\boldsymbol{A} \oplus \boldsymbol{B}) \oplus \boldsymbol{C} = \boldsymbol{A} \oplus (\boldsymbol{B} \oplus \boldsymbol{C})$.

$$\begin{aligned}(\boldsymbol{A} \oplus \boldsymbol{B}) \oplus \boldsymbol{C} &= \downarrow[a_{ij}b_{ij}] \oplus [c_{ij}] \\ &= \downarrow[a_{ij}b_{ij}c_{ij}] \\ &= [a_{ij}] \oplus \downarrow[b_{ij}c_{ij}] \\ &= \boldsymbol{A} \oplus (\boldsymbol{B} \oplus \boldsymbol{C})\end{aligned}$$

**AIII.** ZERO: *There exists a* zero *or* null *stochastic matrix $\boldsymbol{\Omega} \in {}^m\mathbf{S}^n$ such that $\boldsymbol{A} \oplus \boldsymbol{\Omega} = \boldsymbol{A}$.*

The zero stochastic matrix is $\boldsymbol{\Omega} = \downarrow[1] \equiv [1/m]$.

**AIV.** INVERSE: *There exists an* inverse $\ominus\boldsymbol{A} \in {}^m\mathbf{S}^n$ *such that* $\boldsymbol{A} \oplus (\ominus\boldsymbol{A}) = \boldsymbol{\Omega}$.

The inverse of $\boldsymbol{A}$ is $(-1)\cdot\boldsymbol{A} = \downarrow[a_{ij}^{-1}]$.

In addition, for all $\boldsymbol{A} = [a_{ij}], \boldsymbol{B} = [b_{ij}] \in {}^m\mathbf{S}^n$ and all $\lambda, \mu \in \mathbb{R}$:

**MI.** CLOSURE: $\lambda\cdot\boldsymbol{A} \in {}^m\mathbf{S}^n$.

Obvious by definition.

**MII.** ASSOCIATIVITY: $\lambda\cdot(\mu\cdot\boldsymbol{A}) = (\lambda\mu)\cdot\boldsymbol{A}$.

$$\begin{aligned}\lambda\cdot(\mu\cdot\boldsymbol{A}) &= \lambda\cdot(\downarrow[a_{ij}^{\mu}]) \\ &= \downarrow[(a_{ij}^{\mu})^{\lambda}] \\ &= \downarrow[a_{ij}^{\lambda\mu}] \\ &= (\lambda\mu)\cdot\boldsymbol{A}\end{aligned}$$

**MIII.** DISTRIBUTIVITY: (*a*) $(\lambda+\mu)\cdot\boldsymbol{A} = \lambda\cdot\boldsymbol{A} \oplus \mu\cdot\boldsymbol{A}$ and (*b*) $\lambda\cdot(\boldsymbol{A} \oplus \boldsymbol{B}) = \lambda\cdot\boldsymbol{A} \oplus \lambda\cdot\boldsymbol{B}$.

$$\begin{aligned}(a)\quad (\lambda+\mu)\cdot\boldsymbol{A} &= \downarrow[a_{ij}^{(\lambda+\mu)}] \\ &= \downarrow[a_{ij}^{\lambda}a_{ij}^{\mu}] \\ &= \downarrow[a_{ij}^{\lambda}] \oplus \downarrow[a_{ij}^{\mu}] \\ &= \lambda\cdot\boldsymbol{A} \oplus \mu\cdot\boldsymbol{A}\end{aligned}$$

$$\begin{aligned}(b)\quad \lambda\cdot(\boldsymbol{A} \oplus \boldsymbol{B}) &= \lambda\cdot(\downarrow[a_{ij}b_{ij}]) \\ &= \downarrow[(a_{ij}b_{ij})^{\lambda}] \\ &= \downarrow[a_{ij}^{\lambda}b_{ij}^{\lambda}] \\ &= \downarrow[a_{ij}^{\lambda}] \oplus \downarrow[b_{ij}^{\lambda}] \\ &= \lambda\cdot\boldsymbol{A} \oplus \lambda\cdot\boldsymbol{B}\end{aligned}$$

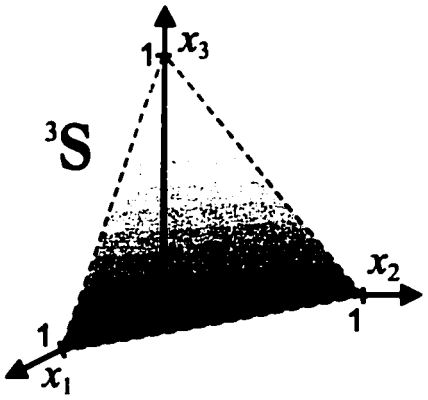



Figure 2.1: Graphical depiction of the vector space, $^3\mathbb{S}$, in relation to the usual Cartesian space, $^3\mathbb{R}$. The new vector space is the two-dimensional shaded triangular surface shown with the zero vector, $\omega$, marked in the center.

**MIV.** IDENTITY: *For the identity element* $1 \in \mathbb{R}$, $1 \cdot \boldsymbol{A} = \boldsymbol{A}$.

$$1 \cdot \boldsymbol{A} = \downarrow[a_{ij}^1] = [a_{ij}] = \boldsymbol{A}$$

Therefore, a vector space. ■

Having thus shown that $^m\mathbb{S}^n$ does constitute a vector space under the appropriately defined vector addition and scalar multiplication, we have recourse to a wealth of results. Among these are the following:

$$0 \cdot \boldsymbol{A} = \boldsymbol{\Omega}, \quad \forall \boldsymbol{A} \in {}^m\mathbb{S}^n$$

$$\lambda \cdot \boldsymbol{\Omega} = \boldsymbol{\Omega}, \quad \forall \lambda \in \mathbb{R}$$

$$(-1) \cdot \boldsymbol{A} = \ominus \boldsymbol{A}, \quad \forall \boldsymbol{A} \in {}^m\mathbb{S}^n$$

$$(-\lambda) \cdot \boldsymbol{A} = \lambda \cdot (\ominus \boldsymbol{A}) = \ominus(\lambda \cdot \boldsymbol{A}), \quad \forall \boldsymbol{A} \in {}^m\mathbb{S}^n, \lambda \in \mathbb{R}$$

These can be easily verified from the definitions but, in fact, they are general vector-space results. Figure 2.1 depicts the vector space, $^3\mathbb{S}$, with the zero vector, $\omega \in {}^3\mathbb{S}$.

## 2.3 An Inner Product Space

An *inner product* associated with $^m\mathbb{S}^n$ is presented.

**Definition.** INNER PRODUCT: Let $\boldsymbol{x} = [x_i]$, $\boldsymbol{y} = [y_i] \in {}^m\mathbb{S}$. Then

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = [\ln x_i]^{\mathrm{T}} \boldsymbol{\Lambda} [\ln y_i]$$

where normal scalar multiplication, matrix addition, and matrix multiplication are used[4]. The matrix $\boldsymbol{\Lambda} \in {}^m\mathbb{R}^m$ is

$$\boldsymbol{\Lambda} = \mathbf{1} - \boldsymbol{\Omega}$$

[4]When working directly with probabilities we must use the new concepts of scalar multiplication, vector addition, and (soon to be presented) vector multiplication. However, when working with the natural logarithm of probabilities (which we will see may be interpreted as the *coordinates* of a vector), we revert to the familar operations of matrix algebra.

For convenience we will sometimes write $\langle \boldsymbol{A}, \boldsymbol{B} \rangle$ where $\boldsymbol{A},\ \boldsymbol{B} \in {}^{m}\mathbb{S}^{n}$. By this we mean

$$\langle \boldsymbol{A}, \boldsymbol{B} \rangle = [\langle \boldsymbol{a}_i, \boldsymbol{b}_j \rangle],\ \forall i,\ j = 1 \ldots n$$

where $\boldsymbol{a}_i, \boldsymbol{b}_i, i = 1 \ldots n$ are the columns of $\boldsymbol{A}$, $\boldsymbol{B}$. Note, $\langle \boldsymbol{A}, \boldsymbol{B} \rangle \in {}^{n}\mathbb{R}^{n}$ is not an inner product itself as it maps to a real matrix not a scalar.

Note it follows that, if $\boldsymbol{u} = [u_i],\ \boldsymbol{v} = [v_i] \in {}^{m}\mathbb{R}$, then

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle = [\ln u_i]^{\mathrm{T}} \boldsymbol{\Lambda} [\ln v_i]$$

That is, the normalization factor does not matter. The reason this works is that we are using logarithms and the rows (and columns) of $\boldsymbol{\Lambda}$ sum to 0. Using the inner product defined above we are now prepared to make the following claim[5].

**Proposition.** *The product* $\langle \cdot, \cdot \rangle$ *defined above is an* inner product *on* ${}^{m}\mathbb{S}$.

**Proof.** There are four axioms which must be satisfied. For all $\boldsymbol{x} = [x_i], \boldsymbol{y} = [y_i], \boldsymbol{z} = [z_i] \in {}^{m}\mathbf{S}$ and $\lambda \in \mathbb{R}$:

**PI.** DISTRIBUTIVITY: $\langle \boldsymbol{x}, \boldsymbol{y} \oplus \boldsymbol{z} \rangle = \langle \boldsymbol{x}, \boldsymbol{y} \rangle + \langle \boldsymbol{x}, \boldsymbol{z} \rangle$.

$$\begin{aligned}
\langle \boldsymbol{x}, \boldsymbol{y} \oplus \boldsymbol{z} \rangle &= [\ln x_i]^{\mathrm{T}} \boldsymbol{\Lambda} \left[ \ln \left( \frac{y_i z_i}{\sum_{j=1}^{m} y_j z_j} \right) \right] \\
&= [\ln x_i]^{\mathrm{T}} \boldsymbol{\Lambda} [\ln y_i z_i] \\
&= [\ln x_i]^{\mathrm{T}} \boldsymbol{\Lambda} [\ln y_i] + [\ln x_i]^{\mathrm{T}} \boldsymbol{\Lambda} [\ln z_i] \\
&= \langle \boldsymbol{x}, \boldsymbol{y} \rangle + \langle \boldsymbol{x}, \boldsymbol{z} \rangle
\end{aligned}$$

**PII.** COMMUTATIVITY: $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{y}, \boldsymbol{x} \rangle$.

Obvious by definition.

**PIII.** SCALAR MULTIPLICATION: $\langle \lambda \cdot \boldsymbol{x}, \boldsymbol{y} \rangle = \lambda \langle \boldsymbol{x}, \boldsymbol{y} \rangle$.

$$\begin{aligned}
\langle \lambda \cdot \boldsymbol{x}, \boldsymbol{y} \rangle &= \langle \Downarrow [x_i^{\lambda}], \boldsymbol{y} \rangle \\
&= \left[ \ln \frac{x_i^{\lambda}}{\sum_{j=1}^{m} x_j^{\lambda}} \right]^{\mathrm{T}} \boldsymbol{\Lambda} [\ln y_i] \\
&= \left[ \ln x_i^{\lambda} \right]^{\mathrm{T}} \boldsymbol{\Lambda} [\ln y_i] \\
&= [\lambda \ln x_i]^{\mathrm{T}} \boldsymbol{\Lambda} [\ln y_i] \\
&= \lambda [\ln x_i]^{\mathrm{T}} \boldsymbol{\Lambda} [\ln y_i] \\
&= \lambda \langle \boldsymbol{x}, \boldsymbol{y} \rangle
\end{aligned}$$

**PIV.** POSITIVE-DEFINITENESS: $\langle \boldsymbol{x}, \boldsymbol{x} \rangle > 0$ *if* $\boldsymbol{x} \neq \boldsymbol{\omega}$.

Since $\boldsymbol{\Lambda}$ is symmetric there exists a matrix $\mathsf{A} \in {}^{m-1}\mathbb{R}^{m}$ such that $\boldsymbol{\Lambda} = \mathsf{A}^{\mathrm{T}}\mathsf{A}$. If $\boldsymbol{x} \neq \boldsymbol{\omega}$, then $x_i \neq x_j$ for some $i \neq j$. Accordingly, $\mathsf{A}[\ln x_i] \neq \boldsymbol{\omega}$. Thus

$$\langle \boldsymbol{x}, \boldsymbol{x} \rangle = (\mathsf{A}[\ln x_i])^{\mathrm{T}} (\mathsf{A}[\ln x_i]) > 0$$

It in fact follows that $\langle \boldsymbol{x}, \boldsymbol{x} \rangle = 0$ if and only if $\boldsymbol{x} = \boldsymbol{\omega}$. We thus have that ${}^{m}\mathbf{S}$ (and, more generally, ${}^{m}\mathbf{S}^{n}$) is an *inner-product space.* ■

[5] We will consider only stochastic columns but the proof can be easily extended to general stochastic matrices.

## 2.4 Bases

As ${}^m\mathbb{S}$ is a vector space[6] we may express every vector $\boldsymbol{x}$ in ${}^m\mathbb{S}$ uniquely in the form

$$\boldsymbol{x} = \bigoplus_{i=1}^{m-1} \lambda_i \cdot \boldsymbol{b}_i$$

where $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{m-1}\}$ is a *basis* for ${}^m\mathbb{S}$ with $\boldsymbol{b}_i \in {}^m\mathbb{S}, \lambda_i \in \mathbb{R}, i = 1 \ldots (m-1)$. The $\lambda_i$ may be thought of as the *coordinates* of $\boldsymbol{x}$ with respect to the basis. Note, we only need $m-1$ vectors to form a basis due to the constraint that columns of stochastic matrices must sum to 1. We now introduce a special matrix which we call the *exponential identity*. The reason behind the name will become clear in the next section.

**Definition.** EXPONENTIAL IDENTITY: The *exponential identity* ${}^m\boldsymbol{\Xi}^m \in {}^m\mathbb{S}^m$ is

$$\begin{aligned} {}^m\boldsymbol{\Xi}^m &= \downarrow[\xi_{ij}] \\ \xi_{ij} &= e^{\delta_{ij}}, \ \forall i,j = 1 \ldots m \end{aligned}$$

where $\delta_{ij}$ is the Kroenecker delta. For brevity $\boldsymbol{\Xi}$ will often be used. Note that $\boldsymbol{\Xi} = \boldsymbol{\Xi}^{\mathsf{T}} = \boldsymbol{\Xi}^{\mathcal{T}}$.

**Proposition.** *Any $m-1$ columns (or rows) of ${}^m\boldsymbol{\Xi}^m$ may be used as a basis for ${}^m\mathbb{S}$.*

**Proof.** We will show the desired result for the first $m-1$ columns of $\boldsymbol{\Xi}$. Let $\boldsymbol{\xi}_i \ \forall i = 1 \ldots m$ represent the $i^{th}$ column of ${}^m\boldsymbol{\Xi}^m$. We see that $\boldsymbol{x} = [x_i] \in {}^m\mathbb{S}$ can be written as

$$\begin{aligned} \boldsymbol{x} &= [x_i] \\ &= \left[e^{\ln x_i}\right] \\ &= \downarrow\left[e^{\ln x_i - \ln x_m}\right] \\ &= \begin{bmatrix} e^{\ln x_1 - \ln x_m} \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ e^{\ln x_2 - \ln x_m} \\ \vdots \\ 1 \\ 1 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} 1 \\ 1 \\ \vdots \\ e^{\ln x_{m-1} - \ln x_m} \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \\ &= (\ln x_1 - \ln x_m) \cdot \begin{bmatrix} e \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \oplus (\ln x_2 - \ln x_m) \cdot \begin{bmatrix} 1 \\ e \\ \vdots \\ 1 \\ 1 \end{bmatrix} \oplus \cdots \oplus (\ln x_{m-1} - \ln x_m) \cdot \begin{bmatrix} 1 \\ 1 \\ \vdots \\ e \\ 1 \end{bmatrix} \\ &= \bigoplus_{i=1}^{m-1} (\ln x_i - \ln x_m) \cdot \boldsymbol{\xi}_i \end{aligned}$$

where we may at last interpret the $[\ln x_i - \ln x_m]$ as the *coordinates* of the vector $\boldsymbol{x}$ in the basis defined by the $m-1$ first columns of $\boldsymbol{\Xi}$. ■

[6] Bases are discussed for stochastic columns only but could easily be generalized to stochastic matrices.

Although we have shown $\Xi$ provides a basis for ${}^m\mathbb{S}$, this result is easily generalized to ${}^m\mathbb{S}^n$. Subsequent usage of the term *coordinates* (without reference to a basis) will imply coordinates in this basis. The coordinates of $\boldsymbol{x}$ will be denoted $\lfloor \boldsymbol{x} \rfloor = [\ln x_i - \ln x_m] \in {}^{m-1}\mathbb{R}$. Note, although we technically only need $m-1$ vectors to form the basis for ${}^m\mathbb{S}$, we could include $m$ for the sake of symmetry.

$$\boldsymbol{x} = \bigoplus_{i=1}^{m-1} (\ln x_i - \ln x_m) \cdot \boldsymbol{\xi}_i = \bigoplus_{i=1}^{m} \ln x_i \cdot \boldsymbol{\xi}_i$$

Vector addition always renormalizes so we may use $m$ with no ill effect. We will refer to *augmented coordinates*, denoted $\lceil \boldsymbol{x} \rceil = [\ln x_i] \in {}^m\mathbb{R}$ but must remember that there is a constraint and the vector space is really only of dimension $m-1$. It is easy to verify that the $\boldsymbol{\xi}_i$ do not form an *orthonormal* basis.

**Definition.** ORTHONORMAL BASIS: Let $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{m-1}\}$ be a *basis* for ${}^m\mathbb{S}$. This basis is *orthonormal* if and only if

$$\langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle = \delta_{ij} \quad \forall i, j = 1 \ldots m-1$$

where $\delta_{ij}$ is the Kroenecker delta.

**Example.** The set of stochastic matrices

$$\left\{ \downarrow \begin{bmatrix} e \\ e \\ 1 \\ 1 \end{bmatrix}, \downarrow \begin{bmatrix} e \\ 1 \\ e \\ 1 \end{bmatrix}, \downarrow \begin{bmatrix} e \\ 1 \\ 1 \\ e \end{bmatrix} \right\}$$

is an orthonormal basis for ${}^4\mathbb{S}$. □

## 2.5 An Algebra

With a valid basis in hand it is now possible to further make the claim that we have an *algebra* for stochastic matrices, or a *stochastic algebra*[7]. To justify this claim, we further require a vector product (in addition to the already established vector space) which satisfies a few axioms.

**Definition.** VECTOR PRODUCT: Let $\boldsymbol{A} \in {}^m\mathbb{S}^n$, $\boldsymbol{B} \in {}^n\mathbb{S}^p$. Let $\boldsymbol{a}_{r,i}$ represent the $i^{th}$ row of $\boldsymbol{A}$ and $\boldsymbol{b}_{c,j}$ represent the $j^{th}$ column of $\boldsymbol{B}$. The *vector product* of $\boldsymbol{A}$ and $\boldsymbol{B}$, denoted $\boldsymbol{A} \otimes \boldsymbol{B}$, is

$$\boldsymbol{A} \otimes \boldsymbol{B} = \downarrow \left[ e^{\langle \boldsymbol{a}_{r,i}^{\dagger}, \boldsymbol{b}_{c,j} \rangle} \right]$$

Note that $\boldsymbol{A} \otimes \boldsymbol{B} \in {}^m\mathbb{S}^p$

Technically speaking, for an algebra the vector product should take two operands from the same vector space and produce a third vector from that space. This may be achieved if we set $m = n = p$. However, the results presented still hold when this is not the case.

**Proposition.** *The vector space together with the vector product defined above constitutes an associative algebra.*

[7] This name is great as it is in complete contrast to "fuzzy logic".

**Proof.** There are four axioms which must be satisfied.

**TI.** LEFT DISTRIBUTIVITY: $\boldsymbol{A}\otimes(\boldsymbol{B}\oplus\boldsymbol{C})=\boldsymbol{A}\otimes\boldsymbol{B}\oplus\boldsymbol{A}\otimes\boldsymbol{C}$.

Let $\boldsymbol{A}\in{}^{m}\mathbf{S}^{n}$ and $\boldsymbol{B},\ \boldsymbol{C}\in{}^{n}\mathbf{S}^{p}$. Let $\boldsymbol{a}_{\mathrm{r},i}$ represent the $i^{th}$ row of $\boldsymbol{A}$ and $\boldsymbol{b}_{\mathrm{c},j},\ \boldsymbol{c}_{\mathrm{c},j}$ represent the $j^{th}$ columns of $\boldsymbol{B},\ \boldsymbol{C}$.

$$\begin{aligned}\boldsymbol{A}\otimes(\boldsymbol{B}\oplus\boldsymbol{C}) &= \downarrow\left[e^{\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{b}_{\mathrm{c},j}\oplus\boldsymbol{c}_{\mathrm{c},j}\rangle}\right]\\ &= \downarrow\left[e^{\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{b}_{\mathrm{c},j}\rangle+\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{c}_{\mathrm{c},j}\rangle}\right]\\ &= \downarrow\left[e^{\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{b}_{\mathrm{c},j}\rangle}e^{\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{c}_{\mathrm{c},j}\rangle}\right]\\ &= \downarrow\left[e^{\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{b}_{\mathrm{c},j}\rangle}\right]\ \oplus\ \downarrow\left[e^{\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{c}_{\mathrm{c},j}\rangle}\right]\\ &= \boldsymbol{A}\otimes\boldsymbol{B}\oplus\boldsymbol{A}\otimes\boldsymbol{C}\end{aligned}$$

**TII.** RIGHT DISTRIBUTIVITY: $(\boldsymbol{A}\oplus\boldsymbol{B})\otimes\boldsymbol{C}=\boldsymbol{A}\otimes\boldsymbol{C}\oplus\boldsymbol{B}\otimes\boldsymbol{C}$.

Let $\boldsymbol{A},\boldsymbol{B}\in{}^{m}\mathbf{S}^{n}$ and $\boldsymbol{C}\in{}^{n}\mathbf{S}^{p}$. Let $\boldsymbol{a}_{\mathrm{r},i},\ \boldsymbol{b}_{\mathrm{r},i}$ represent the $i^{th}$ rows of $\boldsymbol{A},\ \boldsymbol{B}$ and $\boldsymbol{c}_{\mathrm{c},i}$ represent the $j^{th}$ column of $\boldsymbol{C}$.

$$\begin{aligned}(\boldsymbol{A}\oplus\boldsymbol{B})\otimes\boldsymbol{C} &= \downarrow\left[e^{\langle \boldsymbol{a}^{T}_{\mathrm{r},i}\oplus\boldsymbol{b}^{T}_{\mathrm{r},i},\boldsymbol{c}_{\mathrm{c},j}\rangle}\right]\\ &= \downarrow\left[e^{\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{c}_{\mathrm{c},j}\rangle+\langle \boldsymbol{b}^{T}_{\mathrm{r},i},\boldsymbol{c}_{\mathrm{c},j}\rangle}\right]\\ &= \downarrow\left[e^{\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{c}_{\mathrm{c},j}\rangle}e^{\langle \boldsymbol{b}^{T}_{\mathrm{r},i},\boldsymbol{c}_{\mathrm{c},j}\rangle}\right]\\ &= \downarrow\left[e^{\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{c}_{\mathrm{c},j}\rangle}\right]\ \oplus\ \downarrow\left[e^{\langle \boldsymbol{b}^{T}_{\mathrm{r},i},\boldsymbol{c}_{\mathrm{c},j}\rangle}\right]\\ &= \boldsymbol{A}\otimes\boldsymbol{C}\oplus\boldsymbol{B}\otimes\boldsymbol{C}\end{aligned}$$

**TIII.** SCALAR MULTIPLICATION: $(\lambda\cdot\boldsymbol{A})\otimes(\mu\cdot\boldsymbol{B})=(\lambda\mu)\cdot(\boldsymbol{A}\otimes\boldsymbol{B})$.

Let $\boldsymbol{A}\in{}^{m}\mathbf{S}^{n}$ and $\boldsymbol{B}\in{}^{n}\mathbf{S}^{p}$. Let $\boldsymbol{a}_{\mathrm{r},i}$ represent the $i^{th}$ row of $\boldsymbol{A}$ and $\boldsymbol{b}_{\mathrm{c},j}$ represent the $j^{th}$ column of $\boldsymbol{B}$.

$$\begin{aligned}(\lambda\cdot\boldsymbol{A})\otimes(\mu\cdot\boldsymbol{B}) &= \downarrow\left[e^{\langle \lambda\cdot\boldsymbol{a}^{T}_{\mathrm{r},i},\mu\cdot\boldsymbol{b}_{\mathrm{c},j}\rangle}\right]\\ &= \downarrow\left[e^{\lambda\mu\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{b}_{\mathrm{c},j}\rangle}\right]\\ &= \downarrow\left[\left(e^{\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{b}_{\mathrm{c},j}\rangle}\right)^{\lambda\mu}\right]\\ &= (\lambda\mu)\cdot\downarrow\left[e^{\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{b}_{\mathrm{c},j}\rangle}\right]\\ &= (\lambda\mu)\cdot(\boldsymbol{A}\otimes\boldsymbol{B})\end{aligned}$$

**TIV.** ASSOCIATIVITY: $(\boldsymbol{A}\otimes\boldsymbol{B})\otimes\boldsymbol{C}=\boldsymbol{A}\otimes(\boldsymbol{B}\otimes\boldsymbol{C})$.

Let $\boldsymbol{A}=[a_{ij}]\in{}^{m}\mathbf{S}^{n}$, $\boldsymbol{B}=[b_{ij}]\in{}^{n}\mathbf{S}^{p}$, and $\boldsymbol{C}=[c_{ij}]\in{}^{p}\mathbf{S}^{q}$. Let $\boldsymbol{a}_{\mathrm{r},i}$ and $\boldsymbol{b}_{\mathrm{r},j}$ represent the $i^{th}$ rows of $\boldsymbol{A}$ and $\boldsymbol{B}$. Let $\boldsymbol{b}_{\mathrm{c},j}$ and $\boldsymbol{c}_{\mathrm{c},j}$ represent the $j^{th}$ columns of $\boldsymbol{B}$ and $\boldsymbol{C}$.

$$\begin{aligned}(\boldsymbol{A}\otimes\boldsymbol{B})\otimes\boldsymbol{C} &= \left(\downarrow\left[e^{\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{b}_{\mathrm{c},j}\rangle}\right]\right)\otimes\boldsymbol{C}\\ &= \downarrow\left[e^{\langle (\boldsymbol{A}\otimes\boldsymbol{B})^{T}_{\mathrm{r},i},\boldsymbol{c}_{\mathrm{c},j}\rangle}\right]\end{aligned}$$

where $(\boldsymbol{A}\otimes\boldsymbol{B})_{\mathrm{r},i}$ represents the $i^{th}$ row of $\boldsymbol{A}\otimes\boldsymbol{B}$. We have that

$$\begin{aligned}\langle(\boldsymbol{A}\otimes\boldsymbol{B})^{T}_{\mathrm{r},i},\boldsymbol{c}_{\mathrm{c},j}\rangle &= \left[\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{b}_{\mathrm{c},1}\rangle\cdots\langle \boldsymbol{a}^{T}_{\mathrm{r},i},\boldsymbol{b}_{\mathrm{c},p}\rangle\right]\left[\mathbf{1}-\boldsymbol{\Omega}\right]\left[\ln c_{1j}\cdots\ln c_{pj}\right]^{\mathrm{T}}\\ &= \langle \boldsymbol{a}^{T}_{\mathrm{r},i},\bigoplus_{k=1}^{p}\ln c_{kj}\cdot\left(\boldsymbol{b}_{\mathrm{c},k}\ominus\frac{1}{p}\cdot\bigoplus_{l=1}^{p}\boldsymbol{b}_{\mathrm{c},l}\right)\rangle\end{aligned}$$

To complete the proof we must show that $\bigoplus_{k=1}^{p} \ln c_{kj} \cdot \left( \boldsymbol{b}_{c,k} \ominus \frac{1}{p} \cdot \bigoplus_{l=1}^{p} \boldsymbol{b}_{c,l} \right) = (\boldsymbol{B} \otimes \boldsymbol{C})_{c,j}$ where $(\boldsymbol{B} \otimes \boldsymbol{C})_{c,j}$ represents the $j^{th}$ column of $\boldsymbol{B} \otimes \boldsymbol{C}$. We have

$$\begin{aligned}
\bigoplus_{k=1}^{p} \ln c_{kj} \cdot \left( \boldsymbol{b}_{c,k} \ominus \frac{1}{p} \cdot \bigoplus_{l=1}^{p} \boldsymbol{b}_{c,l} \right) &= \downarrow \left[ \prod_{k=1}^{p} \left( b_{ik} \prod_{l=1}^{p} b_{il}^{-\frac{1}{p}} \right)^{\ln c_{kj}} \right] \\
&= \downarrow \left[ e^{\sum_{k=1}^{p} \ln c_{kj} \left( \ln b_{ik} - \frac{1}{p} \sum_{l=1}^{p} \ln b_{il} \right)} \right] \\
&= \downarrow \left[ e^{\left[ \ln b_{i1} \cdots \ln b_{ip} \right] [\mathbf{1} - \boldsymbol{\Omega}] \left[ \ln c_{1j} \cdots \ln c_{pj} \right]^{\mathrm{T}}} \right] \\
&= \downarrow \left[ e^{\langle \boldsymbol{b}_{r,i}^{r}, \boldsymbol{c}_{c,j} \rangle} \right] \\
&= (\boldsymbol{B} \otimes \boldsymbol{C})_{c,j}
\end{aligned}$$

whereupon

$$\begin{aligned}
(\boldsymbol{A} \otimes \boldsymbol{B}) \otimes \boldsymbol{C} &= \downarrow \left[ e^{\langle \boldsymbol{a}_{r,i}^{r}, \bigoplus_{k=1}^{p} \ln c_{kj} \cdot \left( \boldsymbol{b}_{c,k} \ominus \frac{1}{p} \cdot \bigoplus_{l=1}^{p} \boldsymbol{b}_{c,l} \right) \rangle} \right] \\
&= \downarrow \left[ e^{\langle \boldsymbol{a}_{r,i}^{r}, (\boldsymbol{B} \otimes \boldsymbol{C})_{c,j} \rangle} \right] \\
&= \boldsymbol{A} \otimes (\boldsymbol{B} \otimes \boldsymbol{C})
\end{aligned}$$

Therefore, an associative algebra. ■

Note that the property of commutativity does not hold in this *stochastic algebra.*

**Lemma.** There is a *zero* for the vector product. If $\boldsymbol{A} \in {}^{m}\mathbb{S}^{n}$ then

$$\begin{aligned}
\boldsymbol{\Omega} \otimes \boldsymbol{A} &= \boldsymbol{\Omega} \\
\boldsymbol{A} \otimes \boldsymbol{\Omega} &= \boldsymbol{\Omega}
\end{aligned}$$

That is, the zero is the *uniform matrix.* The dimensions of $\boldsymbol{\Omega}$ must be chosen to be consistent with the vector product.

**Lemma.** There is an *identity* for the vector product. If $\boldsymbol{A} \in {}^{m}\mathbb{S}^{n}$ then

$$\begin{aligned}
\boldsymbol{\Xi} \otimes \boldsymbol{A} &= \boldsymbol{A} \\
\boldsymbol{A} \otimes \boldsymbol{\Xi} &= \boldsymbol{A}
\end{aligned}$$

That is, the identity is the *exponential identity.* The dimensions of $\boldsymbol{\Xi}$ must be chosen to be consistent with the vector product.

**Definition.** VECTOR PRODUCT INVERSE: There is an *inverse* associated with the vector product. If $\boldsymbol{A} \in {}^{m}\mathbb{S}^{m}$ then

$$\begin{aligned}
\boldsymbol{A} \otimes \boldsymbol{A}^{-1} &= \boldsymbol{\Xi} \\
\boldsymbol{A}^{-1} \otimes \boldsymbol{A} &= \boldsymbol{\Xi}
\end{aligned}$$

where $\boldsymbol{A}^{-1}$ is the *vector product inverse* of $\boldsymbol{A}$. The dimensions of $\boldsymbol{\Xi}$ must be chosen to be consistent with the vector product. Note, the inverse symbol is the same as that for matrix algebra. It will be clear from context which inverse to use. Specifically, the inverse of any stochastic matrix will always use a stochastic inverse. A formula for the inverse will be given in a later section.

## 2.6 An Outer Product

It will be extremely useful to define an *outer product.*

**Definition.** OUTER PRODUCT: Let $\boldsymbol{x} = [x_i] \in {}^n\mathbb{S}, \boldsymbol{y} = [y_i] \in {}^m\mathbb{S}$. The *outer product* of $\boldsymbol{x}$ and $\boldsymbol{y}$, denoted $\boldsymbol{x} \rangle\langle \boldsymbol{y}$, is

$$\boldsymbol{x} \rangle\langle \boldsymbol{y} = \downarrow\left[e^{\ln y_i \ln x_j}\right]$$

where we note that $\boldsymbol{x} \rangle\langle \boldsymbol{y} \in {}^m\mathbb{S}^n$. The outer product can be generalized to stochastic matrices (not just columns). Let $\boldsymbol{A} = [a_{ij}] \in {}^n\mathbb{S}^p$, $\boldsymbol{B} = [b_{ij}] \in {}^m\mathbb{S}^p$.

$$\boldsymbol{A} \rangle\langle \boldsymbol{B} = \bigoplus_{k=1}^{p} \left[e^{\ln b_{ik} \ln a_{jk}}\right] = \left[\prod_{k=1}^{p} e^{\ln b_{ik} \ln a_{jk}}\right]$$

where we note that $\boldsymbol{A} \rangle\langle \boldsymbol{B} \in {}^m\mathbb{S}^n$. The outer product can be further generalized to include a *real* middle matrix. Let $\mathsf{C} = [c_{ij}] \in {}^p\mathbb{R}^p$.

$$\boldsymbol{A} \rangle\, \mathsf{C} \,\langle \boldsymbol{B} = \bigoplus_{k,l=1}^{p} \left[e^{c_{kl} \ln b_{il} \ln a_{jk}}\right] = \left[\prod_{k,l=1}^{p} e^{c_{kl} \ln b_{il} \ln a_{jk}}\right]$$

The outer product will become extremely useful when defining projections onto a subspace. A series of identities ensues. Let $\boldsymbol{v} = [v_i]$, $\boldsymbol{x} = [x_i] \in {}^n\mathbb{S}$ and $\boldsymbol{u} = [u_i]$, $\boldsymbol{y} = [y_i]$, $\boldsymbol{z} = [z_i] \in {}^m\mathbb{S}$. Let $\lambda, \mu \in \mathbb{R}$.

**OI.** $(\boldsymbol{x} \rangle\langle \boldsymbol{y}) \otimes \boldsymbol{z} \equiv \langle \boldsymbol{y}, \boldsymbol{z} \rangle \cdot \boldsymbol{x}$.

**Proof.**

$$\begin{aligned}
(\boldsymbol{x} \rangle\langle \boldsymbol{y}) \otimes \boldsymbol{z} &= \left[e^{\ln y_i \ln x_j}\right] \otimes \boldsymbol{z} \\
&= \downarrow\left[e^{\left[\ln y_i^{\ln x_j}\right]^{\mathsf{T}} \mathbf{A}[\ln z_k]}\right] \\
&= \downarrow\left[e^{\ln x_j [\ln y_i]^{\mathsf{T}} \mathbf{A}[\ln z_k]}\right] \\
&= \downarrow\left[x_j^{[\ln y_i]^{\mathsf{T}} \mathbf{A}[\ln z_k]}\right] \\
&= \downarrow\left[x_j^{\langle \boldsymbol{y}, \boldsymbol{z} \rangle}\right] \\
&= \langle \boldsymbol{y}, \boldsymbol{z} \rangle \cdot \downarrow[x_j] \\
&= \langle \boldsymbol{y}, \boldsymbol{z} \rangle \cdot \boldsymbol{x}
\end{aligned}$$

■

**OII.** $(\boldsymbol{x} \rangle\langle \boldsymbol{y}) \otimes \boldsymbol{z} \equiv (\boldsymbol{x} \rangle\langle \boldsymbol{z}) \otimes \boldsymbol{y}$.

**Proof.**

$$(\boldsymbol{x} \rangle\langle \boldsymbol{y}) \otimes \boldsymbol{z} = \langle \boldsymbol{y}, \boldsymbol{z} \rangle \cdot \boldsymbol{x} = \langle \boldsymbol{z}, \boldsymbol{y} \rangle \cdot \boldsymbol{x} = (\boldsymbol{x} \rangle\langle \boldsymbol{z}) \otimes \boldsymbol{y}$$

■

**OIII.** $(\lambda \cdot \boldsymbol{x} \rangle\langle \mu \cdot \boldsymbol{y}) \equiv (\lambda\mu) \cdot (\boldsymbol{x} \rangle\langle \boldsymbol{y})$.

**Proof.**

$$\begin{aligned}
(\lambda \cdot \boldsymbol{x} \rangle\langle \mu \cdot \boldsymbol{y}) \otimes \boldsymbol{z} &= \langle \mu \cdot \boldsymbol{y}, \boldsymbol{z} \rangle \cdot (\lambda \cdot \boldsymbol{x}) \\
&= \langle \boldsymbol{y}, (\lambda\mu) \cdot \boldsymbol{z} \rangle \cdot \boldsymbol{x} \\
&= (\boldsymbol{x} \rangle\langle \boldsymbol{y}) \otimes ((\lambda\mu) \cdot \boldsymbol{z}) \\
&= ((\lambda\mu) \cdot (\boldsymbol{x} \rangle\langle \boldsymbol{y})) \otimes \boldsymbol{z}
\end{aligned}$$

■

**QIV.** $(v \oplus x) \rangle\langle (u \oplus y) \equiv (v \rangle\langle u) \oplus (v \rangle\langle y) \oplus (x \rangle\langle u) \oplus (x \rangle\langle y)$.

**Proof.**
$$\begin{aligned}(v \oplus x) \rangle\langle (u \oplus y) \otimes z &= \langle u \oplus y, z\rangle \cdot (v \oplus x) \\ &= \langle u, z\rangle \cdot v \oplus \langle u, z\rangle \cdot x \oplus \langle y, z\rangle \cdot v \oplus \langle y, z\rangle \cdot x \\ &= ((v \rangle\langle u) \oplus (v \rangle\langle y) \oplus (x \rangle\langle u) \oplus (x \rangle\langle y)) \otimes z\end{aligned}$$
■

## 2.7 Subspaces

We have established that the set of stochastic matrices[8], ${}^m\mathbb{S}$, is a vector space of dimension $m-1$. We saw earlier that we may express every vector in ${}^m\mathbb{S}$ as a unique linear combination of $m-1$ basis vectors. We did not address the issue of what happens when only $k$ basis vectors are available, where $0 \leq k < m$. The result is a *subspace* of ${}^m\mathbb{S}$, of dimension $k$.

Let $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\}$ where $\boldsymbol{b}_i \in {}^m\mathbb{S}\ \forall i = 1 \ldots k$ be a set of linearly independent vectors. Define the set $\mathcal{B}$ as

$$\mathcal{B} = \left\{ x \in {}^m\mathbb{S} \;\middle|\; x = \bigoplus_{i=1}^{k} \lambda_i \cdot \boldsymbol{b}_i,\ \lambda_i \in \mathbb{R}\ \forall i = 1 \ldots k \right\}$$

**Proposition.** *The set $\mathcal{B}$ as defined above is a subspace of ${}^m\mathbb{S}$.*

**Proof.** There are three axioms which must be satisfied. Let $x = \bigoplus_{i=1}^{k} \lambda_i \cdot \boldsymbol{b}_i$, $y = \bigoplus_{i=1}^{k} \mu_i \cdot \boldsymbol{b}_i \in \mathcal{B}$ where $\lambda_i, \mu_i \in \mathbb{R}\ \forall i = 1 \ldots k$. Let $\gamma \in \mathbb{R}$.

**SI.** INCLUSION OF ZERO VECTOR: $\omega \in \mathcal{B}$.

$$\omega = \bigoplus_{i=1}^{k} (0) \cdot \boldsymbol{b}_i \ \in \mathcal{B}$$

**SII.** CLOSURE UNDER VECTOR ADDITION: $x \oplus y \in \mathcal{B}$.

$$x \oplus y = \left( \bigoplus_{i=1}^{k} \lambda_i \cdot \boldsymbol{b}_i \right) \oplus \left( \bigoplus_{i=1}^{k} \mu_i \cdot \boldsymbol{b}_i \right) = \bigoplus_{i=1}^{k} (\lambda_i + \mu_i) \cdot \boldsymbol{b}_i \ \in \mathcal{B}$$

**SIII.** CLOSURE UNDER SCALAR MULTIPLICATION: $\gamma \cdot x \in \mathcal{B}$.

$$\gamma \cdot x = \gamma \cdot \left( \bigoplus_{i=1}^{k} \lambda_i \cdot \boldsymbol{b}_i \right) = \bigoplus_{i=1}^{k} (\gamma \lambda_i) \cdot \boldsymbol{b}_i \ \in \mathcal{B}$$

Therefore, a subspace. ■

We can also speak of a subspace as the *span* of basis vectors such that $\mathcal{B} = \mathrm{sp}\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\}$.

[8] Subspaces are discussed for stochastic columns only but could easily be generalized to stochastic matrices.

## 2.8 Projections

There are several important implications of working with an inner product space. The first is that we may immediately discuss the concept of a *projection.*

**Definition.** PROJECTION: The *projection* of vector $\boldsymbol{x} \in {}^{m}\mathbb{S}$ onto non-zero vector $\boldsymbol{p} \in {}^{m}\mathbb{S}$ is

$$\underset{\rightarrow \boldsymbol{p}}{\text{proj}}\, \boldsymbol{x} = \frac{\langle \boldsymbol{x}, \boldsymbol{p} \rangle}{\langle \boldsymbol{p}, \boldsymbol{p} \rangle} \boldsymbol{p}$$

We may also define the *orthogonal complement* of a projection as follows.

**Definition.** ORTHOGONAL COMPLEMENT: Let $\boldsymbol{x}, \boldsymbol{p} \in {}^{m}\mathbb{S}$. Let $\boldsymbol{y} = \underset{\rightarrow \boldsymbol{p}}{\text{proj}}\, \boldsymbol{x} \in {}^{m}\mathbb{S}^{n}$ be the projection of $\boldsymbol{x}$ onto $\boldsymbol{p}$. The *orthogonal complement* of vector $\boldsymbol{y}$, denoted $\boldsymbol{y}^{\perp}$ is

$$\boldsymbol{y}^{\perp} = \boldsymbol{x} \ominus \boldsymbol{y}$$

Notice that

$$\boldsymbol{y} \oplus \boldsymbol{y}^{\perp} = \boldsymbol{y} \oplus (\boldsymbol{x} \ominus \boldsymbol{y}) = \boldsymbol{x}$$
$$\langle \boldsymbol{y}, \boldsymbol{y}^{\perp} \rangle = 0$$

Rather than project a vector onto another vector, we can also project it onto a subspace. This will require a *projection matrix.*

**Definition.** PROJECTION MATRIX: Let $\mathcal{B} = \text{sp}\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\}$ where $\boldsymbol{b}_i \in {}^{m}\mathbb{S}\ \forall i = 1 \ldots k$ be a subspace of ${}^{m}\mathbb{S}$. Let $\boldsymbol{B} = [\boldsymbol{b}_1 \cdots \boldsymbol{b}_k] \in {}^{m}\mathbb{S}^{k}$ have as columns the basis vectors of $\mathcal{B}$. The *projection matrix*, $\boldsymbol{P} \in {}^{m}\mathbb{S}^{m}$, associated with $\mathcal{B}$ is

$$\boldsymbol{P} = \boldsymbol{B} \rangle \langle \boldsymbol{B}, \boldsymbol{B} \rangle^{-1} \langle \boldsymbol{B}$$

where $\langle \boldsymbol{B}, \boldsymbol{B} \rangle^{-1} = [\langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle]^{-1} \in {}^{k}\mathbb{R}^{k}$.

**Proposition.** *The projection of vector* $\boldsymbol{x} \in {}^{m}\mathbb{S}$ *on subspace* $\mathcal{B}$ *(as defined above), denoted* $\underset{\rightarrow \mathcal{B}}{\text{proj}}\, \boldsymbol{x}$, *is*

$$\underset{\rightarrow \mathcal{B}}{\text{proj}}\, \boldsymbol{x} = \boldsymbol{P} \otimes \boldsymbol{x}$$

*where* $\boldsymbol{P}$ *is the projection matrix associated with* $\mathcal{B}$.

**Proof.** Using matrices defined above. Let $\boldsymbol{y}$ be the projection of $\boldsymbol{x}$ onto $\mathcal{B}$. Then

$$\boldsymbol{y} = \underset{\rightarrow \mathcal{B}}{\text{proj}}\, \boldsymbol{x} = \bigoplus_{i=1}^{k} \gamma_i \cdot \boldsymbol{b}_i$$

where the $\gamma_i\ \forall i = 1 \ldots k$ are the coordinates of $\boldsymbol{y}$. We select another arbitrary vector, $\boldsymbol{v} \in \mathcal{B}$, such that

$$\boldsymbol{v} = \bigoplus_{i=1}^{k} \nu_i \cdot \boldsymbol{b}_i$$

where the $\nu_i \; \forall i = 1 \ldots k$ are the coordinates of $\boldsymbol{v}$. The orthogonal complement of $\boldsymbol{y}$ must be perpendicular to $\boldsymbol{v}$ such that

$$\begin{aligned} \langle \boldsymbol{v}, \boldsymbol{x} \ominus \boldsymbol{y} \rangle &= 0 \\ \langle \bigoplus_{i=1}^{k} \nu_i \cdot \boldsymbol{b}_i, \boldsymbol{x} \ominus \boldsymbol{y} \rangle &= 0 \\ \sum_{i=1}^{k} \nu_i \langle \boldsymbol{b}_i, \boldsymbol{x} \ominus \boldsymbol{y} \rangle &= 0 \end{aligned}$$

Since $\boldsymbol{v}$ was selected arbitrarily must have $\forall i = 1 \ldots k$ that

$$\begin{aligned} \langle \boldsymbol{b}_i, \boldsymbol{x} \ominus \boldsymbol{y} \rangle &= 0 \\ \langle \boldsymbol{b}_i, \boldsymbol{x} \rangle &= \langle \boldsymbol{b}_i, \boldsymbol{y} \rangle \\ \langle \boldsymbol{b}_i, \boldsymbol{x} \rangle &= \langle \boldsymbol{b}_i, \bigoplus_{j=1}^{k} \gamma_j \cdot \boldsymbol{b}_j \rangle \\ \langle \boldsymbol{b}_i, \boldsymbol{x} \rangle &= \sum_{j=1}^{k} \gamma_j \langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle \end{aligned}$$

In matrix form this becomes

$$\begin{aligned} [\langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle][\gamma_i] &= [\langle \boldsymbol{b}_i, \boldsymbol{x} \rangle] \\ [\gamma_i] &= [\langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle]^{-1} [\langle \boldsymbol{b}_i, \boldsymbol{x} \rangle] \\ [\gamma_i] &= \mathsf{A} [\langle \boldsymbol{b}_i, \boldsymbol{x} \rangle] \end{aligned}$$

where $\mathsf{A} = [\langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle]^{-1} = \langle \boldsymbol{B}, \boldsymbol{B} \rangle^{-1} = [a_{ij}] \in {}^{k}\mathbb{R}^{k}$. Then we have

$$\begin{aligned} \gamma_i \cdot \boldsymbol{b}_i &= \left( \sum_{j=1}^{k} a_{ij} \langle \boldsymbol{b}_j, \boldsymbol{x} \rangle \right) \cdot \boldsymbol{b}_i \\ &= \bigoplus_{j=1}^{k} (\langle a_{ij} \cdot \boldsymbol{b}_j, \boldsymbol{x} \rangle \cdot \boldsymbol{b}_i) \\ &= \bigoplus_{j=1}^{k} ((a_{ij} \cdot \boldsymbol{b}_j) \; \rangle\langle \; \boldsymbol{b}_i \otimes \boldsymbol{x}) \\ &= \left( \bigoplus_{j=1}^{k} (a_{ij} \cdot \boldsymbol{b}_j) \; \rangle\langle \; \boldsymbol{b}_i \right) \otimes \boldsymbol{x} \\ &= \left( \bigoplus_{j=1}^{k} \left[ e^{a_{ij} \ln b_{pi} \ln b_{qj}} \right] \right) \otimes \boldsymbol{x} \end{aligned}$$

Summing over all the basis vectors we arrive at the desired projection

$$\begin{aligned} \boldsymbol{y} &= \bigoplus_{i=1}^{k} \gamma_i \cdot \boldsymbol{b}_i \\ &= \left( \bigoplus_{i,j=1}^{k} \left[ e^{a_{ij} \ln b_{pi} \ln b_{qj}} \right] \right) \otimes \boldsymbol{x}, \; \forall p, \; q = 1 \ldots m \\ &= (\boldsymbol{B} \; \rangle \; \mathsf{A} \; \langle \; \boldsymbol{B}) \otimes \boldsymbol{x} \\ &= (\boldsymbol{B} \; \rangle \; \langle \boldsymbol{B}, \boldsymbol{B} \rangle^{-1} \langle \; \boldsymbol{B}) \otimes \boldsymbol{x} \\ &= \boldsymbol{P} \otimes \boldsymbol{x} \end{aligned}$$

concluding the proof. ■

**Example.** Let $\mathcal{B} = \text{sp}\{\boldsymbol{b}\}$ where $\boldsymbol{b} \in {}^{m}\mathbf{S}$. The projection matrix, $\boldsymbol{P}$, associated with $\mathcal{B}$ is

$$\boldsymbol{P} = \frac{\boldsymbol{b}\,\rangle\langle\,\boldsymbol{b}}{\langle \boldsymbol{b}, \boldsymbol{b}\rangle}$$

The projection of $\boldsymbol{x}$ on $\mathcal{B}$, denoted $\boldsymbol{y} = \underset{\rightarrow \boldsymbol{b}}{\text{proj}}\, \boldsymbol{x}$, is

$$\boldsymbol{y} = \boldsymbol{P} \otimes \boldsymbol{x} = \frac{\boldsymbol{b}\,\rangle\langle\,\boldsymbol{b}}{\langle \boldsymbol{b}, \boldsymbol{b}\rangle} \otimes \boldsymbol{x} = \frac{\langle \boldsymbol{b}, \boldsymbol{x}\rangle}{\langle \boldsymbol{b}, \boldsymbol{b}\rangle} \cdot \boldsymbol{b}$$

which is the familiar formula for projection onto a vector. □

In the special case that the basis vectors are orthonormal, we have that

$$\langle \boldsymbol{B}, \boldsymbol{B}\rangle = \langle \boldsymbol{B}, \boldsymbol{B}\rangle^{-1} = \mathbf{1}$$

so that

$$\begin{aligned} \boldsymbol{P} &= \boldsymbol{B}\,\rangle\,\mathbf{1}\,\langle\,\boldsymbol{B} \\ &= \boldsymbol{B}\,\rangle\langle\,\boldsymbol{B} \\ &= \bigoplus_{i=1}^{k} (\boldsymbol{b}_i\,\rangle\langle\,\boldsymbol{b}_i) \end{aligned}$$

**Example.** Below are two subspaces and the projection matrices associated with them.

$$\text{sp}\left\{ \downarrow \begin{bmatrix} e \\ e \\ 1 \\ 1 \end{bmatrix} \right\} \rightarrow \frac{1}{2} \cdot \begin{bmatrix} e & e & 1 & 1 \\ e & e & 1 & 1 \\ 1 & 1 & e & e \\ 1 & 1 & e & e \end{bmatrix}$$

$$\text{sp}\left\{ \downarrow \begin{bmatrix} e \\ 1 \\ e \\ 1 \end{bmatrix}, \downarrow \begin{bmatrix} e \\ 1 \\ 1 \\ e \end{bmatrix} \right\} \rightarrow \frac{1}{2} \cdot \begin{bmatrix} e^2 & 1 & e & e \\ 1 & e^2 & e & e \\ e & e & e^2 & 1 \\ e & e & 1 & e^2 \end{bmatrix}$$

The vector addition of these two projection matrices results in $\boldsymbol{\Xi}$, the vector product identity. The reason is that the basis vectors used are orthonormal and thus the projections are orthogonal. □

## 2.9 An Isomorphism

It should not come as a great surprise that there is an *isomorphism*[9] between stochastic algebra and matrix algebra, after all they are both associate algebras.

**Proposition.** *There is an isomorphism,* $\varphi : {}^{m-1}\mathbb{R}^{n-1} \mapsto {}^{m}\mathbb{S}^{n}$, *between stochastic algebra and matrix algebra given by*

$$\varphi : {}^{m-1}\mathbb{R}^{n-1} \mapsto {}^{m}\mathbb{S}^{n} \qquad \boldsymbol{A} = \varphi(\mathsf{A}) = \boldsymbol{\Phi} \rangle\, \mathsf{A} \,\langle \boldsymbol{\Psi}$$
$$\varphi^{-1} : {}^{m}\mathbb{S}^{n} \mapsto {}^{m-1}\mathbb{R}^{n-1} \qquad \mathsf{A} = \varphi^{-1}(\boldsymbol{A}) = \left[\langle \boldsymbol{\phi}_i, \boldsymbol{A} \otimes \boldsymbol{\psi}_j \rangle\right]$$

*where* $\boldsymbol{A} \in {}^{m}\mathbb{S}^{n}$ *and* $\mathsf{A} \in {}^{m-1}\mathbb{R}^{n-1}$. *Also,* $\boldsymbol{\Phi} = \left[\, \boldsymbol{\phi}_1 \;\cdots\; \boldsymbol{\phi}_{m-1} \,\right] \in {}^{m}\mathbb{S}^{m-1}$ *where the* $\boldsymbol{\phi}_i \in {}^{m}\mathbb{S}\ \forall i = 1 \ldots (m-1)$ *form an orthonormal basis for* ${}^{m}\mathbb{S}$. *Similarly,* $\boldsymbol{\Psi} = \left[\, \boldsymbol{\psi}_1 \;\cdots\; \boldsymbol{\psi}_{n-1} \,\right] \in {}^{n}\mathbb{S}^{n-1}$ *where the* $\boldsymbol{\psi}_j \in {}^{n}\mathbb{S}\ \forall j = 1 \ldots (n-1)$ *form an orthonormal basis for* ${}^{n}\mathbb{S}$.

**Proof.** There are 4 axioms which must be satisfied. Let $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$, $\boldsymbol{A}$, $\mathsf{A} = [a_{ij}]$ be defined as above. Let $\lambda \in \mathbb{R}$.

**RI.** SCALAR MULTIPLICATION. $\varphi(\lambda\mathsf{A}) = \lambda \cdot \varphi(\mathsf{A})$.

$$\begin{aligned}
\varphi(\lambda\mathsf{A}) &= \boldsymbol{\Phi} \rangle\, (\lambda\mathsf{A}) \,\langle \boldsymbol{\Psi} \\
&= \bigoplus_{i,j=1}^{m-1,n-1} (\lambda a_{ij}) \cdot \boldsymbol{\phi}_i \,\rangle\langle\, \boldsymbol{\psi}_j \\
&= \lambda \cdot \bigoplus_{i,j=1}^{m-1,n-1} a_{ij} \cdot \boldsymbol{\phi}_i \,\rangle\langle\, \boldsymbol{\psi}_j \\
&= \lambda \cdot \boldsymbol{\Phi} \rangle\, \mathsf{A} \,\langle \boldsymbol{\Psi} \\
&= \lambda \cdot \varphi(\mathsf{A})
\end{aligned}$$

**RII.** ADDITION. $\varphi(\mathsf{A} + \mathsf{B}) = \varphi(\mathsf{A}) \oplus \varphi(\mathsf{B})$.

Let $\mathsf{B} = [b_{ij}] \in {}^{m-1}\mathbb{R}^{n-1}$.

$$\begin{aligned}
\varphi(\mathsf{A} + \mathsf{B}) &= \boldsymbol{\Phi} \rangle\, (\mathsf{A} + \mathsf{B}) \,\langle \boldsymbol{\Psi} \\
&= \bigoplus_{i,j=1}^{m-1,n-1} (a_{ij} + b_{ij}) \cdot \boldsymbol{\phi}_i \,\rangle\langle\, \boldsymbol{\psi}_j \\
&= \bigoplus_{i,j=1}^{m-1,n-1} a_{ij} \cdot \boldsymbol{\phi}_i \,\rangle\langle\, \boldsymbol{\psi}_j \oplus \bigoplus_{i,j=1}^{m-1,n-1} b_{ij} \cdot \boldsymbol{\phi}_i \,\rangle\langle\, \boldsymbol{\psi}_j \\
&= \boldsymbol{\Phi} \rangle\, \mathsf{A} \,\langle \boldsymbol{\Psi} \oplus \boldsymbol{\Phi} \rangle\, \mathsf{B} \,\langle \boldsymbol{\Psi} \\
&= \varphi(\mathsf{A}) \oplus \varphi(\mathsf{B})
\end{aligned}$$

[9] Reminder: An isomorphism is a bijective homomorphism. A homomorphism is a linear map that preserves products. A map that is both surjective and injective is called bijective.

**RIII.** BIJECTIVITY. $\varphi^{-1}(A) = [\langle \phi_i, A \otimes \psi_j \rangle]$

To show that $\varphi$ is bijective we show that $\varphi^{-1}$ is the function above.

$$A = \Phi \rangle\, \mathsf{A}\, \langle \Psi = \bigoplus_{k,l=1}^{m-1,n-1} a_{kl} \cdot \phi_k \rangle\langle\, \psi_l$$

$$A \otimes \psi_j = \bigoplus_{k,l=1}^{m-1,n-1} a_{kl} \cdot \phi_k \rangle\langle\, \psi_l \otimes \psi_j = \bigoplus_{k,l=1}^{m-1,n-1} a_{kl} \cdot \langle \psi_l, \psi_j \rangle \cdot \phi_k = \bigoplus_{k=1}^{m-1} a_{kj} \cdot \phi_k$$

$$\langle \phi_i, A \otimes \psi_j \rangle = \langle \phi_i, \bigoplus_{k=1}^{m-1} a_{kj} \cdot \phi_k \rangle = \bigoplus_{k=1}^{m-1} a_{kj} \langle \phi_i, \phi_k \rangle = a_{ij}$$

$$\mathsf{A} = [a_{ij}] = \left[ \langle \phi_i, A \otimes \psi_j \rangle \right]$$

where we have used that $\langle \phi_i, \phi_j \rangle = \delta_{ij}$, $\langle \psi_i, \psi_j \rangle = \delta_{ij}$ since the bases we choose are orthonormal.

**RIV.** VECTOR PRODUCT. $\varphi(\mathsf{AB}) = \varphi(\mathsf{A}) \otimes \varphi(\mathsf{B})$

Let $B = \varphi(\mathsf{B}) = \Psi \rangle\, \mathsf{B}\, \langle\, \Gamma$ where $B \in {}^{n}\mathbf{S}^{p}$, $\mathsf{B} = [b_{ij}] \in {}^{n-1}\mathbb{R}^{p-1}$ and $\Gamma = [\, \gamma_1 \ \cdots \ \gamma_{p-1} \,] \in {}^{p}\mathbf{S}^{p-1}$ where the $\gamma_k \in {}^{p}\mathbf{S}\ \forall k = 1 \ldots (p-1)$ form an orthonormal basis for ${}^{p}\mathbf{S}$. Then

$$\begin{aligned}
A \otimes B &= \left( \Phi \rangle\, \mathsf{A}\, \langle \Psi \right) \otimes \left( \Psi \rangle\, \mathsf{B}\, \langle\, \Gamma \right) \\
&= \left( \bigoplus_{k,l=1}^{m-1,n-1} a_{kl} \cdot \phi_k \rangle\langle\, \psi_l \right) \otimes \left( \bigoplus_{q,r=1}^{n-1,p-1} b_{qr} \cdot \psi_q \rangle\langle\, \gamma_r \right) \\
&= \bigoplus_{k,l,q,r} (a_{kl} b_{qr}) \cdot (\phi_k \rangle\langle\, \psi_l) \otimes (\psi_q \rangle\langle\, \gamma_r) \\
A \otimes B \otimes \gamma_j &= \bigoplus_{k,l,q,r} (a_{kl} b_{qr}) \cdot (\phi_k \rangle\langle\, \psi_l) \otimes (\psi_q \rangle\langle\, \gamma_r) \otimes \gamma_j \\
&= \bigoplus_{k,l,q,r} (a_{kl} b_{qr}) \cdot \phi_k \rangle\langle\, \psi_l \otimes (\langle \gamma_r, \gamma_j \rangle \cdot \psi_q) \\
&= \bigoplus_{k,l,q} (a_{kl} b_{qj}) \cdot \langle \psi_l, \psi_q \rangle \cdot \phi_k \\
\langle \phi_i, A \otimes B \otimes \gamma_j \rangle &= \langle \phi_i, \bigoplus_{k,l} (a_{kl} b_{lj}) \cdot \langle \psi_l, \psi_q \rangle \cdot \phi_k \rangle \\
&= \sum_{l=1}^{n-1} a_{il} b_{lj} \\
\mathsf{AB} &= \left[ \sum_{l=1}^{n-1} a_{il} b_{lj} \right] \\
&= \left[ \langle \phi_i, A \otimes B \otimes \gamma_j \rangle \right] \\
\varphi(\mathsf{AB}) &= A \otimes B \\
&= \varphi(\mathsf{A}) \otimes \varphi(\mathsf{B})
\end{aligned}$$

where we have used that $\langle \phi_i, \phi_j \rangle = \delta_{ij}$, $\langle \psi_i, \psi_j \rangle = \delta_{ij}$, $\langle \gamma_i, \gamma_j \rangle = \delta_{ij}$ since the bases we choose are orthonormal.

Therefore an isomorphism. ■

## 2.10 Determinant, Rank, Inverse

We have shown that the vector product defined above is a linear transformation of the $m-1$ dimensional space, ${}^m\mathbb{S}$. To define the determinant of a stochastic matrix we follow Greub [1974] and select a non-trivial determinant function, $\Delta : \underbrace{{}^m\mathbb{S} \times \cdots \times {}^m\mathbb{S}}_{m-1} \mapsto \mathbb{R}$, which has the properties

$$\begin{aligned}
\Delta(\boldsymbol{b}_1,\ldots,\lambda\cdot\boldsymbol{b}_i,\ldots,\boldsymbol{b}_{m-1}) &= \lambda\Delta(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_i,\ldots,\boldsymbol{b}_{m-1}) \\
\Delta(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_i \oplus \boldsymbol{x},\ldots,\boldsymbol{b}_{m-1}) &= \Delta(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_i,\ldots,\boldsymbol{b}_{m-1}) + \Delta(\boldsymbol{b}_1,\ldots,\boldsymbol{x},\ldots,\boldsymbol{b}_{m-1}) \\
\Delta(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_i,\ldots,\boldsymbol{b}_j,\ldots,\boldsymbol{b}_{m-1}) &= -\Delta(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_j,\ldots,\boldsymbol{b}_i,\ldots,\boldsymbol{b}_{m-1})
\end{aligned}$$

where $\boldsymbol{x}$, $\boldsymbol{b}_i \in {}^m\mathbb{S}\ \forall i = 1\ldots(m-1)$ and $\lambda \in \mathbb{R}$.

**Definition.** DETERMINANT: Let $\boldsymbol{b}_i \in {}^m\mathbb{S}\ \forall i = 1\ldots(m-1)$ be a basis for ${}^m\mathbb{S}$. The *determinant* of a stochastic matrix is defined as

$$\det \boldsymbol{A} = \frac{\Delta(\boldsymbol{A}\otimes\boldsymbol{b}_1,\ldots,\boldsymbol{A}\otimes\boldsymbol{b}_i,\ldots,\boldsymbol{A}\otimes\boldsymbol{b}_{m-1})}{\Delta(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_i,\ldots,\boldsymbol{b}_{m-1})}$$

where $\boldsymbol{A} \in {}^m\mathbb{S}^m$.

Using the properties of the determinant function defined above it is possible to come up with a more practical formula for the determinant.

**Proposition.** *The determinant may be expressed as*

$$\det \boldsymbol{A} = \left|\left[\langle \boldsymbol{a}_{\mathrm{r},i}^{\mathrm{T}} \ominus \boldsymbol{a}_{\mathrm{r},m}^{\mathrm{T}}, \boldsymbol{\xi}_j\rangle\right]\right|$$

*where $\boldsymbol{a}_{\mathrm{r},i}\ \forall i = 1\ldots m$ represents the $i^{th}$ row of $\boldsymbol{A}$, $\boldsymbol{\xi}_j\ \forall j = 1\ldots(m-1)$ is the $j^{th}$ column of $\Xi$, and $|\cdot|$ is the familiar formula for the determinant of a matrix in ${}^{(m-1)}\mathbb{R}^{(m-1)}$. Note, $\boldsymbol{A} \in {}^m\mathbb{S}^m$.*

**Proof.** Using the definition of the determinant above we select $\boldsymbol{b}_i = \boldsymbol{\xi}_i\ \forall i = 1\ldots(m-1)$ which was shown to be a basis for ${}^m\mathbb{S}$ previously[10]. If we then expand each of the products, $\boldsymbol{A}\otimes\boldsymbol{\xi}_j$, as

$$\boldsymbol{A}\otimes\boldsymbol{\xi}_j = \downarrow\left[e^{\langle \boldsymbol{a}_{\mathrm{r},i}^{\mathrm{T}}, \boldsymbol{\xi}_j\rangle}\right] = \bigoplus_{k=1}^{m}\langle \boldsymbol{a}_{\mathrm{r},i}^{\mathrm{T}}, \boldsymbol{\xi}_j\rangle\cdot\boldsymbol{\xi}_k = \bigoplus_{k=1}^{m-1}(\langle \boldsymbol{a}_{\mathrm{r},i}^{\mathrm{T}}, \boldsymbol{\xi}_j\rangle - \langle \boldsymbol{a}_{\mathrm{r},m}^{\mathrm{T}}, \boldsymbol{\xi}_j\rangle)\cdot\boldsymbol{\xi}_k$$

we may substitute these into the definition of the determinant

$$\begin{aligned}
\det \boldsymbol{A} &= \frac{\Delta(\boldsymbol{A}\otimes\boldsymbol{\xi}_1,\ldots,\boldsymbol{A}\otimes\boldsymbol{\xi}_i,\ldots,\boldsymbol{A}\otimes\boldsymbol{\xi}_{m-1})}{\Delta(\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_i,\ldots,\boldsymbol{\xi}_{m-1})} \\
&\vdots \\
&= \left|\left[\langle \boldsymbol{a}_{\mathrm{r},i}^{\mathrm{T}}, \boldsymbol{\xi}_j\rangle - \langle \boldsymbol{a}_{\mathrm{r},m}^{\mathrm{T}}, \boldsymbol{\xi}_j\rangle\right]\right| \frac{\Delta(\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_i,\ldots,\boldsymbol{\xi}_{m-1})}{\Delta(\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_i,\ldots,\boldsymbol{\xi}_{m-1})} \\
&= \left|\left[\langle \boldsymbol{a}_{\mathrm{r},i}^{\mathrm{T}} \ominus \boldsymbol{a}_{\mathrm{r},m}^{\mathrm{T}}, \boldsymbol{\xi}_j\rangle\right]\right|
\end{aligned}$$

where we have used the properties of determinant functions mentioned at the beginning of the section. ■

[10] We could select any $(m-1)$ columns of $\Xi$ and it would work out exactly the same.

There are three more useful connections to be made as a result of the isomorphism, $\varphi$, between matrix algebra and stochastic algebra.

**Proposition.** *The determinant of the vector product (with a square stochastic matrix, $\boldsymbol{A} \in {}^{m}\mathbb{S}^{m}$) may be expressed as*

$$\det \boldsymbol{A} = |\mathsf{A}|$$

*where* $\mathsf{A} = \varphi^{-1}(\boldsymbol{A}) \in {}^{m-1}\mathbb{R}^{m-1}$.

**Proof.** The proof is identical to the previous one except we expand using an orthonormal basis instead of $m-1$ columns of $\boldsymbol{\Xi}$. ■

**Definition.** RANK: Let $\boldsymbol{A} \in {}^{m}\mathbb{S}^{n}$. The *rank* of $\boldsymbol{A}$, denoted $\operatorname{rank}\boldsymbol{A}$ is

$$\operatorname{rank}\boldsymbol{A} = \operatorname{rank}\mathsf{A}$$

where $\mathsf{A} = \varphi^{-1}(\boldsymbol{A}) \in {}^{m-1}\mathbb{R}^{n-1}$.

**Definition.** INVERSE: Let $\boldsymbol{A} \in {}^{m}\mathbb{S}^{m}$. The *inverse* of $\boldsymbol{A}$, denoted $\boldsymbol{A}^{-1}$ is

$$\boldsymbol{A}^{-1} = \varphi\left(\mathsf{A}^{-1}\right)$$

where $\mathsf{A} = \varphi^{-1}(\boldsymbol{A}) \in {}^{m-1}\mathbb{R}^{n-1}$. Note, the inverse only exists when $\det\boldsymbol{A} \neq 0$ since $\mathsf{A}^{-1}$ only exists when $|\mathsf{A}| = \det\boldsymbol{A} \neq 0$.

## 2.11 The Eigen Problem

We are now prepared to tackle the eigen problem.

**Definition.** EIGENVALUE: Let $\boldsymbol{A} \in {}^{m}\mathbb{S}^{m}$. A scalar, $\lambda$, is an *eigenvalue* of $\boldsymbol{A}$ if there is a nonzero ($\neq \boldsymbol{\omega}$) stochastic column, $\boldsymbol{x} \in {}^{m}\mathbb{S}$, such that

$$\boldsymbol{A} \otimes \boldsymbol{x} = \lambda \cdot \boldsymbol{x}$$

The stochastic column, $\boldsymbol{x}$, is then an *eigenvector* of $\boldsymbol{A}$, corresponding to $\lambda$.

If we rewrite the above equation as

$$\left(\lambda \cdot \boldsymbol{\Xi} \ominus \boldsymbol{A}\right) \otimes \boldsymbol{x} = \boldsymbol{\Omega}$$

then $\boldsymbol{x}$ must be a solution of this homogeneous linear system. This system has a nontrivial ($\boldsymbol{x} \neq \boldsymbol{\omega}$) solution when the determinant of the coefficient matrix is zero

$$p(\lambda) = \det\left(\lambda \cdot \boldsymbol{\Xi} \ominus \boldsymbol{A}\right) = 0$$

where the polynomial, $p(\lambda)$ (degree $(m-1)$) is called the characteristic polynomial of $\boldsymbol{A}$. The eigenvalues of $\boldsymbol{A}$ are the solutions of the characteristic equation, $p(\lambda) = 0$. Notice there are only $(m-1)$ eigenvalues because ${}^{m}\mathbb{S}$ is only an $(m-1)$ dimensional vector space.

**Example.** Consider the stochastic matrix

$$\boldsymbol{A} = \downarrow \begin{bmatrix} e^a & 1 \\ 1 & e^b \end{bmatrix}$$

The characteristic polynomial is

$$\begin{aligned} p(\lambda) &= \det\Big(\lambda \cdot \boldsymbol{\Xi} \ominus \boldsymbol{A}\Big) \\ &= \det\left(\downarrow \begin{bmatrix} e^{\lambda-a} & 1 \\ 1 & e^{\lambda-b} \end{bmatrix}\right) \\ &= \frac{1}{2}\Big|(\lambda-a)+(\lambda-b)\Big| \\ &= \lambda - \frac{a+b}{2} \end{aligned}$$

which has the root $\lambda = \frac{a+b}{2}$. □

**Example.** Consider the stochastic matrix

$$\boldsymbol{B} = \downarrow \begin{bmatrix} e^a & 1 & 1 \\ 1 & e^b & 1 \\ 1 & 1 & e^c \end{bmatrix}$$

The characteristic polynomial is

$$\begin{aligned} p(\lambda) &= \det\Big(\lambda \cdot \boldsymbol{\Xi} \ominus \boldsymbol{B}\Big) \\ &= \det\left(\downarrow \begin{bmatrix} e^{\lambda-a} & 1 & 1 \\ 1 & e^{\lambda-b} & 1 \\ 1 & 1 & e^{\lambda-c} \end{bmatrix}\right) \\ &= \frac{1}{9}\begin{vmatrix} 2(\lambda-a)+(\lambda-c) & -(\lambda-a)+(\lambda-c) \\ -(\lambda-b)+(\lambda-c) & 2(\lambda-b)+(\lambda-c) \end{vmatrix} \\ &= \lambda^2 - \frac{2}{3}\lambda(a+b+c) + \frac{1}{3}(ab+ac+bc) \end{aligned}$$

which has roots

$$\lambda = \frac{1}{3}\Big((a+b+c) \pm \Big((a+b+c)^2 - 3(ab+ac+bc)\Big)^{\frac{1}{2}}\Big)$$

Notice there are only $(m-1) = 2$ eigenvalues which sum to $\frac{2}{3}(a+b+c)$. □

We also have access to the Cayley-Hamilton theorem at this point, namely that every square matrix, $\boldsymbol{A} \in {}^{m}\mathbb{S}^{m}$, satisfies its own characteristic equation.

**Example.** In the above examples we have that

$$\boldsymbol{A} \ominus \frac{a+b}{2} \cdot \boldsymbol{\Xi} = \boldsymbol{\Omega}$$

$$\boldsymbol{B}^2 \ominus \frac{2}{3}(a+b+c) \cdot \boldsymbol{B} \oplus \frac{1}{3}(ab+ac+bc) \cdot \boldsymbol{\Xi} = \boldsymbol{\Omega}$$

where $\boldsymbol{B}^2 = \boldsymbol{B} \otimes \boldsymbol{B}$. □

## 2.12 A Calculus

It is not surprising that we can associate with our stochastic algebra a corresponding stochastic calculus. As ${}^m\mathbb{S}^n$ is a vector space, all the typical results from vector calculus may be obtained. It will be useful to go through some of the concepts. The discussion will be limited to stochastic columns but may be easily generalized to stochastic matrices.

Partial differentiation, in particular, is very useful in the computation of the *Jacobian* of a *stochastic function.* A stochastic function may be defined as

**Definition.** STOCHASTIC FUNCTION: The set of *stochastic functions* ${}^m\mathbb{F}^n$ (with one input variable) is

$$ {}^m\mathbb{F}^n : {}^n\mathbb{S} \mapsto {}^m\mathbb{S} = \left\{ \boldsymbol{f}(\boldsymbol{x}) = [f_i(\boldsymbol{x})] \in {}^n\mathbb{S} \mapsto {}^m\mathbb{R} \;\middle|\; \sum_{i=1}^{m} f_i(\boldsymbol{x}) = 1,\ f_i(\boldsymbol{x}) > 0 \,\forall \boldsymbol{x} \in {}^n\mathbb{S} \right\} $$

This may be generalized to multiple input variables. Let $\boldsymbol{x}_k \ \forall k = 1 \ldots K$ be the input variables so that the set of stochastic functions with multiple input variables is

$$ {}^m\mathbb{F}^{n_1,\ldots,n_K} : {}^{n_1}\mathbb{S} \times \cdots \times {}^{n_K}\mathbb{S} \mapsto {}^m\mathbb{S} = \left\{ \boldsymbol{f}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_K) = [f_i(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_K)] \in {}^{n_1}\mathbb{S} \times \cdots \times {}^{n_K}\mathbb{S} \mapsto {}^m\mathbb{R} \right. $$
$$ \left. \;\middle|\; \sum_{i=1}^{m} f_i(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_K) = 1,\ f_i(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_K) > 0 \,\forall \boldsymbol{x}_k \in {}^{n_k}\mathbb{S} \right\} $$

Based on the definition of a derivative we define the partial derivative of a stochastic function as follows.

**Definition.** PARTIAL DERIVATIVE: The *partial derivative* of a stochastic function, $\boldsymbol{f}(\boldsymbol{x}) \in {}^m\mathbb{F}^n$, with respect to the $j^{th}$ element of $\boldsymbol{x} = [x_j] \in {}^n\mathbb{S}$ is

$$ \frac{\partial}{\partial \boldsymbol{x}_j}\boldsymbol{f}(\boldsymbol{x}) = \lim_{\lambda \to 0} \frac{1}{\lambda} \cdot \Big( \boldsymbol{f}(\boldsymbol{x} \oplus \delta\boldsymbol{x}) \ominus \boldsymbol{f}(\boldsymbol{x}) \Big) $$

where $\delta\boldsymbol{x} = \lambda \cdot \boldsymbol{\xi}_j$ and $\boldsymbol{\xi}_j$ is the $j^{th}$ column of ${}^n\Xi^n$.

**Proposition.** *The partial derivative of a stochastic function with respect to the $j^{th}$ element of $\boldsymbol{x}$ is*

$$ \frac{\partial}{\partial \boldsymbol{x}_j}\boldsymbol{f}(\boldsymbol{x}) = \downarrow\left[ e^{\frac{x_j}{f_i(\boldsymbol{x})}\frac{\partial f_i(\boldsymbol{x})}{\partial x_j}} \right] $$

*where $\frac{\partial f_i(\boldsymbol{x})}{\partial x_j}$ indicates the scalar partial differentiation of matrix algebra.*

**Proof.** Let $y = [y_i] = x \oplus \delta x$. We then have

$$\begin{aligned}
\frac{\partial}{\partial x_j} f(x) &= \lim_{\lambda \to 0} \frac{1}{\lambda} \cdot \Big( f(y) \ominus f(x) \Big) \\
&= \lim_{\lambda \to 0} \frac{1}{\lambda} \cdot \left[ \frac{f_i(y)}{f_i(x)} \right] \\
&= \lim_{\lambda \to 0} \downarrow \left[ \left( \frac{f_i(y)}{f_i(x)} \right)^{\frac{1}{\lambda}} \right] \\
&= \downarrow \left[ e^{\lim_{\lambda \to 0} \frac{\ln f_i(y) - \ln f_i(x)}{\lambda}} \right] \\
&= \downarrow \left[ e^{\lim_{\lambda \to 0} \frac{d}{d\lambda} (\ln f_i(y) - \ln f_i(x))} \right] \\
&= \downarrow \left[ e^{\lim_{\lambda \to 0} \frac{1}{f_i(y)} \frac{d}{d\lambda} f_i(y)} \right] \\
&= \downarrow \left[ e^{\frac{1}{f_i(x)} \lim_{\lambda \to 0} \frac{d}{d\lambda} f_i(y)} \right]
\end{aligned}$$

where l'Hôpital's rule has been employed. Recall that $y = x \oplus \delta x = x \oplus \lambda \cdot \xi_j = \downarrow \left[ x_i e^{\lambda \delta_{ij}} \right]$ where $\delta_{ij}$ is the Kroenecker delta. Then we have

$$\begin{aligned}
\lim_{\lambda \to 0} \frac{d}{d\lambda} f_i(y) &= \lim_{\lambda \to 0} \sum_{k=1}^{m} \frac{\partial f_i(y)}{\partial y_k} \frac{\partial y_k}{\partial \lambda} \\
&= \lim_{\lambda \to 0} \sum_{k=1}^{m} \frac{\partial f_i(y)}{\partial y_k} \frac{\partial}{\partial \lambda} \left( x_k e^{\lambda \delta_{kj}} \right) \\
&= \lim_{\lambda \to 0} \sum_{k=1}^{m} \frac{\partial f_i(y)}{\partial y_k} \left( x_k \delta_{kj} e^{\lambda \delta_{kj}} \right) \\
&= \lim_{\lambda \to 0} \frac{\partial f_i(y)}{\partial y_j} \left( x_j e^{\lambda} \right) \\
&= x_j \frac{\partial f_i(x)}{\partial x_j}
\end{aligned}$$

whereupon

$$\begin{aligned}
\frac{\partial}{\partial x_j} f(x) &= \downarrow \left[ e^{\frac{1}{f_i(y)} \lim_{\lambda \to 0} \frac{d}{d\lambda} f_i(y)} \right] \\
&= \downarrow \left[ e^{\frac{x_j}{f_i(x)} \frac{\partial f_i(x)}{\partial x_j}} \right]
\end{aligned}$$

■

**Lemma.** The partial derivative is a linear operation. Let $\mu, \gamma \in \mathbb{R}$ and $f(x), g(x) \in {}^m\mathbb{F}^n, x \in {}^n\mathbb{S}$. Then

$$\begin{aligned}
\frac{\partial}{\partial x_j} (\mu \cdot f(x) \oplus \gamma \cdot g(x)) &= \lim_{\lambda \to 0} \frac{1}{\lambda} \cdot \Big( (\mu \cdot f(x \oplus \delta x) \oplus \gamma \cdot g(x \oplus \delta x)) \ominus (\mu \cdot f(x) \oplus \gamma \cdot g(x)) \Big) \\
&= \mu \cdot \lim_{\lambda \to 0} \frac{1}{\lambda} \cdot \Big( f(x \oplus \delta x) \ominus f(x) \Big) \oplus \gamma \cdot \lim_{\lambda \to 0} \frac{1}{\lambda} \cdot \Big( g(x \oplus \delta x) \ominus g(x) \Big) \\
&= \mu \cdot \frac{\partial}{\partial x_j} f(x) \oplus \gamma \cdot \frac{\partial}{\partial x_j} g(x)
\end{aligned}$$

Let $A \in {}^m\mathbb{S}^m$. Then

$$\begin{aligned}\frac{\partial}{\partial x_j}(A \otimes f(x)) &= \lim_{\lambda \to 0} \frac{1}{\lambda} \cdot \Big(A \otimes f(x \oplus \delta x) \ominus A \otimes f(x)\Big) \\ &= A \otimes \lim_{\lambda \to 0} \frac{1}{\lambda} \cdot \Big(f(x \oplus \delta x) \ominus f(x)\Big) \\ &= A \otimes \frac{\partial}{\partial x_j} f(x)\end{aligned}$$

With these results in hand we may now compute the *Jacobian* of a stochastic function as follows.

**Definition.** JACOBIAN: The *Jacobian* of a stochastic function, $f(x) = [f_i(x)] \in {}^m\mathbb{F}^n$ with $x \in {}^n\mathbb{S}$, denoted, $\frac{\partial f}{\partial x} \in {}^m\mathbb{S}^n$, is

$$\begin{aligned}\frac{\partial f}{\partial x} &= \left[\frac{\partial}{\partial x_1} f \cdots \frac{\partial}{\partial x_n} f\right] \\ &= \downarrow\left[e^{\frac{x_j}{f_i(x)} \frac{\partial f_i(x)}{\partial x_j}}\right]\end{aligned}$$

where $\frac{\partial f_i(x)}{\partial x_j}$ indicates the scalar partial differentiation of matrix algebra.

Some examples may be useful.

**Examples.** Let $x, y, z \in {}^m\mathbb{S}$. Let $\lambda \in \mathbb{R}$. Let $A \in {}^m\mathbb{S}^m$. Let $f(x), g(x), h(x) \in {}^m\mathbb{F}^n$.

$$\begin{array}{ll} f(x) = y & \frac{\partial f}{\partial x} = \Omega \\ f(x) = x & \frac{\partial f}{\partial x} = \Xi \\ f(x) = \lambda \cdot x & \frac{\partial f}{\partial x} = \lambda \cdot \Xi \\ f(x) = A \otimes x & \frac{\partial f}{\partial x} = A \\ f(x) = \lambda \cdot g(x) & \frac{\partial f}{\partial x} = \lambda \cdot \frac{\partial g}{\partial x} \\ f(x) = g(x) \ominus h(x) & \frac{\partial f}{\partial x} = \frac{\partial g}{\partial x} \oplus \frac{\partial h}{\partial x} \\ f(x) = A \otimes g(x) & \frac{\partial f}{\partial x} = A \otimes \frac{\partial g}{\partial x} \\ f(x) = (x \rangle\langle y) \otimes z & \frac{\partial f}{\partial x} = \langle y, z \rangle \cdot \Xi \end{array}$$

□

## 2.13 Probability Distributions

It should be obvious at this point that stochastic matrices are meant to represent probability distributions. There are a few points which must be mentioned for completeness. Let $X$ be a random variable which may take on discrete values $X_1, \ldots, X_m$. The $X_i$ $i = 1 \ldots m$ are mutually exclusive events or *states* . The frequency with which $X$ visits each of the states, $X_i$, is a probability distribution which is represented by a stochastic matrix $x = [x_i] \in {}^m\mathbb{S}$. We have that the probability of state $i$, denoted $p(X = X_i)$ is $x_i$. Note that $x_i > 0 \; \forall i = 1 \ldots m$ and $\sum_{i=1}^m x_i = 1$ which is an axiom of probability theory known as the *axiom of total probability.*

## 2.14 Joint Probability Distributions

The notion of the joint probability distribution is of paramount importance in what is to follow. One of the central ideas behind stochastic algebra is that statistical independence is embodied as linearity. To discuss this further we must introduce the *joint probability distribution.*

Let $X$ and $Y$ be random variables with $m$ and $n$ states respectively. A fundamental axiom of probability theory (Bayes' law) is that the joint probability distribution of the two random variables may be expressed as

$$\begin{aligned} p(X = X_i, Y = Y_j) &= p(X_i, Y_j) \\ &= p(X_i)p(Y_j|X_i) \\ &= p(Y_j)p(X_i|Y_j) \\ &= p(X_i)p(Y_j)\left(\frac{p(X_i|Y_j)p(Y_j|X_i)}{p(X_i)p(Y_j)}\right)^{\frac{1}{2}} \end{aligned}$$

This is called the *product rule of probabilities.* Note that when $p(X_i|Y_j) = p(X_i)$ and $p(Y_j|X_i) = p(Y_j)$ we have that

$$p(X_i, Y_j) = p(X_i)p(Y_j)$$

which is called *statistical independence.* When we do not have statistical independence, the individual probability distributions may be computed as

$$\begin{aligned} p(X_i) &= \sum_{j=1}^{n} p(X_i, Y_j) \\ p(Y_j) &= \sum_{i=1}^{m} p(X_i, Y_j) \end{aligned}$$

In the stochastic algebra we may represent joint distributions as follows.

**Definition.** JOINT DISTRIBUTION[11]: Let $X$ and $Y$ be random variables with $m$ and $n$ states respectively. The *joint distribution* between $X$ and $Y$ is denoted $(\boldsymbol{x},\boldsymbol{y}) \in {}^{mn}\mathbb{S}$.

$$(\boldsymbol{x},\boldsymbol{y}) = \left[p(X_1, Y_1) \cdots p(X_1, Y_n)\ p(X_2, Y_1) \cdots p(X_m, Y_n)\right]^{\mathrm{T}}$$

This may be generalized to more than two random variables.

It is not difficult to see that when $X$ and $Y$ are independent, the joint probability distribution may be expressed in the form

$$(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{A} \otimes \boldsymbol{x} \oplus \boldsymbol{B} \otimes \boldsymbol{y}$$

where $\boldsymbol{A} \in {}^{mn}\mathbb{S}^m$ and $\boldsymbol{B} \in {}^{mn}\mathbb{S}^n$. That is, the joint probability distribution of two statistically independent random variables is a linear combination of their individual distributions. The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are used to combine the independent distributions; they are similar to projection matrices (an example follows). In the case that $X$ and $Y$ are not independent, we have

$$(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{A} \otimes \boldsymbol{x} \oplus \boldsymbol{B} \otimes \boldsymbol{y} \oplus \boldsymbol{z}$$

[11] When $X$ and $Y$ are independent, the joint distribution as defined here is equivalent to the Kroenecker product.

where $z = (x,y) \ominus A \otimes x \ominus B \otimes y \in {}^{mn}\mathbb{S}$. For $X$ and $Y$ independent $z = \omega$ which restores linearity. When $z \neq \omega$ it at the very least represents an offset which destroys the linear property.

**Example.** Consider the case where both $X$ and $Y$ have only 2 states each and are statistically independent.

$$(x,y) = \begin{bmatrix} x_1y_1 \\ x_1y_2 \\ x_2y_1 \\ x_2y_2 \end{bmatrix} = f(x,y) = \frac{\partial f}{\partial x} \otimes x \oplus \frac{\partial f}{\partial y} \otimes y = \begin{bmatrix} e & 1 \\ e & 1 \\ 1 & e \\ 1 & e \end{bmatrix} \otimes \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \oplus \begin{bmatrix} e & 1 \\ 1 & e \\ e & 1 \\ 1 & e \end{bmatrix} \otimes \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = A \otimes x \oplus B \otimes y$$

where we have employed the Jacobian to compute $A = \frac{\partial f}{\partial x}$, $B = \frac{\partial f}{\partial y} \in {}^4\mathbb{S}^2$. In this case the function, $f(x,y)$, is linear in $x$ and $y$ precisely because $X$ and $Y$ are statistically independent. Furthermore, notice that

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} e & e & 1 & 1 \\ 1 & 1 & e & e \end{bmatrix} \otimes (x,y) \qquad \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} e & 1 & e & 1 \\ 1 & e & 1 & e \end{bmatrix} \otimes (x,y)$$

such that

$$\begin{aligned}
(x,y) &= \frac{1}{2} \cdot \begin{bmatrix} e & 1 \\ e & 1 \\ 1 & e \\ 1 & e \end{bmatrix} \otimes \begin{bmatrix} e & e & 1 & 1 \\ 1 & 1 & e & e \end{bmatrix} \otimes (x,y) \oplus \frac{1}{2} \cdot \begin{bmatrix} e & 1 \\ 1 & e \\ e & 1 \\ 1 & e \end{bmatrix} \otimes \begin{bmatrix} e & 1 & e & 1 \\ 1 & e & 1 & e \end{bmatrix} \otimes (x,y) \\
&= \frac{1}{2} \cdot \begin{bmatrix} e & e & 1 & 1 \\ e & e & 1 & 1 \\ 1 & 1 & e & e \\ 1 & 1 & e & e \end{bmatrix} \otimes (x,y) \oplus \frac{1}{2} \cdot \begin{bmatrix} e & 1 & e & 1 \\ 1 & e & 1 & e \\ e & 1 & e & 1 \\ 1 & e & 1 & e \end{bmatrix} \otimes (x,y) \\
&= \underset{\to b_1}{\text{proj}}\,(x,y) \oplus \underset{\to b_2}{\text{proj}}\,(x,y) \\
&= \underset{\to \{b_1, b_2\}}{\text{proj}}\,(x,y)
\end{aligned}$$

where $b_1 = \begin{bmatrix} e & e & 1 & 1 \end{bmatrix}^T$, $b_2 = \begin{bmatrix} e & 1 & e & 1 \end{bmatrix}^T$, $b_3 = \begin{bmatrix} e & 1 & 1 & e \end{bmatrix}^T$ form a basis for ${}^4\mathbb{S}$. Since $(x,y)$ is unaffected by the projection it must lie in the subspace formed by $\text{sp}\{b_1, b_2\}$ which is of dimension 2 not 3 (the dimension of ${}^4\mathbb{S}$). In fact, $X$ and $Y$ are statistically independent if and only if $(x,y) \in \text{sp}\{b_1, b_2\}$, the *linear subspace* of ${}^4\mathbb{S}$. □

The example has brought to light a new concept, the *linear subspace*. In general, if the set of all possible joint probability distributions, $(x,y)$, is ${}^{mn}\mathbb{S}$, there will be a subspace of dimension $m + n - 2 < mn$ which will be called the *linear subspace*, where $x \in {}^m\mathbb{S}$ and $y \in {}^n\mathbb{S}$. Joint probability distributions are in the linear subspace if and only if $X$ and $Y$ are independent.

## 2.15 The Biased Inverse

If we consider $A \in {}^m\mathbb{S}^n$ to be a conditional probability distribution then we may speak of $y = Ax$ where $y = [p(Y_i)]$ and $x = [p(X_j)]$ and $A = [p(Y_i|X_j)]$.

**Definition.** BIASED INVERSE: Let $A = [p(Y_i|X_j)] \in {}^m\mathbb{S}^n$. The *biased inverse*, denoted $A^\iota$, is

$$A^\iota = [p(X_i|Y_j)]$$

where $A^\iota \in {}^n\mathbb{S}^m$.

The biased inverse of a stochastic matrix is very important when working backwards from a measured effect to an unmeasured cause. Unfortunately, $A^\iota$ cannot be computed directly from $A$ as we do not know the correct bias.

As the names allude, there is a connection between the biased and unbiased inverse operators.

**Proposition.** *The unbiased inverse is related to the biased inverse in the following way for any matrix* $A = [a_{ij}] \in {}^m\mathbb{S}^n$

$$A^\iota \equiv A^\intercal \oplus X_b \tag{2.1}$$

*where* $X_b \in {}^n\mathbb{S}^m$ *can be thought of as a bias which has been vectorially added to the unbiased inverse.*

**Proof.** From the definition of the biased inverse and Bayes' axiom we have

$$\begin{aligned} A^\iota &= [p(X_i|Y_j)] \\ &= \left[\frac{p(Y_j|X_i)p(X_i)}{\sum_k p(Y_j|X_k)p(X_k)}\right] \\ &= \left[\frac{p(Y_j|X_i)}{\sum_k p(Y_j|X_k)}\right] \oplus [p(X_i)] \\ &= A^\intercal \oplus X_b \end{aligned}$$

where $X_b = [p(X_i)] \in {}^n\mathbf{S}^m$. Notice in $X_b$ there is only one subscript, $i$, shown. This implies that each column of $X_b$ is identical. It is possible to have a different bias for each column but typically the same one is used. Note that when $X_b = \Omega$ we have $A^\iota = A^\intercal$. ■

**Lemma.** When postmultiplying by a deterministic column, $\mathbf{y} \in {}^m\mathbb{S}$ we have

$$A^\iota \mathbf{y} \equiv A^\intercal \mathbf{y} \oplus x_b \tag{2.2}$$

where $x_b \in {}^n\mathbb{S}$ is again the bias. Note, in this case the bias may be added at the end rather than before computing an inverse.

If we wish to work backwards from a measured effect to an unmeasured cause, we would like to know $A^\iota$. However, what we are saying is that there is a part of $A^\iota$ which we do not know, $x_b$. This may be thought of as a bias. Without knowing the bias we cannot calculate $A^\iota$. Instead, we must assume a bias. The most logical choice is to assume $x_b = {}^m\omega$ which we will refer to as a *uniform bias.*

## 2.16 A Nonlinear Operation

Matrix multiplication[12] plays a large role in this framework. A stochastic matrix may be thought of as a conditional probability distribution. With this in mind, matrix multiplication (e.g., $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ where $\boldsymbol{y} = [p(Y_i)] \in {}^n\mathbb{S}$ and $\boldsymbol{x} = [p(X_j)] \in {}^m\mathbb{S}$ and $\boldsymbol{A} = [p(Y_i|X_j)] \in {}^n\mathbb{S}^m$) embodies the axiom of ***conditional probability***[13].

$$p(Y_i) = \sum_{j=1}^{m} p(Y_i|X_j)p(X_j), \quad \forall i = 1 \ldots n$$

Often normal matrix multiplication will appear in mathematical expressions involving the new stochastic algebra. In general it must be thought of as a ***nonlinear operation*** within the new framework[14]. In general we have the following situation. Let $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \in \mathbb{S}$.

$$\begin{aligned} (\boldsymbol{A} \oplus \boldsymbol{B})\boldsymbol{C} &\neq \boldsymbol{AC} \oplus \boldsymbol{BC} \\ \boldsymbol{C}(\boldsymbol{A} \oplus \boldsymbol{B}) &\neq \boldsymbol{CA} \oplus \boldsymbol{CB} \end{aligned}$$

That is, vector addition does not distribute over matrix multiplication, which certainly is not desirable. However, there are a few exceptions to this statement. There are two distributive results involving normal matrix multiplication between deterministic and stochastic matrices which are as follows. Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{S}$ and $\mathbf{C} \in \mathbb{D}$.

$$(\boldsymbol{A} \oplus \boldsymbol{B})\mathbf{C} = \boldsymbol{A}\mathbf{C} \oplus \boldsymbol{B}\mathbf{C} \tag{2.3}$$

$$\mathbf{C}^T(\boldsymbol{A} \oplus \boldsymbol{B}) = \mathbf{C}^T\boldsymbol{A} \oplus \mathbf{C}^T\boldsymbol{B} \tag{2.4}$$

where the dimensions of each matrix should be compatible for the given matrix multiplications. The first of these is now proved (the second follows from similar reasoning).

**Proof.** $(\boldsymbol{A} \oplus \boldsymbol{B})\mathbf{c} = \boldsymbol{A}\mathbf{c} \oplus \boldsymbol{B}\mathbf{c}$

We need only prove the result for a single deterministic column. Let $\boldsymbol{A} = [a_{ij}]$, $\boldsymbol{B} = [b_{ij}] \in {}^m\mathbb{S}^n$ and $\mathbf{c} = [c_j] \in {}^n\mathbb{D}$. We have that

$$c_j = \begin{cases} 1 & j = j_o \\ 0 & j \neq j_o \end{cases}, \quad \forall j = 1 \ldots n$$

Starting with the left hand side we have

$$\begin{aligned} (\boldsymbol{A} \oplus \boldsymbol{B})\mathbf{c} &= \left[\frac{a_{ij}b_{ij}}{\sum_{k=1}^{m} a_{kj}b_{kj}}\right]\mathbf{c} \\ &= \left[\frac{a_{ij_o}b_{ij_o}}{\sum_{k=1}^{m} a_{kj_o}b_{kj_o}}\right] \\ &= [a_{ij_o}] \oplus [b_{ij_o}] \\ &= \boldsymbol{A}\mathbf{c} \oplus \boldsymbol{B}\mathbf{c} \end{aligned}$$

concluding the proof. ■

[12]This is the regular matrix multiplication of matrix algebra not the vector product of stochastic algebra.
[13]This is sometimes called the axiom of total probability or the *Chapman-Kolmogoroff* equation [Papoulis, 1965].
[14]We could just as well think of vector addition as a nonlinear operation within normal linear algebra.

It should be mentioned that these are not the only situations in which matrix multiplication distributes over vector addition. To show this we continue the example from the section on joint distributions.

**Example.** Continuing from the previous example. Instead of using the projections of stochastic algebra to compute the individual distributions we may use matrix multiplication to do the same job as follows

$$x = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} (x,y) = \mathbf{P}_x (x,y) \qquad y = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} (x,y) = \mathbf{P}_y (x,y)$$

where $\mathbf{P}_x,\ \mathbf{P}_y \in {}^2\mathbb{D}^4$. We also have that

$$(x,y) = \mathbf{P}_x{}^T x \oplus \mathbf{P}_y{}^T y$$

and we note that

$$\mathbf{P}_x{}^T \mathbf{P}_x \oplus \mathbf{P}_y{}^T \mathbf{P}_y = \mathbf{1} \qquad \mathbf{P}_x{}^T \mathbf{P}_y = \mathbf{\Omega} \qquad \mathbf{P}_y{}^T \mathbf{P}_x = \mathbf{\Omega}$$

such that

$$\begin{aligned} (\mathbf{P}_x{}^T \mathbf{P}_x \oplus \mathbf{P}_y{}^T \mathbf{P}_y)(x,y) &= \mathbf{1}(x,y) \\ &= \mathbf{P}_x{}^T x \oplus \mathbf{P}_y{}^T y \\ &= \mathbf{P}_x{}^T \mathbf{P}_x (x,y) \oplus \mathbf{P}_y{}^T \mathbf{P}_y (x,y) \end{aligned}$$

which shows that $(x,y) \in {}^4\mathbf{S}$ distributes over the stochastic matrices $\mathbf{P}_x{}^T \mathbf{P}_x,\ \mathbf{P}_y{}^T \mathbf{P}_y \in {}^4\mathbf{S}^4$. This of course only occurs when $(x,y)$ is in the linear subspace of ${}^4\mathbf{S}$. □

We may see from the example that in special situations, a projection matrix (with a vector product) may be replaced by a normal matrix multiplication. In general if $X_1, \ldots, X_k$ are statistically independent random variables with a joint distribution, $x = (x_1, \ldots, x_k)$, then it is possible to express the individual distributions as $x_i = \mathbf{P}_i{}^T x$ with $\mathbf{P}_i \in \mathbb{D}\ \forall i = 1 \ldots k$. The following are then true

$$\mathbf{P}_i{}^T \mathbf{P}_j = \mathbf{\Omega} \quad i \neq j \qquad\qquad \boldsymbol{B}_i \otimes \boldsymbol{B}_j = \mathbf{\Omega} \quad i \neq j$$

$$\bigoplus_{i=1}^{k} \mathbf{P}_i{}^T \mathbf{P}_i = \mathbf{1} \qquad\qquad \bigoplus_{i=1}^{k} \boldsymbol{B}_i = \boldsymbol{\Xi}$$

$$\left( \bigoplus_{i=1}^{k} \mathbf{P}_i{}^T \mathbf{P}_i \right) x = \bigoplus_{i=1}^{k} \mathbf{P}_i{}^T \mathbf{P}_i x \qquad \left( \bigoplus_{i=1}^{k} \boldsymbol{B}_i \right) \otimes x = \bigoplus_{i=1}^{k} \boldsymbol{B}_i \otimes x$$

where the $\boldsymbol{B}_i$ are equivalent orthogonal projection matrices (computed using the Jacobian, for example).

## 2.17 The Unbiased Inverse

There are three more properties involving the unbiased inverse which are stated here for completeness. Let $\mathbf{A}, \mathbf{B} \in \mathbb{D}$ with no zero rows.

$$(\mathbf{A}^T \oplus \mathbf{B}^T) = (\mathbf{A} \oplus \mathbf{B})^T \tag{2.5}$$

$$(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T \tag{2.6}$$

$$\mathbf{A}\mathbf{A}^T = \mathbf{1} \tag{2.7}$$

The dimensions of $\mathbf{A}$ and $\mathbf{B}$ must compatible with the given operations.

## 2.18 Information Theory

There are several existing *information* [Shannon, 1948] measures defined for probability distributions. Some of the possibilities are as follows. Let $\boldsymbol{x} = [x_i]$, $\boldsymbol{y} = [y_i] \in {}^m\mathbb{S}$.

| | |
|---|---|
| SUM-OF-SQUARES INFORMATION | $S(\boldsymbol{x}) = \sum_{i=1}^{m} x_i^2$ |
| SHANNON INFORMATION | $H(\boldsymbol{x}) = -\sum_{i=1}^{m} x_i \ln x_i$ |
| KULLBECK-LEIBLER DISTANCE | $G(\boldsymbol{x},\boldsymbol{y}) = -\sum_{i=1}^{m} x_i \ln\left(\frac{x_i}{y_i}\right)$ |

Given the connection with vector spaces, a new information measure is now defined which we will refer to simply as *information*

**Definition.** INFORMATION: Let $\boldsymbol{x} = [x_i] \in {}^m\mathbb{S}$. The *information* contained in $\boldsymbol{x}$, denoted $||\boldsymbol{x}||$, is

$$||\boldsymbol{x}|| = \sqrt{\langle \boldsymbol{x},\boldsymbol{x} \rangle}$$

which is the *vector norm* or *magnitude* of $\boldsymbol{x}$.

This measure can be thought of as the "distance" from the uniform probability distribution. Based on the general properties of an inner product space, we may now immediately state the following *information properties* (using the same vectors as above).

| | | | |
|---|---|---|---|
| TRIANGLE INEQUALITY | $\|\boldsymbol{x} \oplus \boldsymbol{y}\|$ | $\leq$ | $\|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ |
| SCHWARZ INEQUALITY | $\|\langle \boldsymbol{x},\boldsymbol{y} \rangle\|$ | $\leq$ | $\|\boldsymbol{x}\|\,\|\boldsymbol{y}\|$ |
| PYTHAGOREAN THEOREM ($\boldsymbol{x} \perp \boldsymbol{y}$) | $\|\boldsymbol{x} \ominus \boldsymbol{y}\|^2$ | $=$ | $\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2$ |
| INFORMATION DISTANCE | | | $\|\boldsymbol{x} \ominus \boldsymbol{y}\|$ |

It is very natural to discuss information theory in the context of projections. If projections are thought of as filters, then there is an *information loss* when a vector is projected. Let $\boldsymbol{A} \in {}^m\mathbb{S}^n$ and let $\boldsymbol{P} \in {}^m\mathbb{S}^m$ be a projection matrix. The loss in information as a result of the projection is

$$||\boldsymbol{A}|| - ||\boldsymbol{P} \otimes \boldsymbol{A}||$$

## 2.19 Summary

An algebra based on stochastic matrices has been constructed. The set of stochastic matrices was shown to constitute a vector space, an inner product space, and an algebra under appropriate operations. The zero vector is the uniform probability distribution and addition is akin to statistical independence. A stochastic calculus and an isomorphism to matrix algebra were also presented. The original purpose of this algebra was to allow Markov systems to be cast in a matrix form but it has the potential for a great deal more. We may now discuss stochastic dynamic equations and control systems in the well known mathematical framework of linear algebra.

The following tables summarize the connections between probability theory and the stochastic algebra defined in this chapter as well as the main properties and identities derived.

Table 2.1: Summary of connections between stochastic algebra and probability theory.

| STOCHASTIC ALGEBRA | PROBABILITY THEORY |
|---|---|
| vector (stochastic matrix) | probability distribution |
| zero vector (uniform matrix) | uniform probability distribution |
| linearity | statistical independence |
| nonlinearity | statistical dependence |
| columns of a stochastic matrix sum to 1 | axiom of total probability |
| matrix multiplication | conditional probability axiom |
| biased/unbiased inverse | Bayes' axiom |
| vector addition | product rule of probabilities |
| scalar multiplication | exponent of probability distribution |
| inner product of two vectors | logarithm of probability distribution |
| vector product | |
| norm of a vector | information |
| triangle inequality | information never more than the sum of the parts |
| Schwarz inequality | information statement |
| Pythagorean theorem | information statement |
| projection to a subspace | information filtering |

Table 2.2: Summary of the stochastic algebra properties and identities.

| Properties of Addition | |
|---|---|
| Identity | $\boldsymbol{A} \oplus \boldsymbol{\Omega} = \boldsymbol{\Omega} \oplus \boldsymbol{A} = \boldsymbol{A}$ |
| Left Distributivity | $\lambda \cdot (\boldsymbol{A} \oplus \boldsymbol{B}) = \lambda \cdot \boldsymbol{A} \oplus \lambda \cdot \boldsymbol{B}$ |
| Right Distributivity | $(\lambda + \mu) \cdot \boldsymbol{A} = \lambda \cdot \boldsymbol{A} \oplus \mu \cdot \boldsymbol{A}$ |
| Scalar Multiplication | $(\lambda\mu) \cdot \boldsymbol{A} = \lambda \cdot (\mu \cdot \boldsymbol{A}) = \mu \cdot (\lambda \cdot \boldsymbol{A})$ |
| Commutativity | $\boldsymbol{A} \oplus \boldsymbol{B} = \boldsymbol{B} \oplus \boldsymbol{A}$ |
| Associativity | $(\boldsymbol{A} \oplus \boldsymbol{B}) \oplus \boldsymbol{C} = \boldsymbol{A} \oplus (\boldsymbol{B} \oplus \boldsymbol{C})$ |

| Properties of Inner Product | |
|---|---|
| Distributivity | $\langle \boldsymbol{x}, \boldsymbol{y} \oplus \boldsymbol{z} \rangle = \langle \boldsymbol{x}, \boldsymbol{y} \rangle + \langle \boldsymbol{x}, \boldsymbol{z} \rangle$ |
| Scalar Multiplication | $\langle \lambda \cdot \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{x}, \lambda \cdot \boldsymbol{y} \rangle = \lambda \langle \boldsymbol{x}, \boldsymbol{y} \rangle$ |
| Commutativity | $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{y}, \boldsymbol{x} \rangle$ |

| Properties of Outer Product | |
|---|---|
| Left Distributivity | $\boldsymbol{x} \rangle\langle (\boldsymbol{y} \oplus \boldsymbol{z}) = (\boldsymbol{x} \rangle\langle \boldsymbol{y}) \oplus (\boldsymbol{x} \rangle\langle \boldsymbol{z})$ |
| Right Distributivity | $(\boldsymbol{x} \oplus \boldsymbol{y}) \rangle\langle \boldsymbol{z} = (\boldsymbol{x} \rangle\langle \boldsymbol{z}) \oplus (\boldsymbol{y} \rangle\langle \boldsymbol{z})$ |
| Scalar Multiplication | $(\lambda \cdot \boldsymbol{x}) \rangle\langle \boldsymbol{y} = \boldsymbol{x} \rangle\langle (\lambda \cdot \boldsymbol{y}) = \lambda \cdot (\boldsymbol{x} \rangle\langle \boldsymbol{y})$ |
| Commutativity | $(\boldsymbol{x} \rangle\langle \boldsymbol{y}) \otimes \boldsymbol{z} = (\boldsymbol{x} \rangle\langle \boldsymbol{z}) \otimes \boldsymbol{y}$ |
| Relation to Inner Product | $(\boldsymbol{x} \rangle\langle \boldsymbol{y}) \otimes \boldsymbol{z} = \langle \boldsymbol{y}, \boldsymbol{z} \rangle \cdot \boldsymbol{x}$ |

| Properties of Vector Product | |
|---|---|
| Identity | $\boldsymbol{A} \otimes \boldsymbol{\Xi} = \boldsymbol{\Xi} \otimes \boldsymbol{A} = \boldsymbol{A}$ |
| Left Distributivity | $\boldsymbol{A} \otimes (\boldsymbol{B} \oplus \boldsymbol{C}) = \boldsymbol{A} \otimes \boldsymbol{B} \oplus \boldsymbol{A} \otimes \boldsymbol{C}$ |
| Right Distributivity | $(\boldsymbol{A} \oplus \boldsymbol{B}) \otimes \boldsymbol{C} = \boldsymbol{A} \otimes \boldsymbol{C} \oplus \boldsymbol{B} \otimes \boldsymbol{C}$ |
| Scalar Multiplication | $(\lambda \cdot \boldsymbol{A}) \otimes \boldsymbol{B} = \boldsymbol{A} \otimes (\lambda \cdot \boldsymbol{B}) = \lambda \cdot (\boldsymbol{A} \otimes \boldsymbol{B})$ |
| Associativity | $(\boldsymbol{A} \otimes \boldsymbol{B}) \otimes \boldsymbol{C} = \boldsymbol{A} \otimes (\boldsymbol{B} \otimes \boldsymbol{C})$ |

| Properties Involving Matrix Algebra | |
|---|---|
| Self-Inverse | $(\mathbf{A}^T)^T = \mathbf{A}$ |
| Distributivity | $(\mathbf{A} \oplus \mathbf{B})^T = \mathbf{A}^T \oplus \mathbf{B}^T$ |
| Product | $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$ |
| Left Distributivity | $\mathbf{A}^T (\boldsymbol{B} \oplus \boldsymbol{C}) = \mathbf{A}^T \boldsymbol{B} \oplus \mathbf{A}^T \boldsymbol{C}$ |
| Right Distributivity | $(\boldsymbol{A} \oplus \boldsymbol{B})\mathbf{C} = \boldsymbol{A}\mathbf{C} \oplus \boldsymbol{B}\mathbf{C}$ |

It is only the observer of the Creature who imputes a central representation or central control. The Creature itself has none; it is a collection of competing behaviours. Out of the local chaos of their interactions there emerges, in the eye of the observer, a coherent pattern of behaviour. There is no central purposeful locus of control.

—Rodney A Brooks

INTELLIGENCE WITHOUT REPRESENTATION, 1999

# Chapter 3

# STOCHASTIC CONTROL SYSTEMS

There are a number of mathematical frameworks available to the control engineer. Examples include linear control systems [Kalman, 1960], discrete event systems [Ramadge and Wonham, 1987], and Markov systems [Bellman, 1957]. Each of these frameworks has its own mathematical structure which may be used to derive general results concerning control and observation. In this thesis we are primarily interested in decentralized systems, a general concept which has been visited in all of the above frameworks. We will begin discussing decentralized control in the context of Markov systems but the stochastic algebra (and calculus) of the previous chapter will allow a new link to be made between Markov systems and the classic linear system studied by Kalman [1960]. Markov systems were originally chosen for this study for their strong connection to probability theory and artificial intelligence (through machine learning). Many machine learning approaches are stochastic in nature (e.g., Boltzmann machines, Hidden Markov models, reinforcement learning, Hopfield networks). Markov systems are furthermore described by stochastic dynamic equations which provide a connection to statistical physics and entropy. They seem a promising place to begin a study of decentralized systems and self-organization.

The problem of decentralized control is concerned with producing useful system behaviour through the *independent* observations and actions of $K > 1$ controllers. The fact that the controllers are acting independently is such a fundamental point that it will occupy a great deal of this chapter. Decentralization has a major benefit in that it offers redundancy. For example, should one part of the system fail, it may be possible for the other $K-1$ to produce the appropriate behaviour through compensation. Furthermore, it allows modularity as each controller may be designed somewhat independently. How-

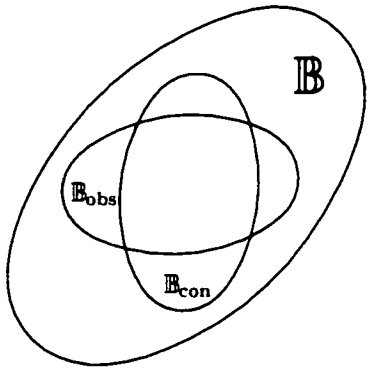



Figure 3.1: Reduction of the set of possible system behaviour due to constraints imposed by decentralization of observation and control. The set of possible behaviours is reduced from $\mathbb{B}$ to $\mathbb{B}_{obs} \cap \mathbb{B}_{con}$.

ever, there is often a cost associated with decentralization, namely that by themselves, decentralized controllers may not be able to perform all of the tasks inherently possible with a centralized controller. Thus the focus of this chapter will be to discuss when decentralized controllers can perform as well as their centralized counterparts. This will be investigated in the context of a few different systems.

We will see that when strictly decentralized controllers are asked to behave in a statistically dependent manner, they may only approximate their centralized counterpart. Decentralization essentially adds two new constraints to the centralized system. The first constraint is on the observations (input). Each decentralized controller will in general receive only a part of the system state as input. The second constraint is on the control (output). Each decentralized controller must select its own actions, thus making it impossible to coordinate actions with other controllers. Abstractly, these two constraints each divide the set of possible system behaviours into two subsets. Figure 3.1 graphically depicts these constraints.

We conjecture that the very process of decentralization in general contrains system behaviour in two fundamental ways. Let $\mathbb{B}$ be the set of system behaviours (trajectories) which are possible using centralized control. Let $\mathbb{B}_{obs} \subseteq \mathbb{B}$ be the set of behaviours which are possible using decentralized observations. Let $\mathbb{B}_{con} \subseteq \mathbb{B}$ be the set of behaviours which are possible using decentralized controls. Then the set of possible behaviours as a result of decentralized observations and controls is

$$\left(\mathbb{B}_{obs} \cap \mathbb{B}_{con}\right) \subseteq \begin{matrix} \mathbb{B}_{obs} \\ \mathbb{B}_{con} \end{matrix} \subseteq \mathbb{B} \tag{3.1}$$

This is completely analogous to the controllability and observability of centrally controlled linear systems [Kalman, 1960]. There the constraint on observations is a limited number of sensors. The constraint on controls is a limited number of actuators. This parallel may be seen throughout control theory. We will study this idea in a few specific situations.

If the decentralized controllers are allowed to communicate perfectly with one another, they may exactly produce the centralized behaviour. Perfect communication serves a dual role, effectively removing the constraints described above. First, it allows sharing of observations which relieves the constraint on

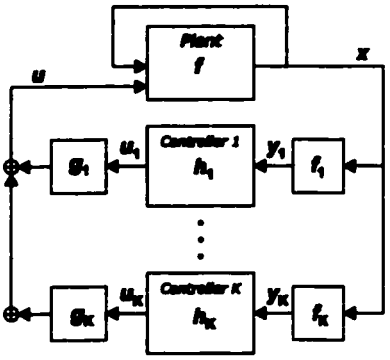



Figure 3.2: Graphical depiction of a decentralized control system. $f \in {}^{m}\mathbb{F}^{m,n}$ is the plant we are trying to control, $(\forall k = 1 \ldots K)$ the $f_k \in {}^{m_k}\mathbb{F}^{m}$ are the observation projection functions, the $h_k \in {}^{n_k}\mathbb{F}^{m_k}$ are the control laws, and the $g_k \in {}^{n}\mathbb{F}^{n_k}$ are the action projection functions.

input. Second, it allows actions to be coordinated before execution. Limited bandwidth[1] of communication is certainly of concern but this issue aside, it is not a stretch to see that in general, decentralized controllers that communicate both observations and actions may produce exactly the same behaviour as a centralized controller. This idea will be investigated in the context of reactive controllers for Markov systems.

It is hoped that the stochastic algebra detailed in the previous chapter will help to formalize our study of decentralized Markov systems. The fact that statistical independence has been embodied as linearity will allow us to examine the nature of decentralization in a formal mathematical setting while providing insights into its very nature.

From a control theoretic viewpoint, we are primarily concerned with systems which may be described by the discrete-time dynamic stochastic equation

$$\boldsymbol{x}[t+1] = \boldsymbol{f}(\boldsymbol{x}[t], \boldsymbol{u}_1[t], \ldots, \boldsymbol{u}_K[t])$$

where $\boldsymbol{x} \in {}^{m}\mathbb{S}$ is a probability distribution over the states of the system, $\boldsymbol{u}_k \in {}^{n_k}\mathbb{S}$ $\forall k = 1 \ldots K$ are decentralized controls, and $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}_1[t], \ldots, \boldsymbol{u}_K[t]) \in {}^{m}\mathbb{F}^{m,n_1,\ldots,n_K}$ describes how the system moves forward in time, $t$. In general, $\boldsymbol{f}$ may be a nonlinear stochastic function.

Our definition of decentralized controls is as follows.

**Definition.** DECENTRALIZED CONTROLS: Let $\boldsymbol{u}_k \in {}^{n_k}\mathbb{S}$ $\forall k = 1 \ldots K$ be the controls of $K$ controllers. If a centralized control, $\boldsymbol{u} \in {}^{n}\mathbb{S}$, may be expressed in the additive form

$$\boldsymbol{u} = \bigoplus_{k=1}^{K} \boldsymbol{g}_k(\boldsymbol{u}_k) \tag{3.2}$$

where $\boldsymbol{g}_k \in {}^{n}\mathbb{F}^{n_k}$ $\forall k = 1 \ldots K$, then the $\boldsymbol{u}_k$ are *decentralized controls* for $\boldsymbol{u}$.

This embodies the notion that the decentralized controllers must act independently. Figure 3.2 depicts the mathematical model with which we will be concerned. This chapter will investigate two different possibilities for the plant function, $\boldsymbol{f}$: linear and nonlinear. Before continuing, we pause to review the history of studies on decentralized control.

[1]This will be discussed in the chapter on communication.

## 3.1 Related Work

This review of decentralized control will begin chronologically but after a certain point in the timeline there will be a number of bifurcations as decentralized research has branched into many separate fields. Some of these will be described in more detail than others.

The review will tend to use the the term *decentralized control system* to mean a group of agents or decision makers which interact with some common environment or plant, whose group behaviour is to be controlled in some way. Research in game theory could arguably be put in this category but it will be avoided to leave room for work which fits more in the area of *control theory*. The focus is on control of Markov systems but we also mention the fundamental results from control of linear systems as we will later be suggesting a link between the two.

Decentralized research can be traced back over 45 years to the work of Marschak [1955] who looked at human organizations (e.g., military, corporations) as technologies. He sought to improve their efficiency by looking at decentralized decision making from a formal perspective. He noticed the fundamental role communication must play in all such systems.

Shortly after this Radner [1962] published a paper on teams (decentralized systems with a common goal). He pointed out that "differences of opinion (as embodied in different a priori distributions, for example)" between controllers could not be handled in his formulation "since these result in formally the same game-theoretic difficulties as do conflicts of interest". We will revisit this notion later but it is of paramount importance in decentralized systems with different state estimates of the world. Radner was the first to connect decentralized systems to Markov systems (to be described later).

### 3.1.1 Decentralized Control of Linear Systems

We must here mention the landmark work of Kalman [1960, 1962] who introduced the notions of controllability and observability as well as the canonical structure of centralized linear dynamical systems.

Ten years later Ho and Chu [1972] published two papers on decentralized systems as optimal control problems. They chose a particular type of discrete-time linear control problem for their investigations and looked at when optimal linear solutions existed. They also studied the role of communication links between decision makers.

Shortly after this Aoki [1972] looked at the decentralized version of the familiar linear time-invariant multivariable system (continuous time)

$$\dot{\mathsf{x}} = \mathsf{A}\mathsf{x} + \sum_{k=1}^{K} \mathsf{B}_k \mathsf{u}_k \tag{3.3}$$

where $\mathsf{A} \in {}^{m}\mathbb{R}^{m}$, $\mathsf{x} \in {}^{m}\mathbb{R}$, $\mathsf{B}_k \in {}^{m}\mathbb{R}^{n_k}$, $\mathsf{u}_k \in {}^{n_k}\mathbb{R}$ $\forall k = 1 \ldots K$. They had a dynamic feedback law for each controller depending on different partial system outputs.

$$\mathsf{y}_k = \mathsf{C}_k \mathsf{x} \quad \forall k = 1 \ldots K \tag{3.4}$$

where $\mathsf{y}_k \in {}^{p_k}\mathbb{R}$, $\mathsf{C}_k \in {}^{p_k}\mathbb{R}^{m}$ $\forall k = 1 \ldots K$. In was shown that the agents must in general communicate state information with one another for the system to be feedback stabilizable.

Wang and Davison [1973] generalized the notions of uncontrollable and unobservable modes to the decentralized case. They show that the process of decentralization introduces *fixed modes* which cannot be changed through feedback. These fixed modes must be stable in order to stabilize the system using decentralized controllers. They further provide a procedure for constructing a set of stabilizing feedback control laws using decentralized pole placement. This work was furthered by Davison [1976] which shows that under mild conditions, a general robust decentralized servomechanism problem always has solution regardless of how the controllers are interconnected.

A great deal of work came after these first fundamental results continuing to the present. Decentralized control for linear systems is generally applied in the area of *Large Scale Systems* (e.g., large flexible space structures) where centralized control is impractical or impossible.

### 3.1.2 Centralized Control of Markov Decision Processes

The history of this particular area is elaborate and is perhaps best described from the beginning which was in the area of single agent or centralized control. The work on completely observable and partially observable systems will be described first and then we will jump back to the point along this timeline that decentralized work began and carry this forward to the present.

Bellman [1957] laid the foundations for work in *Markov decision processes* (MDPs). As in Markov chains, Markov decision processes usually use a discrete state space to describe the system to be controlled. This state changes probabilistically over discrete-time.

$$\boldsymbol{x}[t+1] = \boldsymbol{A}(\boldsymbol{x}[t], \boldsymbol{u}[t])$$

where $\boldsymbol{x} \in {}^{m}\mathbb{S}, \boldsymbol{u} \in {}^{n}\mathbb{S}, \boldsymbol{A} \in {}^{m}\mathbb{S}^{mn}$. Note, $(\boldsymbol{x}, \boldsymbol{u}) \in {}^{mn}\mathbb{S}$ is the joint probability distribution defined in the previous chapter. The difference between Markov chains and MDPs is that the latter allows control inputs, $\boldsymbol{u}[t]$, a probability distribution over discrete actions which may influence the future behaviour of the system. Furthermore, there is a set of rewards (which are a function of the current state, previous action, previous state) which are to be maximized (according to different measures) by the choice of actions. Thus, an MDP is actually an *optimal control problem.* It should be noted that MDPs can also work with continuous state spaces, continuous time and continuous actions but this is less common. Bellman showed that a stationary optimal policy always exists when the control law can be based directly on the state of the system

$$\boldsymbol{u} = \boldsymbol{F}\boldsymbol{x}$$

where $\boldsymbol{F} \in {}^{n}\mathbb{S}^{m}$. In fact, for each state of the system, there is one best (or a finite number of equally good) action(s) to take.

This result was extended to the case of continuous state and action spaces by Howard [1960] not long after. Still, however, the control was based directly on the complete system state; this is generally called a *completely observable* system.

Drake [1962] pointed out the fact that it was unrealistic in real systems to have direct access to the system state and looked at observing the state of an MDP through a noisy channel.

$$\begin{aligned} \boldsymbol{x}[t+1] &= \boldsymbol{A}(\boldsymbol{x}[t], \boldsymbol{u}[t]) \\ \boldsymbol{y}[t] &= \boldsymbol{C}\boldsymbol{x}[t] \end{aligned}$$

where $\boldsymbol{y} \in {}^{p}\mathbb{S}$, $\boldsymbol{C} \in {}^{p}\mathbb{S}^{m}$. The control law was based on the noisy observation, $\boldsymbol{y}$, rather than the system state, $\boldsymbol{x}$. This type of system is now commonly referred to as a *Partially Observable Markov Decision Process* (POMDP). Åstrom [1965] showed that a discrete-state POMDP could be converted to an MDP with continuous state space. This meant that in principle it could be solved as Howard [1960] had shown the converted problem to be solvable. Åstrom [1965] was able to solve a small POMDP using dynamic programming (value iteration).

Smallwood and Sondik [1973] were the first to notice that if there were a finite number of time-steps remaining in the control problem, the optimal payoff function is a piecewise-linear, convex function of the current state estimate of the system. They derive a method to calculate the optimal control policy for any finite horizon problem. Their method falls into the category of *policy iteration* which means that each iteration of the algorithm produces a better policy than the previous one. This is in contrast to *value iteration* which works to successively better approximate the optimal payoff functions for following a particular action from a particular state estimate. Sondik [1973] extended the previous work to consider the case of an infinite horizon but was unable to prove conclusively whether this problem can be solved in general (this has been looked at again more recently and is discussed below).

Papadimitriou and Tsitsiklis [1987] put forth an important paper describing some of the complexity results for MDPs and POMDPs. They find that MDPs with stochastic transitions are P-complete which means that they could not be solved by highly parallel algorithms. However, when the transitions are deterministic, MDPs could be solved very quickly in parallel. This implies that the stochastic nature of these problems introduces a much deeper complexity than expected. They further show that POMDPs are PSPACE-complete which means they are even harder to solve than NP-complete problems (e.g., the travelling salesman problem). These results were fundamental in guiding future research in this field as they illuminated for the first time the boundaries of computational tractability.

Ramadge and Wonham [1987] introduced *discrete-event systems* (DES) and proved a number of control theoretic results. These are not Markov processes exactly but they are related and there have even been some decentralized control results proved in this framework. They have states and transitions between them but do not assign probabilities to these transitions. Instead they are labelled either controllable or not and the idea to make sure the system can always get to a marked goal state.

All of the work detailed to this point has assumed that the transition matrix, $\boldsymbol{A}$, containing the conditional probabilities of the world moving from one state to another, is known. Some techniques, however, do not require this matrix to be known ahead of time. Watkins [1989], Watkins and Dayan [1992] proposed a now famous method known as *Q-learning* which falls into the category of reinforcement learning. The agent or planner is given the current state of the world as well as the reward signal. It learns online and has been shown under certain conditions to converge to the correct solutions. In fact, in practice it often works fairly well even when these conditions are violated (within reason). Schmidhuber [1991] also looked at reinforcement learning for Markovian as well as non-Markovian domains. The advantage of reinforcement learning methods is that complicated models of the physical system being controlled need not be constructed nor approximated. They may be learned online instead. Chrisman [1992] and McCallum [1992] both looked at learning a solution to a POMDP within the reinforcement learning framework. [Barto et al., 1995] is one of the best references on generalized reinforcement learning methods.

A group of researchers at Brown University took up the investigation of POMDPs starting in the mid

1990s. Most of this work focussed on creating algorithms which could solve larger POMDPs (more states and actions) in more computationally efficient ways. Littman [1994] looked at the limitations of using memoryless policies (e.g., reactive control) in non-Markovian environments. He showed how memory was able to circumvent the problem of partial observability. Cassandra et al. [1994] and Kaelbling et al. [1998] put forth the *Witness algorithm* which was an approximate value iteration method. Still, it was limited to problems with about 16 states. A good deal of work in this area turned to approximations in order to combat the results of Papadimitriou and Tsitsiklis [1987]. Littman et al. [1995b] provided a more detailed look at some of the complexity results for MDPs in order to help further guide the search for efficient, practical algorithms for these problems. Littman et al. [1995a] reviewed the current POMDP research and suggested ways of combining some techniques in order to scale up to larger, realistic problems. Parr and Russell [1995] introduced *Smooth Partially Observable Value Approximation* (SPOVA) to approximately solve POMDPs. Their technique can be combined with reinforcement learning methods. Cassandra et al. [1996] used POMDP techniques to deal with navigation of a mobile robot under uncertainty. Cassandra et al. [1997] proposed *Incremental Pruning*, a fast exact method for solving POMDPs and showed it outperformed all previous exact methods on a set of benchmark problems. Cassandra [1998] provided exact and approximate algorithms for POMDPs and is an excellent introduction to the field. [Hansen, 1998a] is also a valuable reference for policy-iteration techniques.

Zhang and Liu [1997] proposed to transform a POMDP into another POMDP which would be easier to solve because an "oracle" provided additional information about the world. The optimal policy computed for the easier problem could then be modified for the original one. Hansen et al. [1997, 1998b] put forth an improved policy iteration algorithm for POMDPs which used a finite-state controller representation. At each iteration of his algorithm, a successively better finite-state controller was produced. His algorithms are able to handle problems with 100 to 1000 states. Thrun et al. [1999] proposed using Monte Carlo methods to solve POMDPs with continuous state and action spaces. Baxter et al. [1999] used gradient methods to solve parametrized versions of large POMDPs. Peshkin et al. [1999] looked at solving POMDPs by using an external memory which agents can write to or read from in order to deal with non-Markovian states.

Madami et al. [1999] revisited the issue of whether infinite horizon POMDPs are decidable or not. Their result states that given an infinite horizon POMDP, it is not possible to state whether or not a policy exists that will succeed with probability better than some threshold. Goldsmith and Mundhenk [1998] and other researchers at the University of Kentucky have produced a large number of results on complexity of MDPs and POMDPs.

Currently there are several techniques for solving both centralized MDPs and centralized POMDPs exactly and approximately. The approximate techniques have gained popularity due to various complexity results showing how hard it will be to scale up the exact methods to realistic size problems. The largest POMDPs which can be dealt with are on the order of 1000 states but this requires knowing the model a priori and requires a great deal of computation. It seems that in general value iteration works better on small problems but policy iteration takes over and begins to become more efficient at finding approximate solutions as the problems become more complex (more states, actions). A similar phenomenon is prevalent in nature. Genetics (policy iteration) seems to solve the large problems while individual learning (value iteration) is able to deal with smaller, fine tuning problems.

This long list of centralized control results has been included mainly to make the point that when the system is partially observable, even when the model is known perfectly, it is extremely burdensome to use POMDP algorithms for the centralized control case. It is certainly even worse in the decentralized case. Our focus will be to suggest avenues that may make the decentralized case only as bad as the centralized case, not worse. Still, we find that for a system that will be implemented in the real world (e.g., a robotics system), POMDP algorithms are currently not practical (but may be so in the future).

### 3.1.3 Decentralized Control of Markov Decision Processes

We now jump back in time to describe the start of research in decentralized control of MDPs and POMDPs as well as other related models. Some of the earliest work on decentralized decision making comes from the control literature. Wheeler and Narenda [1985, 1986] proposed various models for decentralized decision making under uncertainty (an MDP being a specific case). They stress that "any study of interacting decision makers inevitably entails game-theoretic issues". Tsitsiklis and Athans [1985] analyzed the complexity of decentralized decision making and detection problems. They found that for a "team decision problem" (similar to an MDP) with 3 or more decision makers is NP-complete. In fact, a polynomial-time algorithm only exists for 2 decision makers with 2 action choices each. More decision makers or action choices makes the problem NP-complete (no polynomial time algorithm exists).

Aicardi et al. [1987] looked at decentralized control in the POMDP framework where the decision makers share their past observations and actions with a delay of $k$ time-steps. It was shown that a dynamic programming procedure can be applied to this situation to come up with a solution. Aicardi [1995] looked at a control theoretic approach to coordination of a team of mobile robots.

Zhang [1994] introduced *decision networks* which are a generalization of MDPs and influence diagrams. He looks at decentralized coordination in this framework by using a separability property in the reward function.

Schmidhuber [1996a, 1996b, 1997] proposed various high level learning methods for "realistic multi-agent reinforcement learning". These methods are approximate schemes that work like an overseer to other learning methods, to make sure they were progressing adequately. They are somewhat ad hoc and do not provide any great insight as to what is really going on in a decentralized POMDP. Ono and Fukumoto [1997] attempted to build a reinforcement learning algorithm that will scale to more agents and bigger worlds by imposing a radius of influence on each agent (outside of which they did not make observations). In a small predator-prey simulation they showed that reactive agents can still learn good policies despite their partial observability. Versino and Gambardella [1997] analyzed various decentralized reinforcement learning paradigms (e.g., private vs. public policies, homogeneous vs. heterogeneous agents, the credit assignment problem) in the context of a simple simulation.

Boutilier [1996] proposed *Multiagent MDPs* (MMDPs) in which each agent has access to the full state and the idea is to get the agents to make good group decisions.

$$\boldsymbol{x}[t+1] = \boldsymbol{A}(\boldsymbol{x}[t], \boldsymbol{u}_k[t]) \quad \forall k = 1 \ldots K$$

where $\boldsymbol{x} \in {}^{m}\mathbb{S}$, $\boldsymbol{u}_k \in {}^{n_k}\mathbb{S}\ \forall k = 1 \ldots K$, $\boldsymbol{A} \in {}^{m}\mathbb{S}^{mn}$ where $n = \prod_{k=1}^{K} n_k$. This work builds on Bellman's result that for every state there is a finite set (minimum size of one) of equally good centralized actions

to take. The main problem addressed is how to coordinate the decentralized actions, $\boldsymbol{u}_k$, when there are more than one equally good centralized actions. Various coordination mechanisms are discussed. This work is very relevant to the later chapter on *self-organization* in this document.

Claus and Boutilier [1997] discussed the dynamics of decentralized reinforcement learning on coordination games. They discuss game-theoretic issues and stability of solutions. Wessels [1997] attempted to formally introduce multiagent reinforcement learning techniques which use average rewards and stochastic policies to deal with partial observability. Ooi et al. [1997] looked at decentralized control of POMDPs where the decision makers share their information with one-step delay every $k$ time-steps. They show a useful separation property in this situation such that control laws may be designed independently for each decision maker.

Sun and Peterson [1999] described multiagent reinforcement learning heuristics which partition a global task and use agents selectively based on that partitioning. They find some of their techniques outperform single agent monolithic approaches to learning for groups of agents (centralized approaches). desJardins et al. [1999] and Boutilier [1999] provided excellent overviews of multiagent research and outline future challenges and opportunities in this area.

Berstein et al. [2000] introduced *Decentralized POMDPs* (DecPOMDPs) which are similar to models used by previous researchers. They combine the decentralized control of MMDPs with the partial observability of POMDPs.

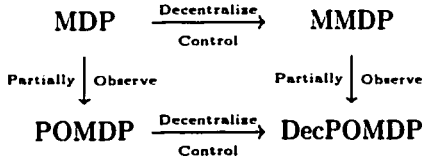
$$\begin{aligned} \boldsymbol{x}[t+1] &= \boldsymbol{A}(\boldsymbol{x}[t], \boldsymbol{u}_k[t]) \;\; \forall k = 1 \ldots K \\ \boldsymbol{y}_k[t] &= \boldsymbol{C}_k \boldsymbol{x}[t] \end{aligned}$$

where $\boldsymbol{x} \in {}^{m}\mathbb{S}$, $\boldsymbol{u}_k \in {}^{n_k}\mathbb{S}$, $\boldsymbol{y}_k \in {}^{p_k}\mathbb{S}$, $\boldsymbol{C}_k \in {}^{p_k}\mathbb{S}^m \; \forall k = 1 \ldots K$ and $\boldsymbol{A} \in {}^{m}\mathbb{S}^{mn}$ where $n = \prod_{k=1}^{K} n_k$. The decentralized controls, $\boldsymbol{u}_k$, must be based on the observations, $\boldsymbol{y}_k$, not the full system state $\boldsymbol{x}$. They extend the complexity results of Tsitsiklis and Athans [1985] to their model and show that it would extremely difficult (doubly exponential time) to solve these types of problems. They hope these results will guide researchers to appropriate areas of approximate solution (advice which is certainly heeded in this thesis).

Peshkin et al. [2000] looked at a decentralized version of POMDPs in which decision makers may have different views of the world but have a common reward signal (or goal). They use gradient descent in a parametrized policy space to come up with solutions and demonstrate their technique on a two-agent soccer environment.

There is not a great deal of theoretical work in the area of decentralized planning or control (of Markov systems) although Weiss [1999] is a good summary of work in the general field of multiagent systems. There are, however, countless examples of researchers applying variants of reinforcement learning, genetic techniques and other heuristics to simulations and physical systems.

The relationships between the various Markov decision models described above may be drawn as follows.



It would be more consistent if MMDP were renamed to DecMDP but that name has been used to mean something else [Berstein et al., 2000]. Alternatively, perhaps DecPOMDP should be renamed to MPOMDP. To avoid confusion we will keep with the names in the diagram.

Certainly decentralized systems are being looked at from a number of very different perspectives. It is hoped this review has been able to provide a broad view of current efforts and past accomplishments. There is a great deal more out there but these are examples are representative in their complexity and depth.

## 3.2 Statistical Dynamics

It will be useful to touch on an important point concerning stochastic difference equations as they pertain to all the statistical dynamics models considered here. A difference equation (with discrete time, $t$) of the form

$$\boldsymbol{x}[t+1] = \boldsymbol{f}(\boldsymbol{x}[t])$$

where $\boldsymbol{x} \in {}^{n}\mathbb{S}$ and $\boldsymbol{f}(\boldsymbol{x}) \in {}^{n}\mathbb{F}^{n}$ may be considered on two quite different levels. We must recall that $\boldsymbol{x}$ is really a probability distribution for the random variable, $X$. This distribution describes with what probability each of $n$ discrete states, $X_i \; \forall i = 1 \ldots n$, is occupied. The stochastic function, $\boldsymbol{f}(\boldsymbol{x}) \in {}^{n}\mathbb{F}^{n}$, describes the probability of each state at the new time-step, given the probabilities at the old time-step. It is a *transition function*.

On one level, we may think of the system occupying a specific state, $\mathbf{x} \in {}^{n}\mathbb{D}$, at each time-step with probability 1. The transition function comes into play only at the instant of transition. A single *trajectory* of specific states is the result. The second level on which we must consider difference equations is the *ensemble*. Here all possible trajectories are considered simultaneously and the state vector, $\boldsymbol{x} \in {}^{n}\mathbb{S}$, now indicates with what probability each state is occupied. It is essentially an average over all possible trajectories. This concept is used very frequently in such fields as statistical mechanics. We will refer to these two different modes of analysis as *trajectory* and *ensemble* and will distinguish between them as necessary. There is a nice property of ensemble analysis. Namely that if we wish to evaluate the state vector (probability distribution) at some time $s$ steps into the future we may do so as follows

$$\boldsymbol{x}[t+s] = \underbrace{\boldsymbol{f}\Big(\boldsymbol{f}\big(\cdots \boldsymbol{f}(\boldsymbol{x}[t]) \cdots\big)\Big)}_{s \text{ compositions}} = \boldsymbol{f}^{s}(\boldsymbol{x}[t])$$

where we simply take compositions of the function, $\boldsymbol{f}(\boldsymbol{x})$. Throughout much of this chapter we will be using ensemble analysis as it allows us to make sure the system behaves properly on average, not just by good fortune in a single specific instance. However, it will sometimes be necessary to discuss both paradigms.

## 3.3 Markov Decision Problems

Given the results of Berstein et al. [2000] and Tsitsiklis and Athans [1985] it would not be productive to plunge directly into creating solutions for the DecPOMDP model. Even if it is theoretically possible to

come up with optimal controllers, it does not mean they will be either practical, scalable to a realistic number of states, or even interesting. We will instead investigate the effect decentralization has on this type of system in general. What types of limitations does it impose? When can decentralized controls perform as well as a centralized one? What can be done to combat the effects of decentralization? These fundamental questions are those with which we will be concerned. We will argue that decentralized controllers that communicate may perform as well as a centralized controller. We first must set up the DecPOMDP model in stochastic algebra[2].

## 3.3.1 Projection Matrices

*Projection matrices*[3] are used to move between global (centralized) and local (decentralized) versions of variables. It should be pointed out that we will be assuming that there are $K$ sensor/actuator clusters. It would be more general to allow the number of clusters of sensors to be different than the number of clusters of actuators but this will not be considered here.

### Control

*Control projection matrices* are used to move between the global and local versions of the control variables. They are deterministic matrices but may also be considered to be nonlinear stochastic functions which act on the global control. They are of the form

$$\mathbf{B}_k = [b_{ij}] = [p(w_i|u_j)] \in {}^{m_k}\mathbb{F}^m : {}^m\mathbb{S} \mapsto {}^{m_k}\mathbb{S} \tag{3.5}$$

where $k = 1 \ldots K$ and $m = \prod_{k=1}^{K} m_k$. Given some global control, $\boldsymbol{u} \in {}^m\mathbb{S}$ , the local controls, $\boldsymbol{w}_k \in {}^{m_k}\mathbb{S}$, are given by

$$\begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{w}_K \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_K \end{bmatrix} \boldsymbol{u} \tag{3.6}$$

where $K$ is the number of decentralized controllers, and the $\mathbf{B}_k$, are the control projection matrices. In general, the control projection matrices could be stochastic but here we will consider them to be deterministic. Furthermore, to ensure the local controls are able to construct all the possible global controls we require that

$$\mathbf{B}_1^T\mathbf{B}_1 \oplus \mathbf{B}_2^T\mathbf{B}_2 \oplus \cdots \oplus \mathbf{B}_K^T\mathbf{B}_K = \mathbf{1} \tag{3.7}$$

To ensure the local controls are independent this identity should never be satisfied if any of the terms are removed. This requires that

$$\mathbf{B}_i^T\mathbf{B}_j = \mathbf{\Omega} \qquad i \neq j \tag{3.8}$$

[2]Although uncontrolled Markov chains have been studied using stochastic matrices, it seems that Markov control models typically are not. The stochastic algebra described in the previous chapter makes this possible.

[3]These are not the *projection matrices* described in the chapter on Stochastic Algebra but their function is similar enough to warrant the term, which will be used throughout this section.

If the global control is a statistically independent joint probability distribution of the local controls[4], then we have

$$\begin{aligned} \boldsymbol{u} &= \mathbf{1}\boldsymbol{u} \\ &= (\mathbf{B}_1^{\mathcal{T}}\mathbf{B}_1 \oplus \cdots \oplus \mathbf{B}_K^{\mathcal{T}}\mathbf{B}_K)\boldsymbol{u} \\ &= \mathbf{B}_1^{\mathcal{T}}\mathbf{B}_1\boldsymbol{u} \oplus \cdots \oplus \mathbf{B}_K^{\mathcal{T}}\mathbf{B}_K\boldsymbol{u} \\ &= \mathbf{B}_1^{\mathcal{T}}\boldsymbol{w}_1 \oplus \cdots \oplus \mathbf{B}_K^{\mathcal{T}}\boldsymbol{w}_K \end{aligned} \tag{3.9}$$

This allows us to go from the local controls to the global control. It is essentially (3.6) subject to the constraint that the local controls are independent. Versions of this equation can be used to combine any two statistically independent local variables (individual probability distributions) into a global variable (joint probability distribution).

### State-Observation

*State-observation projection matrices* are used to move between the global state of the system and local observations of that state. For now we consider the state-observation projection matrices to be deterministic but will later relax this condition.

$$\mathbf{C}_k = [c_{ij}] = [p(y_i|x_j)] \in {}^{p_k}\mathbb{F}^n : {}^n\mathbb{S} \mapsto {}^{p_k}\mathbb{S} \tag{3.10}$$

where $k = 1 \ldots K$ and $n \geq p_k$. Given some global state, $\boldsymbol{x} \in {}^n\mathbb{S}$ , the local observations, $\boldsymbol{y}_k \in {}^{p_k}\mathbb{S}$, are given by

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_K \end{bmatrix} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \vdots \\ \mathbf{C}_K \end{bmatrix} \boldsymbol{x} \tag{3.11}$$

where $K$ is the number of decentralized controllers, and the $\mathbf{C}_k$, are the state-observation projection matrices. In general, the state-observation projection matrices could be stochastic but here we will consider them to be deterministic. Note, we might choose that by combining all information available to all controllers would result in perfect observations of the state or

$$\mathbf{C}_1^{\mathcal{T}}\mathbf{C}_1 \oplus \mathbf{C}_2^{\mathcal{T}}\mathbf{C}_2 \oplus \cdots \oplus \mathbf{C}_K^{\mathcal{T}}\mathbf{C}_K \equiv \mathbf{1} \tag{3.12}$$

This case is equivalent to the DecMDP framework described in [Berstein et al., 2000]. If (3.12) does not hold then in general we have that

$$\begin{aligned} \hat{\boldsymbol{x}} &= (\mathbf{C}_1^{\mathcal{T}}\mathbf{C}_1 \oplus \cdots \oplus \mathbf{C}_K^{\mathcal{T}}\mathbf{C}_K)\mathbf{x} && (3.13) \\ &= \mathbf{C}_1^{\mathcal{T}}\mathbf{C}_1\boldsymbol{x} \oplus \cdots \oplus \mathbf{C}_K^{\mathcal{T}}\mathbf{C}_K\boldsymbol{x} \\ &= \mathbf{C}_1^{\mathcal{T}}\boldsymbol{y}_1 \oplus \cdots \oplus \mathbf{C}_K^{\mathcal{T}}\boldsymbol{y}_K && (3.14) \\ &\neq \boldsymbol{x} \end{aligned}$$

[4] Recall example from the chapter on stochastic algebra.

Notice that there is no unique inverse to (3.11) as a result of (3.12) not being true. Instead, we get an estimate, $\hat{x}$, of the state which is no longer deterministic.

Note, in the literature, the state-observation projection matrices (or their equivalents) sometimes are a function of both the global state and the global control. We find the model to be rich enough without this added complication.

#### Control-Observation

*Control-observation projection matrices* are used to move between the global control of the system and the local observations of that control.

$$\mathbf{D}_k = [d_{ij}] = [p(v_i|u_j)] \in {}^{q_k}\mathbb{F}^m : {}^m\mathbb{S} \mapsto {}^{q_k}\mathbb{S} \tag{3.15}$$

where $k = 1 \ldots K$ and $m \geq q_k$. Given some global control, $\boldsymbol{u} \in {}^m\mathbb{S}$ , the local observations, $\boldsymbol{v}_k \in {}^{q_k}\mathbb{S}$, are given by

$$\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_K \end{bmatrix} = \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \\ \vdots \\ \mathbf{D}_K \end{bmatrix} \boldsymbol{u} \tag{3.16}$$

where $K$ is the number of decentralized controllers, and the $\mathbf{D}_k$, are the control-observation projection matrices. Often in centralized control literature, it is assumed that the controller has access to past controls. However, in decentralized control, this is a dangerous assumption as one controller may not be able to observe which controls the other controllers have effected. The control-observation projection matrices have functionally similar equations to those of the state-observation projection matrices.

### 3.3.2 Transition Matrix

The transition matrix works in a slightly different way than the other system matrices. It must depend on both the global system state, $\boldsymbol{x}$, as well as the global system control, $\boldsymbol{u}$.

$$\boldsymbol{A} = [a_{ij}] = [p(x_i|x_l, u_r)] \in {}^n\mathbb{F}^{mn} : {}^{mn}\mathbb{S} \mapsto {}^n\mathbb{S} \tag{3.17}$$

It describes how the system unfolds over time. Given some global state, $\boldsymbol{x}[t]$, and global control $\boldsymbol{u}[t]$, we have

$$\boldsymbol{x}[t+1] = \boldsymbol{A}(\boldsymbol{x}[t], \boldsymbol{u}[t]) \tag{3.18}$$

where $(\boldsymbol{x}[t], \boldsymbol{u}[t])$ is the (statistically independent) joint probability distribution constructed from $\boldsymbol{x}$ and $\boldsymbol{u}$. We may also express this difference equation in the form

$$\boldsymbol{x}[t+1] = \left[\boldsymbol{A}_1 \boldsymbol{A}_2 \cdots \boldsymbol{A}_m\right](\boldsymbol{x}[t], \boldsymbol{u}[t]) \tag{3.19}$$

where the $\boldsymbol{A}_i \in {}^n\mathbb{S}^n \; \forall i = 1 \ldots n$ are individual transition matrices for each of the $m$ global controls. In general, the transition matrix is stochastic.

### 3.3.3 The DecPOMDP Model

It is useful at this juncture to summarize the equations which constitute the DecPOMDP model [Berstein et al., 2000]. These are essentially (3.19), (3.9), (3.11), and (3.16) written more succinctly.

**Definition.** DECPOMDP: A *Decentralized Partially Observable Markov Decision Process* is described by the system of stochastic equations

$$\begin{aligned} \boldsymbol{x}[t+1] &= \left[\boldsymbol{A}_1 \boldsymbol{A}_2 \cdots \boldsymbol{A}_m\right](\boldsymbol{x}[t], \boldsymbol{u}[t]) && (3.20)\\ \boldsymbol{u}[t] &= \bigoplus_{k=1}^{K} \mathbf{B}_k^T \boldsymbol{w}_k[t] && (3.21)\\ \boldsymbol{y}_k[t] &= \mathbf{C}_k \boldsymbol{x}[t] \quad \forall k = 1 \ldots K && (3.22)\\ \boldsymbol{v}_k[t] &= \mathbf{D}_k \boldsymbol{u}[t] \quad \forall k = 1 \ldots K && (3.23) \end{aligned}$$

where $\boldsymbol{x} \in {}^{n}\mathbb{S}$ and $\boldsymbol{A}_i \in {}^{n}\mathbb{S}^{n}$ $\forall i = 1 \ldots n$ with $n$, the number of global states of the system. $\boldsymbol{u} \in {}^{m}\mathbb{S}$ with $m$, the number of possible global controls. $\boldsymbol{w}_k \in {}^{m_k}\mathbb{S}$, $\boldsymbol{y}_k \in {}^{p_k}\mathbb{S}$, $\boldsymbol{v}_k \in {}^{q_k}\mathbb{S}$, $\mathbf{B}_k \in {}^{m_k}\mathbb{D}^{m}$, $\mathbf{C}_k \in {}^{p_k}\mathbb{D}^{n}$, $\mathbf{D} \in {}^{q_k}\mathbb{D}^{m}$ $\forall k = 1 \ldots K$. $K$ is the number of agents or decentralized controllers for the system. Thus, the tuple[5]

$$\left\{\boldsymbol{A}_i, \mathbf{B}_k, \mathbf{C}_k, \mathbf{D}_k\right\} \quad \forall i = 1 \ldots m, \forall k = 1 \ldots K \tag{3.24}$$

uniquely characterizes a DecPOMDP.

In our stochastic algebra, this is a nonlinear system due to the structure of the transition matrix. Given a DecPOMDP, the goal is to come up with controllers which produce appropriate behaviour. What constitutes appropriate behaviour will be discussed later. This brings us to another definition.

**Definition.** DECPOMDP SOLUTION: A solution to a DecPOMDP consists of a set of $K$ controllers

$$\left\{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K\right\}$$

where $\boldsymbol{h}_k \in {}^{m_k}\mathbb{F}$ $\forall k = 1 \ldots K$ which produce local controls, $\boldsymbol{w}_k$, given the past local observations of both the global system state, $\boldsymbol{y}_k$, and the global system control, $\boldsymbol{v}_k$.

$$\mathbf{w}_k[t] = \boldsymbol{h}_k \begin{pmatrix} \boldsymbol{h}_1, \ldots, \boldsymbol{h}_K, \\ \mathbf{y}_k[t], \ldots, \mathbf{y}_k[0], \\ \mathbf{v}_k[t-1], \ldots, \mathbf{v}_k[0] \end{pmatrix} \tag{3.25}$$

Notice that each controller is a function of the other controllers. This is meant to indicate one type of coupling that occurs between controllers which will be explained shortly.

### 3.3.4 Separation

There is fundamental result which is often exploited in centralized control, namely separation of state estimator design and control law design. It allows one to design a rule which estimates the global system

[5]The word "tuple" will be used to describe a group (of arbitrary size) of associated matrices. It is derived from quin*tuple*, sex*tuple*, sep*tuple*, oc*tuple*, ...

state, $\boldsymbol{x}[t]$, independently of the control law, which produces controls based on this state estimate such that

$$\hat{\boldsymbol{x}}[t] = \boldsymbol{h}_{\text{se}}\begin{pmatrix} \boldsymbol{y}[t],\ldots,\boldsymbol{y}[0], \\ \boldsymbol{u}[t-1],\ldots,\boldsymbol{u}[0] \end{pmatrix} \tag{3.26}$$

$$\boldsymbol{u}[t] = \boldsymbol{h}_{\text{con}}\left(\hat{\boldsymbol{x}}[t]\right) \tag{3.27}$$

where $\boldsymbol{h}_{\text{se}} \in {}^{n}\mathbb{F}$ is the state estimator and $\boldsymbol{h}_{\text{con}} \in {}^{m}\mathbb{F}$ is the control law.

**Theorem 1.** *In centralized control, the design of a state estimator (observer) may be carried out independently of the design of a control law.*

**Proof.** This was shown by Åstrom [1965] for POMDPs. ■

This relies on the fact that the past global controls, $\mathbf{u}[t-1],\ldots,\mathbf{u}[0]$, are known perfectly which is usually a good assumption for centralized control.

However, for decentralized control this is not necessarily so. While a controller may have perfect recall of its own past local controls, $\boldsymbol{w}_k[t-1],\ldots,\boldsymbol{w}_k[0]$, it may not have access to the past local controls of the other $K-1$ controllers. This is why the *control-observation matrices* are in the DecPOMDP model. A *control estimator* may also be necessary to estimate what global controls were used in the past. Furthermore, the control estimator can be a function of the control laws of the other agents (one can guess the controls of the other agents better if one has access to their control laws). In decentralized control, both a state estimator and a control estimator are required in the most general case. Furthermore, the design of the control estimator is a function of the local control laws of all $K$ controllers so that

$$\hat{\boldsymbol{u}}_k[t] = \boldsymbol{h}_{\text{ce},k}\begin{pmatrix} \boldsymbol{h}_{\text{con},1},\ldots,\boldsymbol{h}_{\text{con},K} \\ \hat{\boldsymbol{x}}_k[t], \\ \mathbf{v}_k[t],\ldots,\mathbf{v}_k[0] \end{pmatrix} \tag{3.28}$$

$$\hat{\boldsymbol{x}}_k[t] = \boldsymbol{h}_{\text{se},k}\begin{pmatrix} \boldsymbol{y}_k[t],\ldots,\mathbf{y}_k[0], \\ \hat{\boldsymbol{u}}_k[t-1] \end{pmatrix} \tag{3.29}$$

$$\mathbf{w}_k[t] = \boldsymbol{h}_{\text{con},k}\left(\hat{\boldsymbol{x}}_k[t]\right) \tag{3.30}$$

where $\boldsymbol{h}_{\text{ce},k} \in {}^{m}\mathbb{F}$ is the control estimator, $\boldsymbol{h}_{\text{se},k} \in {}^{n}\mathbb{S}$ is the state estimator and, $\boldsymbol{h}_{\text{con},k} \in {}^{m}\mathbb{F}$ is the control law $\forall k = 1\ldots K$.Thus in decentralized control, the design of the state estimators is coupled to the design of controllers through the control estimators. This is because each controller does not have access to the past controls of the other $K-1$ controllers and that a better estimate of this unknown information may be achieved with knowledge of the other control laws. This coupling makes the general decentralized case much more difficult than simply solving $K$ problems of the centralized difficulty. There are, however, a number of simplifications which can be made to recover the separation property.

The first simplification which regains separation is to assume that the control estimators do not have access to the control laws.In decentralized control, when control estimators do not have access to control

laws, we have

$$\hat{\boldsymbol{u}}_k[t] = \boldsymbol{h}_{\mathrm{ce},k}\left(\begin{array}{l}\hat{\boldsymbol{x}}_k[t],\\ \mathbf{v}_k[t],\ldots,\mathbf{v}_k[0]\end{array}\right) \tag{3.31}$$

$$\hat{\boldsymbol{x}}_k[t] = \boldsymbol{h}_{\mathrm{se},k}\left(\begin{array}{l}\mathbf{y}_k[t],\ldots,\mathbf{y}_k[0],\\ \hat{\boldsymbol{u}}_k[t-1]\end{array}\right) \tag{3.32}$$

$$\mathbf{w}_k[t] = \boldsymbol{h}_{\mathrm{con},k}\left(\hat{\boldsymbol{x}}_k[t]\right) \tag{3.33}$$

where we see that separation of the design of control laws, state estimators, and control estimators results.

Another simplification which regains separation is to assume that the controllers are able to perfectly observe each others past controls, possibly through communication.In decentralized control, when controllers have perfect access to the past controls of the other controllers we have

$$\mathbf{D}_1 = \mathbf{D}_2 = \cdots = \mathbf{D}_K = \mathbf{1} \tag{3.34}$$

rendering the control estimator unnecessary. We then have

$$\hat{\boldsymbol{x}}_k[t] = \boldsymbol{h}_{\mathrm{se},k}\left(\begin{array}{l}\mathbf{y}_k[t],\ldots,\mathbf{y}_k[0],\\ \hat{\boldsymbol{u}}_k[t-1]\end{array}\right) \tag{3.35}$$

$$\mathbf{w}_k[t] = \boldsymbol{h}_{\mathrm{con},k}\left(\hat{\boldsymbol{x}}_k[t]\right) \tag{3.36}$$

where we see that separation of the design of control laws and state estimators results.

Another simplification which can be made is that only *reactive* control is to be used. Reactive control refers to a controller only using its most recent observation in the state estimate.In decentralized reactive control, the control estimator is unnecessary and we have

$$\hat{\boldsymbol{x}}_k[t] = \boldsymbol{h}_{\mathrm{se},k}\left(\mathbf{y}_k[t]\right) \tag{3.37}$$

$$\boldsymbol{w}_k[t] = \boldsymbol{h}_{\mathrm{con},k}\left(\hat{\boldsymbol{x}}_k[t]\right) \tag{3.38}$$

where we see that separation of the design of control laws and state estimators results.

## 3.4 State Estimation

We first consider the case of centralized state estimation. The goal of state estimation is to come up with an estimate of the global state of the system, $\boldsymbol{x}[t]$ based on past observations and controls

$$\hat{\boldsymbol{x}}[t] = \boldsymbol{h}_{\mathrm{se}}\left(\begin{array}{l}\boldsymbol{y}[t],\ldots,\boldsymbol{y}[0],\\ \boldsymbol{u}[t-1],\ldots,\boldsymbol{u}[0]\end{array}\right) \tag{3.39}$$

In the stochastic algebra, a single-step (or static) state estimator can be constructed as

$$\hat{\boldsymbol{x}}[t] = \mathbf{C}^T\boldsymbol{y}[t] = \mathbf{C}^T\mathbf{C}\boldsymbol{x}[t] \tag{3.40}$$

where $\mathbf{C}^T\mathbf{C}$ may be thought of as a filter through which the true state is observed. As we can see, this estimator is based on (2.2), the ***uniformly biased inverse*** from Section 2.15, Page 34. This single

step state estimate may be combined with an old state estimate to arrive at a recursive state estimator [Åstrom, 1965]

$$\hat{\boldsymbol{x}}[t+1] = \boldsymbol{A}[t]\hat{\boldsymbol{x}}[t] \oplus \mathbf{C}^{\boldsymbol{r}}\mathbf{y}[t+1] \tag{3.41}$$

where $\boldsymbol{A}[t] = \boldsymbol{A}(\boldsymbol{u}[t])$. Notice the role of the past global control $\boldsymbol{u}[t]$ in determining $\boldsymbol{A}[t]$. Here the estimate at the old time has been brought forward in time using the transition matrix and becomes the bias for the estimate at the new time. We assume $\hat{\boldsymbol{x}}[0] = \boldsymbol{\omega}$. This is where the main difference between centralized state estimation and decentralized state estimation occurs. In the decentralized version we have $\boldsymbol{A}[t] = \boldsymbol{A}(\hat{\boldsymbol{u}}[t])$ where the past global control is now estimated as each controller does not necessarily know which controls were effected by the other $K - 1$ controllers.

## 3.5 Control Laws

Some discussion must be given to what constitutes appropriate behaviour of the system. Essentially there are two possibilities. First, we wish to use controls to get the system into a particular goal state. This is the problem of *controllability*. Second, there is some cumulative reward function over the states/controls which must be maximized. This is the question of *optimal control*. For now we assume the latter situation but some discussion will be given to controllability later. Thus assume there is a stationary reward function, $\rho(\mathbf{x}, \mathbf{u})$, over the states/controls

$$\rho : {}^{n}\mathbb{S} \times {}^{m}\mathbb{S} \mapsto \mathbb{R} \tag{3.42}$$

Furthermore, assume we are trying to minimize the infinite horizon cumulative discounted sum, $\rho_{tot}$, of this function

$$\begin{aligned} \rho_{tot} &= \rho(\boldsymbol{x}[0], \boldsymbol{u}[0]) + \gamma\rho(\boldsymbol{x}[1], \boldsymbol{u}[1]) + \cdots \\ &= \sum_{t=0}^{\infty} \gamma^t \rho(\boldsymbol{x}[t], \boldsymbol{u}[t]) \end{aligned} \tag{3.43}$$

where $\gamma \in [0, 1)$ is the future discount constant.

Bellman [1957] showed, using dynamic programming, that for the centralized case with $\mathbf{C} = \mathbf{1}$ (an MDP) that the optimal control law is stationary (time invariant) and that there is always one best (or a few equally good) actions to be taken from each state. The general centralized case ($\boldsymbol{C} \neq \mathbf{1}$; a POMDP) is much more complicated and involves transforming the problem into a continuous state MDP which is fully observable [Åstrom, 1965]. The continuous state MDP problem was solved by Howard [1960]. Recent results [Madani et al., 1999], however, have shown that the infinite-horizon POMDP case to be generally undecidable (impossible to prove a control law exists in the general POMDP case). Note there are other cumulative reward functions available including average cumulative reward and finite-horizon cumulative reward. This results in different types of solutions to the centralized problem. We will examine only the infinite-horizon case as it is easier to manage a stationary control law. The decentralized results to follow should extend to these other cases.

In the DecPOMDP framework, there are not currently any results proving the existence of decentralized control laws for an arbitrary problem. As mentioned in the introduction to this chapter, the

approach here will be to present situations where decentralized controllers can carry out identical global controls to a centralized controller. Thus, we may be able to capitalize on existence proofs for the centralized case. As we will see, this will involve communication between the decentralized controllers. The next section will treat the specific case of *reactive* control, where only the most recent observations are used in the control process.

## 3.6 Reactive Control

We consider the case of reactive control in some depth. Bellman's [1957] optimal stationary control law for the case of $\mathbf{C} = \mathbf{1}$ can be represented as

$$\boldsymbol{u}[t] = \boldsymbol{F}\mathbf{x}[t] \tag{3.44}$$

where $\boldsymbol{F} \in {}^{m}\mathbb{S}^{n}$ is the *control law matrix*. We have chosen to represent this matrix as generally stochastic as there can be more than one equally good action to follow from a particular state. This will have ramifications for decentralized control [Boutilier, 1996]. However, in every case of more than one equally good action we could choose one of these to follow only and the resulting control law, $\mathbf{F} \in {}^{m}\mathbb{D}^{n}$ would still be optimal

$$\mathbf{u}[t] = \mathbf{F}\mathbf{x}[t] \tag{3.45}$$

This will provide a useful simplification later but it should be noted that it may involve more work in the design of $\mathbf{F}$. Before moving on to decentralized control we make the observation that although it was assumed that $\mathbf{C} = \mathbf{1}$ to come up with this solution, there could exist $\boldsymbol{C} \in {}^{p}\mathbb{S}^{n} \neq \mathbf{1}$ that still allow $\boldsymbol{F}$ to be implemented perfectly. In this case, the control law must be based on a state estimate rather than the perfect state so that

$$\boldsymbol{u}[t] = \boldsymbol{F}\hat{\boldsymbol{x}}[t] \tag{3.46}$$

Combining (3.46) with (3.40) we arrive at

$$\boldsymbol{u}[t] = \boldsymbol{F}\boldsymbol{C}^{\tau}\boldsymbol{C}\mathbf{x}[t] \tag{3.47}$$

Comparing (3.46) with (3.44) we arrive at the following statement.

**Theorem 2.** *The tuple*

$$\{\boldsymbol{F}, \boldsymbol{C}\}$$

*is exactly implementable (in the POMDP case) if*

$$\boldsymbol{F} \equiv \boldsymbol{F}\boldsymbol{C}^{\tau}\boldsymbol{C} \tag{3.48}$$

*where $\boldsymbol{F} \in {}^{m}\mathbb{S}^{n}$ is the control law matrix (designed for the $\mathbf{C} = \mathbf{1}$ case) and $\boldsymbol{C} \in {}^{p}\mathbb{S}^{n}$ is the actual observation projection matrix.*

**Proof.** Substituting $\boldsymbol{F} = \boldsymbol{F}\boldsymbol{C}^{\tau}\boldsymbol{C}$ into (3.47) results in (3.44). ■

This condition is sufficient but not necessary as more complicated state estimators may help. Furthermore, $\boldsymbol{F}$ need not be an optimal control law when considering Theorem 2; it is a general result. It is simply saying that if two different actions are to be taken in two different states, the controller must be able to distinguish between those two states. States from which the same action is to be taken need not be distinguishable.

Consider now that there are $K$ clusters of sensors/actuators and that all sensor information is communicated to a centralized control law which in turn communicates controls to all actuators. In the framework presented here this may be represented by the following centralized control law

$$\begin{aligned} \mathbf{u}[t] &= (\boldsymbol{B}_1^T\boldsymbol{B}_1 \oplus \cdots \oplus \boldsymbol{B}_K^T\boldsymbol{B}_K)\,\boldsymbol{F}\,(\boldsymbol{C}_1^T\boldsymbol{C}_1 \oplus \cdots \oplus \boldsymbol{C}_K^T\boldsymbol{C}_K)\,\mathbf{x}[t] \\ &= \left(\bigoplus_{k=1}^{K}\boldsymbol{B}_k^T\boldsymbol{B}_k\right)\boldsymbol{F}\left(\bigoplus_{j=1}^{K}\boldsymbol{C}_j^T\boldsymbol{C}_j\right)\mathbf{x}[t] \end{aligned} \tag{3.49}$$

The extension to Theorem 2 is thus

**Theorem 3.** *The tuple*

$$\{\boldsymbol{B}_k, \boldsymbol{F}, \boldsymbol{C}_k\} \quad \forall k = 1 \ldots K$$

*is exactly implementable (in the POMDP case) if*

$$\boldsymbol{F} \equiv \left(\bigoplus_{k=1}^{K}\boldsymbol{B}_k^T\boldsymbol{B}_k\right)\boldsymbol{F}\left(\bigoplus_{j=1}^{K}\boldsymbol{C}_j^T\boldsymbol{C}_j\right) \tag{3.50}$$

*where $\boldsymbol{F} \in {}^{m}\mathbb{S}^{n}$ is the control law matrix (designed for the $\mathbf{C}_k = \mathbf{1}$ case), $\boldsymbol{B}_k \in {}^{m_k}\mathbb{S}^{m}$ are the action projection matrices, and $\boldsymbol{C}_k \in {}^{p_k}\mathbb{S}^{n}$ are the observation projection matrices, where $k = 1 \ldots K$ and $K$ is the number of controllers.*

**Proof.** Same as that for Theorem 2. ■

In the case that $\left(\boldsymbol{F}^T\right)^T = \boldsymbol{F}$, there is also a *dual* to Theorem 3.

**Corollary 3.1.** Given $\left(\boldsymbol{F}^T\right)^T = \boldsymbol{F}$, the tuple

$$\{\boldsymbol{B}_k, \boldsymbol{F}, \boldsymbol{C}_k\} \quad \forall k = 1 \ldots K$$

is implementable if and only if the tuple

$$\{\boldsymbol{C}_k, \boldsymbol{F}^T, \boldsymbol{B}_k\} \quad \forall k = 1 \ldots K$$

is implementable.

**Proof.** Based on (2.3), (2.4), (2.5), and (2.6). ■

### 3.6.1 Classes

Based on Theorem 3, we now define four distinct classes of global reactive controllers based on how much communication is necessary to implement them using decentralized controllers.

**Definition.** CLASS I: To implement these controllers neither *sensory sharing* nor *actuator coordination* is required. The tuple

$$\{\boldsymbol{B}_k, \boldsymbol{F}, \boldsymbol{C}_k\} \quad \forall k = 1 \ldots K$$

can be expressed in the form

$$\boldsymbol{F} = \bigoplus_{k=1}^{K} \boldsymbol{B}_k^T \boldsymbol{B}_k \boldsymbol{F} \boldsymbol{C}_k^T \boldsymbol{C}_k \tag{3.51}$$

where $\boldsymbol{F} \in {}^{m}\mathbb{S}^{n}$ is the control law matrix, $\boldsymbol{B}_k \in {}^{m_k}\mathbb{S}^{m}$ are the action projection matrices, and $\boldsymbol{C}_k \in {}^{p_k}\mathbb{S}^{n}$ are the observation projection matrices, where $k = 1 \ldots K$.

**Definition.** CLASS II: To implement these controllers *sensory sharing* is required. The tuple

$$\{\boldsymbol{B}_k, \boldsymbol{F}, \boldsymbol{C}_k\} \quad \forall k = 1 \ldots K$$

can be expressed in the form

$$\boldsymbol{F} = \bigoplus_{k=1}^{K} \left( \bigoplus_{j=1}^{K} \boldsymbol{B}_j^T \boldsymbol{B}_j \right) \boldsymbol{F} \boldsymbol{C}_k^T \boldsymbol{C}_k \tag{3.52}$$

but cannot be expressed in the form of Class I. $\boldsymbol{F} \in {}^{m}\mathbb{S}^{n}$ is the control law matrix, $\boldsymbol{B}_k \in {}^{m_k}\mathbb{S}^{m}$ are the action projection matrices, and $\boldsymbol{C}_k \in {}^{p_k}\mathbb{S}^{n}$ are the observation projection matrices, where $k = 1 \ldots K$.

**Definition.** CLASS III: To implement these controllers *actuator coordination* is required. The tuple

$$\{\boldsymbol{B}_k, \boldsymbol{F}, \boldsymbol{C}_k\} \quad \forall k = 1 \ldots K$$

can be expressed in the form

$$\boldsymbol{F} = \bigoplus_{k=1}^{K} \boldsymbol{B}_k^T \boldsymbol{B}_k \boldsymbol{F} \left( \bigoplus_{j=1}^{K} \boldsymbol{C}_j^T \boldsymbol{C}_j \right) \tag{3.53}$$

but cannot be expressed in the form of Class I. $\boldsymbol{F} \in {}^{m}\mathbb{S}^{n}$ is the control law matrix, $\boldsymbol{B}_k \in {}^{m_k}\mathbb{S}^{m}$ are the action projection matrices, and $\boldsymbol{C}_k \in {}^{p_k}\mathbb{S}^{n}$ are the observation projection matrices, where $k = 1 \ldots K$.

**Definition.** CLASS IV: To implement these controllers both *sensory sharing* and *actuator coordination* is required. The tuple

$$\{\boldsymbol{B}_k, \boldsymbol{F}, \boldsymbol{C}_k\} \quad \forall k = 1 \ldots K$$

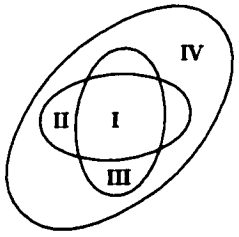



Figure 3.3: Partitioning of the set of global reactive controllers into four mutually exclusive classes due to constraints imposed by decentralization of observation and control. Only those global controllers in Class I may be exactly implemented using decentralized controllers (without communication).

can be expressed in the form

$$\boldsymbol{F} = \left(\bigoplus_{k=1}^{K} \boldsymbol{B}_k^T \boldsymbol{B}_k\right) \boldsymbol{F} \left(\bigoplus_{j=1}^{K} \boldsymbol{C}_j^T \boldsymbol{C}_j\right) \tag{3.54}$$

but cannot be expressed in the form of Class I,II, or III. $\boldsymbol{F} \in {}^{m}\mathbb{S}^{n}$ is the control law matrix, $\boldsymbol{B}_k \in {}^{m_k}\mathbb{S}^{m}$ are the action projection matrices, and $\boldsymbol{C}_k \in {}^{p_k}\mathbb{S}^{n}$ are the observation projection matrices, where $k = 1 \ldots K$.

These four classes are useful in that each type of global reactive controller requires a qualitatively different types of communication for implementation using decentralized controllers. Only Class I controllers can be implemented using no communication between decentralized controllers. However, as will be shown, with the use of communication between controllers, the other classes may also be implemented (by essentially transforming them into Class I controllers).

## 3.6.2 Communication

We will now discuss the following relationships between the classes of reactive controllers.

$$\begin{array}{ccc} \text{Class I} & \xleftarrow[\text{Controls}]{\text{Communicate}} & \text{Class II} \\ \text{Commu}\uparrow\text{nicate}\ \text{Observa}\uparrow\text{tions} & & \text{Commu}\uparrow\text{nicate}\ \text{Observa}\uparrow\text{tions} \\ \text{Class III} & \xleftarrow[\text{Controls}]{\text{Communicate}} & \text{Class IV} \end{array} \tag{3.55}$$

By communicating controls and observations, it is possible to transform all centralized reactive controllers into Class I type controllers which can be implemented exactly using decentralized controllers.

**Theorem 4.** *Reactive decentralized controllers which communicate can implement any reactive centralized controller exactly. Communication must be perfect and instantaneous.*

**Proof.** Let the implemented centralized reactive controller, $\boldsymbol{F}_{\text{cen}} \in {}^{m}\mathbf{S}^{n}$, be given by

$$\boldsymbol{F}_{\text{cen}} = \left(\bigoplus_{k=1}^{K} \mathbf{B}_k^T \mathbf{B}_k\right) \boldsymbol{F} \left(\bigoplus_{j=1}^{K} \mathbf{C}_j^T \mathbf{C}_j\right) \tag{3.56}$$

where $\boldsymbol{F} \in {}^{m}\mathbf{S}^{n}$ is a Class IV control law matrix, $\mathbf{B}_k \in {}^{m_k}\mathbb{D}^{m}$ are the action projection matrices, and $\mathbf{C}_k \in {}^{p_k}\mathbb{D}^{n}$ are the observation projection matrices, where $k = 1 \ldots K$. The projection matrices

are assumed to be deterministic as they must be to implement $\boldsymbol{F}$ perfectly even in the centralized case. This equation represents bringing together all of the sensor data, computing the control, and sending out the controls to all of the actuators in a centralized manner. Because $\boldsymbol{F} \in {}^{m}\mathbf{S}^{n}$ is a Class IV control law matrix, we cannot immediately implement it on noncommunicating decentralized controllers. The implemented decentralized controller, $\boldsymbol{F}_{\text{decen}} \in {}^{m}\mathbf{S}^{n}$, is given by

$$\boldsymbol{F}_{\text{decen}} = \bigoplus_{k=1}^{K} \mathbf{B}_k^{\mathcal{T}} \mathbf{B}_k \boldsymbol{F}_{\text{cen}} \mathbf{C}_{k,\text{new}}^{\mathcal{T}} \mathbf{C}_{k,\text{new}} \tag{3.57}$$

where $\mathbf{C}_{k,\text{new}} \in {}^{p_k}\mathbf{S}^{n}$ represent new projection matrices which result from the process of communicating sensor information. The sharing of observation information may be described as follows. The state-observation projection matrix for each controller is replaced with another which combines the communicated state-observation information from all controllers[6].

$$\mathbf{C}_{k,\text{new}}^{\mathcal{T}} \mathbf{C}_{k,\text{new}} = \bigoplus_{l=1}^{K} \mathbf{C}_l^{\mathcal{T}} \mathbf{C}_l \tag{3.58}$$

which must be done $\forall k = 1 \ldots K$. This makes the assumption that communication is perfect and instantaneous. We also notice that we now have

$$\mathbf{C}_{1,\text{new}}^{\mathcal{T}} \mathbf{C}_{1,\text{new}} = \cdots = \mathbf{C}_{K,\text{new}}^{\mathcal{T}} \mathbf{C}_{K,\text{new}} \tag{3.59}$$

Using (3.58) and (3.56) in (3.54) we have

$$\begin{aligned}
\boldsymbol{F}_{\text{decen}} &= \bigoplus_{k=1}^{K} \mathbf{B}_k^{\mathcal{T}} \mathbf{B}_k \boldsymbol{F}_{\text{cen}} \mathbf{C}_{k,\text{new}}^{\mathcal{T}} \mathbf{C}_{k,\text{new}} && (3.60)\\
&= \bigoplus_{k=1}^{K} \mathbf{B}_k^{\mathcal{T}} \mathbf{B}_k \left( \bigoplus_{j=1}^{K} \mathbf{B}_j^{\mathcal{T}} \mathbf{B}_j \right) \boldsymbol{F} \left( \bigoplus_{l=1}^{K} \mathbf{C}_l^{\mathcal{T}} \mathbf{C}_l \right) \mathbf{C}_{k,\text{new}}^{\mathcal{T}} \mathbf{C}_{k,\text{new}} && (3.61)\\
&= \bigoplus_{k=1}^{K} \mathbf{B}_k^{\mathcal{T}} \mathbf{B}_k \left( \bigoplus_{j=1}^{K} \mathbf{B}_j^{\mathcal{T}} \mathbf{B}_j \right) \boldsymbol{F} \left( \bigoplus_{l=1}^{K} \mathbf{C}_l^{\mathcal{T}} \mathbf{C}_l \right) \left( \bigoplus_{p=1}^{K} \mathbf{C}_i^{\mathcal{T}} \mathbf{C}_i \right) && (3.62)\\
&= \bigoplus_{k=1}^{K} \mathbf{B}_k^{\mathcal{T}} \mathbf{B}_k \left( \bigoplus_{j=1}^{K} \mathbf{B}_j^{\mathcal{T}} \mathbf{B}_j \right) \boldsymbol{F} \left( \bigoplus_{l=1}^{K} \mathbf{C}_l^{\mathcal{T}} \mathbf{C}_l \right) && (3.63)\\
&= \bigoplus_{k=1}^{K} \mathbf{B}_k^{\mathcal{T}} \mathbf{B}_k \boldsymbol{F}_{\text{cen}} && (3.64)
\end{aligned}$$

where we have used (2.7) to deduce

$$\left( \bigoplus_{i=1}^{K} \mathbf{C}_i^{\mathcal{T}} \mathbf{C}_i \right) = \left( \bigoplus_{j=1}^{K} \mathbf{C}_j^{\mathcal{T}} \mathbf{C}_j \right) \left( \bigoplus_{k=1}^{K} \mathbf{C}_k^{\mathcal{T}} \mathbf{C}_k \right) \tag{3.65}$$

The communication of controls is taken care of in a slightly different fashion. Consider first the centralized case. When the control matrix, $\boldsymbol{F}_{\text{cen}} \in \mathbf{S}$, is not deterministic the centralized controller must select a deterministic action from the distribution defined by $\boldsymbol{F}_{\text{cen}}$. It is only at the ensemble level that a stochastic control has been implemented. At the single-trajectory level, deterministic actions are chosen.

The $K$ decentralized controllers must also have a method to agree on a deterministic action selected from the possibilities defined by $\boldsymbol{F}_{\text{cen}}$. Agreeing on controls will in general involve repeated communications which will be explained in Chapters 4 and 5. Here we will *assume* they can do so

[6] In general we will not need to share all observation information, just enough to satisfy the condition of Theorem 3.

instantaneously and in a completely decentralized manner and leave Chapters 4 and 5 to explain this assumption. If we assume that the decentralized controllers are (at each call to the control law) able to select a deterministic control from the distribution defined by $\boldsymbol{F}_{\text{cen}}$, then $\boldsymbol{F}_{\text{cen}}$ is able to effectively distribute (since deterministic matrices distribute) so that

$$\begin{aligned}
\boldsymbol{F}_{\text{decen}} &= \bigoplus_{k=1}^{K} \mathbf{B}_k^{\mathcal{T}} \mathbf{B}_k \boldsymbol{F}_{\text{cen}} && (3.66)\\
&= \left( \bigoplus_{k=1}^{K} \mathbf{B}_k^{\mathcal{T}} \mathbf{B}_k \right) \boldsymbol{F}_{\text{cen}} && (3.67)\\
&= \left( \bigoplus_{k=1}^{K} \mathbf{B}_k^{\mathcal{T}} \mathbf{B}_k \right) \left( \bigoplus_{j=1}^{K} \mathbf{B}_j^{\mathcal{T}} \mathbf{B}_j \right) \boldsymbol{F} \left( \bigoplus_{l=1}^{K} \mathbf{C}_l^{\mathcal{T}} \mathbf{C}_l \right) && (3.68)\\
&= \left( \bigoplus_{j=1}^{K} \mathbf{B}_j^{\mathcal{T}} \mathbf{B}_j \right) \boldsymbol{F} \left( \bigoplus_{l=1}^{K} \mathbf{C}_l^{\mathcal{T}} \mathbf{C}_l \right) && (3.69)\\
&= \boldsymbol{F}_{\text{cen}} && (3.70)
\end{aligned}$$

thus concluding the proof. We have used that

$$\left( \bigoplus_{i=1}^{K} \mathbf{B}_i^{\mathcal{T}} \mathbf{B}_i \right) = \left( \bigoplus_{j=1}^{K} \mathbf{B}_j^{\mathcal{T}} \mathbf{B}_j \right) \left( \bigoplus_{k=1}^{K} \mathbf{B}_k^{\mathcal{T}} \mathbf{B}_k \right) \tag{3.71}$$

By assuming decentralized controllers may agree on an action from a list of possibilities, we have avoided half of this proof. This is the problem of *self-organization* and it is large enough to warrant treatment in a chapter of its own (Chapter 5).

In the case that the control matrix, $\boldsymbol{F}_{\text{cen}} \in \mathbb{D}$, really is deterministic, this has the simplification of limiting the centralized controller to being in Class III or I. The decentralized controllers only need to communicate sensor information and do not need to communicate to agree on an action to carry out (since there is only one possibility). This can be viewed as control coordination being imposed at design time rather than at implementation. ■

Thus through the communication of controls and observations, it is possible to implement all classes of centralized reactive controllers as Class I which can be handled by reactive decentralized controllers just as well as a reactive centralized controller. This is an important result for two reasons. First, better performance can be achieved than without the communication, and second, the solution of the more difficult control problems which arise if communication is not used (e.g., solving a DecPOMDP directly) is computationally much more involved.

If we assume the condition of Theorem 3 is satisfied (for the centralized case), then there is a mapping between the diagrams, (3.1.3) and (3.55), and we may see the complete role of communication in decentralized reactive control. The process of decentralizing control can be countered by the communication (and thus coordination) of local controls between controllers. The process of partially observing actions may be countered by the communication (coordination) of observations between controllers. There is a nice dual between control and observation, both being simplified through communication.

### 3.6.3 Decoupling the System

It should be stressed that the reason Class I global controllers are much easier to work with is that the contribution of the decentralized controllers can be considered independently of one another. This is

reflected in the mathematical form of (3.51) which is a simple vector addition of the contributions of the $K$ controllers. We here see the benefit of using stochastic algebra as we may interpret this addition as statistical independence. Once communication has occurred, the controllers have been effectively decoupled, allowing each to behave independently of the others. This however does not mean that the *effects* of the controls are decoupled. There can still be interaction between the controllers but this is reflected through the transition matrix not the controllers themselves.

## 3.7 Observability and Controllability

To investigate the possibilities for non-reactive control, in which a multistep (dynamic) state estimator is used, we consider the notions of ***controllability*** and ***observability***[7] in the context of the DecPOMDP model. These concepts will be presented for the centralized control case and may be generalized to decentralized systems using similar methods as for the reactive control case. The main purpose of this section is to show the benefits of improving the state estimator (observability) which applies equally to centralized and decentralized control. Any differences for the decentralized control case are pointed out. To complete the section, controllability is discussed briefly.

### 3.7.1 Observability

When using a multistep state estimator, it is important to know whether or not this estimator, $\hat{\mathbf{x}}$, can exactly reproduce the true state, $\mathbf{x}$, in a finite period of time. To answer this, we must make the distinction between two types of "partial observability". First, we could have a state-observation projection matrix of size $O \times N$ with $O = N$ but which is nondeterministic. Second, we could have a deterministic state-observation projection matrix of size $O \times N$ where $O < N$.

Partially observability of the first kind truly is partially observable because the true state can only be constructed exactly in the limit of an infinite number of measurements. However, with $\mathbf{C}$ deterministic, it is sometimes possible to reconstruct the state perfectly. In general if a multistep estimator is required, observability requires that the transition matrix, $\mathbf{A}$, is also deterministic. Thus the term "partially observable" is somewhat misleading in the case of deterministic $\mathbf{C}$ and $\mathbf{A}$ as the system could actually be fully observable (in the traditional control theory sense). This is of course a very limited case to which we devote a small amount of attention for mathematical curiosity.

#### No Transient States

Consider first a Markov chain only (no control variable) with deterministic transition matrix.

$$\mathbf{x}[t+1] = \mathbf{A}\mathbf{x}[t] \tag{3.72}$$

where $\mathbf{x} \in {}^{n}\mathbb{D}$ and $\mathbf{A} \in {}^{n}\mathbb{D}^{n}$. Furthermore, add the condition that $\mathbf{A}$ has no zero rows such that

$$\mathbf{A}^{\intercal}\mathbf{A} = \mathbf{1} \tag{3.73}$$

[7] When speaking of observability, we mean the observation of the global state, $\boldsymbol{x}$, as opposed to observation of the global control, $\boldsymbol{u}$.

This ensures there are no transient states, the Markov chain is regular. This condition will be relaxed later. Let us observe this chain through a deterministic observation matrix.

$$\mathbf{y}[t] = \mathbf{C}\mathbf{x}[t] \tag{3.74}$$

where $\mathbf{y} \in {}^{p}\mathbb{D}$ and $\mathbf{C} \in {}^{p}\mathbb{D}^{n}$. Then based on (3.41), the state estimate at time $t_f$ is the following

$$\hat{\mathbf{x}}[t_f] = \bigoplus_{t=0}^{t_f} \mathbf{A}^t \mathbf{C}^T \mathbf{C}\mathbf{x}[t_f - t] \tag{3.75}$$

$$= \bigoplus_{t=0}^{t_f} \mathbf{A}^t \mathbf{C}^T \mathbf{C}\mathbf{A}^{t_f - t}\mathbf{x}[0] \tag{3.76}$$

If we want $\hat{\mathbf{x}}[t_f] = \mathbf{x}[t_f] = \mathbf{A}^{t_f}\mathbf{x}[0]$ then we require that

$$\mathbf{A}^{t_f} = \bigoplus_{t=0}^{t_f} \mathbf{A}^t \mathbf{C}^T \mathbf{C}\mathbf{A}^{t_f - t} \tag{3.77}$$

Then using (3.73) we find

$$(\mathbf{A}^T)^{t_f}\mathbf{A}^{t_f} = (\mathbf{A}^T)^{t_f}\left(\bigoplus_{t=0}^{t_f} \mathbf{A}^t \mathbf{C}^T \mathbf{C}\mathbf{A}^{t_f - t}\right) \tag{3.78}$$

and then (2.3)

$$\mathbf{1} = \bigoplus_{t=0}^{t_f} (\mathbf{A}^T)^{t_f}\mathbf{A}^t \mathbf{C}^T \mathbf{C}\mathbf{A}^{t_f - t} \tag{3.79}$$

$$\mathbf{1} = \bigoplus_{t=0}^{t_f} (\mathbf{A}^T)^{t_f - t}\mathbf{C}^T \mathbf{C}\mathbf{A}^{t_f - t} \tag{3.80}$$

$$\mathbf{1} = \bigoplus_{t=0}^{t_f} \left(\mathbf{A}^{t_f - t}\right)^T \mathbf{C}^T \mathbf{C}\mathbf{A}^{t_f - t} \tag{3.81}$$

This last step requires that $\left(\mathbf{A}^T\right)^{t_f - t} = \left(\mathbf{A}^{t_f - t}\right)^T$ which is true for $\mathbf{A} \in \mathbb{D}$ with no transient states. We in fact need only to consider $n$ terms in the addition of (3.81) owing to the Cayley-Hamilton theorem. The system is essentially cycling through all $n$ states, some of which we may distinguish as observed through $\mathbf{C}$. Once we have gone all the way through the cycle once, no more observations will improve the estimate (this will not be so when $\boldsymbol{C}$ or $\boldsymbol{A}$ is nondeterministic).

### Detectability

We now consider transient states. After a finite time, the system will never return to these states. It is a well known property of Markov chains that the transition, $\mathbf{A} \in {}^{n}\mathbb{D}^{n}$, and state-observation, $\mathbf{C} \in {}^{p}\mathbb{D}^{n}$, projection matrices may always be expressed (by reordering the states) in the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix} \tag{3.82}$$

where $\mathbf{A}_{11} \in {}^{n_1}\mathbb{D}^{n_1}$ is the transition matrix of nontransient states, $\mathbf{A}_{22} \in {}^{n_2}\mathbb{D}^{n_2}$ is the transition matrix of transient states and $\mathbf{A}_{12} \in {}^{n_2}\mathbb{D}^{n_1}$ involves transition from the transient states to the nontransient

states. Note $n_1 + n_2 = n$. The presence of transient states means that (3.73) no longer holds and thus to check observability we must use (3.77) rather than (3.81). However, it may not be important to be able to observe the transient states as the system will leave these eventually and never return. Instead, we consider only the nontransient states which are still being occupied. If we require only that we be able to observe the nontransient states this is analogous to *detectability* of a linear system. We have that

$$\mathbf{A}_{11}^{\tau}\mathbf{A}_{11} = \mathbf{1} \tag{3.83}$$

which means we may again use (3.81) to test that the nontransient states are observable. We must replace $\mathbf{A}$ with $\mathbf{A}_{11}$ and $\mathbf{C}$ with $\mathbf{C}_1$.

### Implementability

Certainly if the system is truly observable as defined above then we should be able to implement any control matrix, $\boldsymbol{F}$, as though we had access to the system state directly. Furthermore, if it is only detectable it might be reasonable to say the same. However, as in the reactive case we may further take advantage of the fact that we only need to tell states apart if they require taking different controls. We might consider the following

$$\hat{\mathbf{x}}(n-1) = \bigoplus_{t=0}^{n-1} \mathbf{A}^t\mathbf{C}^{\tau}\mathbf{C}\mathbf{x}(n-1-t) \tag{3.84}$$

$$= \left(\bigoplus_{t=0}^{n-1} \mathbf{A}^t\mathbf{C}^{\tau}\mathbf{C}(\mathbf{A}^{\tau})^t\right)\mathbf{x}(n-1) \tag{3.85}$$

so we may check that

$$\boldsymbol{F} = \boldsymbol{F}\left(\bigoplus_{t=0}^{n_1-1} \mathbf{A}^t\mathbf{C}^{\tau}\mathbf{C}(\mathbf{A}^{\tau})^t\right) \tag{3.86}$$

for *implementable observability*, where $\boldsymbol{F} \in {}^m\mathbb{S}^n$ is the control matrix. This is a generalization of Theorem 2. If we only want to worry about the nontransient states then we can test

$$\boldsymbol{F}_1 = \boldsymbol{F}_1\left(\bigoplus_{t=0}^{n-1} \mathbf{A}_{11}^t\mathbf{C}_1{}^{\tau}\mathbf{C}_1(\mathbf{A}_{11}{}^{\tau})^t\right) \tag{3.87}$$

for *implementable detectability* where

$$\boldsymbol{F} = \begin{bmatrix}\boldsymbol{F}_1 \\ \boldsymbol{F}_2\end{bmatrix} \tag{3.88}$$

with $\boldsymbol{F}_1 \in {}^m\mathbb{S}^{n_1}$ and $\boldsymbol{F}_2 \in {}^m\mathbb{S}^{n_2}$.

### Reintroducing the Controls

The discussion of observability so far has been limited to Markov chains. We now return to the controlled system where there are different transition matrices for different control inputs. Observability now means we must be able to reconstruct the system state regardless of what the controls are. We must require (3.81) for all possible control sequences. This is clearly not a very practical condition to compute for large $n$ and $m$.

### Nondeterminism

The test of (3.81) usual fails when either the transition matrix or state-observation projection matrix is nondeterministic. For example, consider

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{C} = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix} \tag{3.89}$$

which fails the test of (3.81) for finite $t_f$. However, in the limit of an infinite number of measurements we have

$$\lim_{t_f \to \infty} \bigoplus_{t=0}^{t_f} (\mathbf{A}^t)^T \boldsymbol{C}^T \boldsymbol{C} \mathbf{A}^t \tag{3.90}$$

$$= \lim_{t_f \to \infty} \bigoplus_{t=0}^{t_f} \boldsymbol{C}^T \boldsymbol{C}$$

$$= \lim_{t_f \to \infty} (2t_f) \cdot \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}$$

$$= \begin{cases} \boldsymbol{\Omega} & \text{if } p = \frac{1}{2} \\ \mathbf{1} & \text{otherwise} \end{cases} \tag{3.91}$$

which means we will get a perfect state estimate but only in the infinite limit with $p \neq \frac{1}{2}$. It is unlikely that a controller will have the luxury of waiting that long before being required to output a control. Similar arguments could be made for the case of $\boldsymbol{A}$ nondeterministic. This has led researchers to develop controllers that do not require a perfect state estimate of the world in order to act (POMDP algorithms), as described in the literature review section of this chapter. We have been investigating where these methods are not necessary in the hope of more clearly defining the boundaries. Still, if the above test of observability does fail even with an infinite number of measurements (e.g., when $p = \frac{1}{2}$ above) the POMDP algorithm will not have any information to work with at all.

POMDP algorithms are extremely interesting but unfortunately are at the moment very computationally intensive. A POMDP can only be exactly solved for a few hundred states. Approximate algorithms extend the possibilites to a few thousand states but this is still very limited. We suggest that if the partial observability is actually introduced by the process of decentralization, then communication is a very reasonable alternative to overcome this problem. Particularly in a practical situation where a communication infrastructure is also necessary to implement a "centralized" controller.

### Decentralized Differences

The main difference between centralized and decentralized state-estimation turns out to be a result of partial observability of the past global control. In using the recursive state-estimator of (3.41) we see that the old control is needed in order to know which transition matrix to use in the estimator. If we have that $\mathbf{D} = \mathbf{1}$ then this problem is avoided as the past global control is then known. This is almost a universal assumption of centralized control. However, if it is not (as in decentralized control) then we have the added complication that an estimate of this control must be built and then an estimate of the

transition matrix

$$\hat{\mathbf{x}}[t] = \mathbf{A}(\hat{\mathbf{u}}[t-1])\hat{\mathbf{x}}[t-1] \oplus \mathbf{C}^{\mathcal{T}}\mathbf{y}[t] \tag{3.92}$$

$$\hat{\mathbf{u}}[t] = \mathbf{F}\hat{\mathbf{x}}[t] \oplus \mathbf{D}^{\mathcal{T}}\mathbf{v}[t] \tag{3.93}$$

where we now notice that knowledge of the control matrix, $\mathbf{F} \in \mathbb{S}$, can improve the state estimate. This coupling between control law and state-estimator makes this case much more difficult than when $\mathbf{D} = \mathbf{1}$. There currently are not any algorithms which have been designed to explicitly take into account this coupling but if one were found it would surely be more computationally intensive than solving a POMDP. In the decentralized case, it would be more complicated than solving $K$ POMDPs. However, if we assume that each controller can recall its own past controls perfectly then

$$\mathbf{D}_1^{\mathcal{T}}\mathbf{D}_1 \oplus \mathbf{D}_2^{\mathcal{T}}\mathbf{D}_2 \oplus \cdots \oplus \mathbf{D}_K^{\mathcal{T}}\mathbf{D}_K \equiv \mathbf{1} \tag{3.94}$$

which means that communication may once again come to the rescue of decentralized control. If we define a new control-observation matrix, $\mathbf{D}_{k,\text{new}}$, for each controller, $k = 1 \ldots K$, as

$$\mathbf{D}_{k,\text{new}}^{\mathcal{T}}\mathbf{D}_{k,\text{new}} = \bigoplus_{l=1}^{K} \mathbf{D}_l^{\mathcal{T}}\mathbf{D}_l = \mathbf{1} \tag{3.95}$$

such that knowledge of $\mathbf{F}$ no longer improves the estimate of the past global control and

$$\begin{aligned} \hat{\mathbf{u}}_k[t] &= \mathbf{D}_{k,\text{new}}^{\mathcal{T}}\mathbf{v}_k[t] \\ &= \mathbf{D}_{k,\text{new}}^{\mathcal{T}}\mathbf{D}_{k,\text{new}}\mathbf{u}[t] \\ &= \mathbf{u}[t] \end{aligned} \tag{3.96}$$

which means we may now solve $K$ POMDPs as a worst case. As discussed earlier, further communication may be able to ensure an even simpler problem to solve than this. With full communication of both observations and controls, decentralized controllers should be able to implement the same global controller as the centralized case.

## 3.7.2 Controllability

In linear systems, controllability refers to the ability to achieve any system state through appropriate control sequences, regardless of the initial conditions. Controllability of an MDP really only makes sense when the transition matrix is deterministic. It is possible to derive a rank condition involving the transition matrix but it is not particularly practical. The main reason for this is that the transition matrix depends jointly on the old system state and the control; it is nonlinear. The contributions of these must be combined before multiplying by the transition matrix, not after.

The more general concept, *regularity* or *ergodicity*, handles the case of nondeterministic transition matrices. A simple test, in the case of deterministic transitions, to determine regularity is

**Theorem 5.** *Compute the following*

$$\hat{\mathbf{A}}\mathbf{x}[t] = \left[\mathbf{A}_1\mathbf{A}_2\cdots\mathbf{A}_m\right](\mathbf{x}[t],\mathbf{u}[t])\Big|_{u=\omega} \tag{3.97}$$

$$N = \downarrow\left(\mathbf{1}+\hat{\mathbf{A}}\right)^{n-1} \tag{3.98}$$

*The system is regular if and only if N has no zeros.*

**Proof.** Based on a result in [Nazin and Poznyak, 1986] which says that the underlying Markov structure (e.g., the number and size of the ergodic subclasses) under an arbitrary control strategy is the same as under a *randomized control strategy* (i.e., it is invariant to the control strategy). See also [Poznyak and Najim, 1999]. Expanding we have

$$\downarrow\left(\mathbf{1}+\hat{\boldsymbol{A}}\right)^{n-1} = \downarrow\underbrace{\left(\mathbf{1}+\hat{\boldsymbol{A}}+\hat{\boldsymbol{A}}^2+\cdots+\hat{\boldsymbol{A}}^{n-1}\right)}_{n \text{ terms}} \tag{3.99}$$

where we need only $n$ terms in the sum to fully characterize $\hat{\boldsymbol{A}}$ (Cayley-Hamiliton Theorem). If there is an entry of $\boldsymbol{N}$ which is zero then it is zero in all $n$ terms above. This implies that there is a transition path between two particular states which may never occur (no matter how long we wait) and the system is not regular under random controls and thus not regular under any controls. ■

This relies on the fact that under a random control matrix, the system will visit all possible states if the system has only one *ergodic set* [Nazin and Poznyak, 1986]. This means that there is a control sequence which will (with probability $> 0$ ) eventually take the system to any state from any state. The fact that $n$ is finite means that it can be done in finite time (on average). We are essentially testing that the transition matrix, under random controls (and thus any controls), is *regular*. The above theorem handles the case of transient states automatically. The theorem says nothing about being able to stay at a particular state or how often the system will visit it, only that it is possible to visit it.

## 3.8 Linearizing a Markov Chain

Recognizing that a Markov chain is a nonlinear equation in our stochastic algebra, we turn to some of the common techniques of nonlinear analysis (see, for example, [Khalil, 1996]). These will later be extended to the DecPOMDP model of (3.20) through (3.23). It is well known that a Markov transition matrix, $\boldsymbol{A} \in {}^{n}\mathbb{S}^{n}$, may always be expressed (by reordering the states) in the form

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{A}_{1m} \\ \boldsymbol{0} & \boldsymbol{A}_{22} & \boldsymbol{0} & \cdots & \boldsymbol{A}_{2m} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{33} & \cdots & \boldsymbol{A}_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{A}_{mm} \end{bmatrix} \tag{3.100}$$

where $\boldsymbol{A}_{ii} \in {}^{n_i}\mathbb{S}^{n_i}\ \forall i = 1 \ldots m$ and $\boldsymbol{A}_{im} \in {}^{n_i}\mathbb{S}^{n_m}\ \forall i = 1 \ldots m-1$ [Kemeny and Snell, 1976]. Here $m$ is the number of *ergodic sets*[8] associated with the transition matrix $\boldsymbol{A}$ and $n = \sum_{i=1}^{m} n_i$. The bottom $n_m$ states are the transient states.

When there is only a single ergodic set the Markov transition matrix is sometimes called *regular* (a.k.a., ergodic, stationary, normal). Each of the $\boldsymbol{A}_{ii}\ \forall i = 1 \ldots m$ is a regular transition matrix. A very nice property of a regular transition matrix, $\boldsymbol{A} \in {}^{n}\mathbb{S}^{n}$, is that

$$\lim_{t \to \infty} \boldsymbol{A}^{t} = \begin{bmatrix} \boldsymbol{b}\,\boldsymbol{b} \cdots \boldsymbol{b} \end{bmatrix} \tag{3.101}$$

where $\boldsymbol{b} \in {}^{n}\mathbb{S}$. That is, all the columns of the combined transition matrix, $\boldsymbol{A}^{t}$, approach a single distribution, $\boldsymbol{b}$, as $t \to \infty$ [Papoulis, 1965]. Thus $\boldsymbol{b}$ is an equilibrium of the transition function

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{b} \tag{3.102}$$

We may interpret this equilibrium distribution in one of two ways. First, we may think of it as the probability of a single trajectory passing through a particular point in state-space. Second, it has been shown that the equilibrium distribution actual defines the fraction of time (over an infinitely long interval) that a single trajectory will spend in each state (see, for example, [Papoulis, 1965] or [Cox and Miller, 1965]). It is in fact possible to use an information theoretic measure, such as Shannon information, as a Lyapunov function to show that $\boldsymbol{b}$ is in fact globally asympototically stable [Prigogine and Stengers, 1984].

Although the stochastic equation

$$\boldsymbol{x}[t+1] = \boldsymbol{A}\boldsymbol{x}[t] \tag{3.103}$$

with $\boldsymbol{x} \in {}^{n}\mathbb{S}$ and $\boldsymbol{A} \in {}^{n}\mathbb{S}^{n}$ is linear in matrix algebra, the addition and zero of matrix algebra are not relevant to stochastic matrices, hence we turn to stochastic algebra. In stochastic algebra, this is a nonlinear equation even when $\boldsymbol{A}$ is regular. Let $\boldsymbol{b} \in {}^{n}\mathbb{S}$ be the equilibrium of $\boldsymbol{A}$. We may then shift the equilibrium, $\boldsymbol{b}$, to the origin, $\boldsymbol{\omega}$, through the change of variables $\boldsymbol{x} = \boldsymbol{y} \oplus \boldsymbol{b}$ where $\boldsymbol{y} \in {}^{n}\mathbb{S}$ so that

$$\boldsymbol{y}[t+1] = \boldsymbol{A}(\boldsymbol{y}[t] \oplus \boldsymbol{b}) \ominus \boldsymbol{b} \tag{3.104}$$

[8] The number of ergodic sets (a.k.a., ergodic subclasses) may be computed as the number of eigenvalues of $\boldsymbol{A}$ equal to 1 where the characteristic polynomial is $\det(\lambda \boldsymbol{1} - \boldsymbol{A})$ not $\det(\lambda \cdot \boldsymbol{\Xi} \ominus \boldsymbol{A})$. Kemeny and Snell [1976].

so that now when $y[t] = \omega$ we have $y[t+1] = \omega$. We now have a stochastic equation with a stable equilibrium at $\omega$. Still, however, this is not a linear equation. Another common technique from nonlinear analysis is *linearization*. We select an operating point, typically an equilibrium, around which the nonlinear equation is replaced by a linear one. The linear equation is an approximation of the nonlinear one. To do this we let

$$\boldsymbol{x} = \boldsymbol{b} \oplus \delta\boldsymbol{x} \tag{3.105}$$

where $\delta\boldsymbol{x} \in {}^n\mathbb{S}$ and $\boldsymbol{b} \in {}^n\mathbb{S}$ is the operating point. Then, the linearized equation (about $\boldsymbol{b}$) is given by

$$\delta\boldsymbol{x}[t+1] = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{b}} \otimes \delta\boldsymbol{x}[t] \tag{3.106}$$

where

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} \tag{3.107}$$

and

$$\left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{b}} \tag{3.108}$$

is the Jacobian of $\boldsymbol{f}(\boldsymbol{x})$ evaluated at the desired operating point $\boldsymbol{x} = \boldsymbol{b}$. We at last have a linear equation to which the full force of linear algebra may be applied.

**Example.** Consider the system

$$\boldsymbol{x}[t+1] = \begin{bmatrix} 1-p & q \\ p & 1-q \end{bmatrix} \boldsymbol{x}[t]$$

which has a regular transition matrix (for $p \in (0,1)$, $q \in (0,1)$) and an equilibrium at

$$\boldsymbol{x} = \downarrow \begin{bmatrix} q \\ p \end{bmatrix}$$

The linearized equation is then

$$\begin{aligned} \delta\boldsymbol{x}[t+1] &= \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{b}} \otimes \delta\boldsymbol{x}[t] \\ &= \downarrow \begin{bmatrix} e^{1-p-q} & 1 \\ 1 & e^{1-p-q} \end{bmatrix} \otimes \delta\boldsymbol{x}[t] \\ &= (1-p-q)\cdot\boldsymbol{\Xi} \otimes \delta\boldsymbol{x}[t] \end{aligned}$$

where $\delta\boldsymbol{x} = \boldsymbol{x} \ominus \begin{bmatrix} q \\ p \end{bmatrix}$. □

One of the main advantages of this technique is that we may now consider the eigen problem, thus characterizing the equilibrium point by its eigenvalues. For a linearized regular Markov chain these eigenvalues should always have magnitude $< 1$ as we know the equilibrium of the nonlinear system is globally asymptotically stable [Papoulis, 1965]. Note, solving the eigen problem for a regular, $\boldsymbol{A}$, with characteristic polynomial $\det(\lambda\mathbf{1} - \boldsymbol{A})$ always has as one eigenvalue, $\lambda = 1$ (since there is a single ergodic subclass). However, in the new eigenvalue problem (in stochastic algebra) with characteristic polynomial, $\det(\lambda\cdot\boldsymbol{\Xi} \ominus \boldsymbol{A})$, we lose this eigenvalue. This is because the new characteristic polynomial is of a smaller degree (by one). It makes a great deal more sense to consider the new stochastic algebra eigen problem as the system really has only $n-1$ degrees of freedom and thus will have $n-1$, not $n$, eigenvalues.

**Example.** Continuing from the previous example. The characteristic polynomial is

$$\begin{aligned} p(\lambda) &= \det\Big(\lambda\cdot\boldsymbol{\Xi} \ominus \boldsymbol{A}\Big) \\ &= \det\Big(\lambda\cdot\boldsymbol{\Xi} \ominus (1-p-q)\cdot\boldsymbol{\Xi}\Big) \\ &= \lambda - (1-p-q) \end{aligned}$$

which has the root $\lambda = 1-p-q$. Notice for $p \in (0,1)$, $q \in (0,1)$ we have $|\lambda| < 1$ which means the equilibrium is stable as expected. For comparison, consider the matrix algebra eigen problem.

$$\begin{aligned} p(\lambda) &= \det\Big(\lambda\mathbf{1} - \boldsymbol{A}\Big) \\ &= \det\left(\downarrow\begin{bmatrix} \lambda-(1-p) & -q \\ -p & \lambda-(1-q) \end{bmatrix}\right) \\ &= (\lambda-1)\big(\lambda-(1-p-q)\big) \end{aligned}$$

which has the roots, $\lambda = 1$ and $\lambda = 1-p-q$. Note, in this case the eigenvalue, $\lambda = 1-p-q$, is the same in both eigen problems but in general this will not always be the case (since linearization is an approximation). □

The method described above falls into the category of *approximate linearization* where a nonlinear stochastic equation has been replaced by a linear stochastic equation which is easier to analyze. The drawback is that the linearized system may only be a good approximation in a small neighbourhood near the equilibrium point.

In the special case of a *doubly stochastic*[9] transition matrix, $\boldsymbol{A} = [a_{ij}] \in {}^n\mathbb{S}^n$, we may transform the Markov chain equation into an exactly equivalent linear one by

$$\boldsymbol{y} = \downarrow\big[e^{x_i}\big] \qquad \boldsymbol{B} = \downarrow\big[e^{a_{ij}}\big] \tag{3.109}$$

where $\boldsymbol{x} = [x_i] \in {}^n\mathbb{S}$ such that we have

$$\boldsymbol{y}[t+1] = \boldsymbol{B}\otimes\boldsymbol{y}[t] \tag{3.110}$$

which is a linear equation in stochastic algebra.

[9] Both the rows and columns sum to 1.

## 3.9 Linearizing a Markov Decision Process

We now turn to the DecPOMDP model of (3.20) through (3.23) in which we will again attempt a linearization to recover the classic *linear* decentralized system studied by Wang and Davison [1973].

We first require an assumption. We assume that under a random set of controls, $\boldsymbol{u} = \boldsymbol{\omega}$, the resulting Markov transition matrix is *regular* such that (3.20) becomes

$$\begin{aligned} \boldsymbol{x}[t+1] &= \Big[\boldsymbol{A}_1 \boldsymbol{A}_2 \cdots \boldsymbol{A}_m\Big](\boldsymbol{x}[t], \boldsymbol{u}[t])\Big|_{\boldsymbol{u}=\boldsymbol{\omega}} && (3.111)\\ &= \hat{\boldsymbol{A}}\boldsymbol{x}[t] && (3.112) \end{aligned}$$

with $\hat{\boldsymbol{A}} \in {}^n\mathbb{S}^n$ and $\boldsymbol{x} \in {}^n\mathbb{S}$. This is equivalent to the ergodicity test suggested previously. If $\hat{\boldsymbol{A}}$ is not regular then there are at least two ergodic sets. This implies that regardless of the controls applied, we may never get from one ergodic set to another. Let $\boldsymbol{x}^* \in {}^n\mathbb{S}$ be the equilibrium distribution of the regular transition matrix, $\hat{\boldsymbol{A}}$. If we let

$$\boldsymbol{f}(\boldsymbol{x}) = \Big[\boldsymbol{A}_1 \boldsymbol{A}_2 \cdots \boldsymbol{A}_m\Big](\boldsymbol{x}[t], \boldsymbol{u}[t]) \tag{3.113}$$

then we may create the *linearized* system

$$\delta\boldsymbol{x}[t+1] = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^*\\ \boldsymbol{u}=\boldsymbol{\omega}}} \otimes \delta\boldsymbol{x}[t] \oplus \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^*\\ \boldsymbol{u}=\boldsymbol{\omega}}} \otimes \delta\boldsymbol{u}[t] \tag{3.114}$$

where $\delta\boldsymbol{x} = \boldsymbol{x} \ominus \boldsymbol{x}^* \in {}^n\mathbb{S}$ and $\delta\boldsymbol{u} = \boldsymbol{u} \ominus \boldsymbol{\omega} = \boldsymbol{u} \in {}^m\mathbb{S}$. Note, the Jacobians must be evaluated at the equilibrium $(\boldsymbol{x}^*, \boldsymbol{\omega})$.

**Example.** Consider the Markov system

$$\begin{gathered} \boldsymbol{x}[t+1] = \boldsymbol{f}(\boldsymbol{x}[t], \boldsymbol{u}[t])\\ \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = \big[\boldsymbol{A}_1 \boldsymbol{A}_2\big](\boldsymbol{x}, \boldsymbol{u}) \qquad (\boldsymbol{x}, \boldsymbol{u}) = \Big[\boldsymbol{x}_1\boldsymbol{u}_1 \quad \boldsymbol{x}_2\boldsymbol{u}_1 \quad \boldsymbol{x}_1\boldsymbol{u}_2 \quad \boldsymbol{x}_2\boldsymbol{u}_2\Big]^{\mathrm{T}}\\ \boldsymbol{A}_1 = \begin{bmatrix} 1-p & q \\ p & 1-q \end{bmatrix} \qquad \boldsymbol{A}_2 = \begin{bmatrix} 1-q & p \\ q & 1-p \end{bmatrix} \end{gathered}$$

which has an equilibrium $(\boldsymbol{x}, \boldsymbol{u}) = (\boldsymbol{\omega}, \boldsymbol{\omega})$ such that we have

$$\left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{\omega}\\ \boldsymbol{u}=\boldsymbol{\omega}}} = (1-p-q)\cdot\boldsymbol{\Xi} \qquad \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{\omega}\\ \boldsymbol{u}=\boldsymbol{\omega}}} = (q-p)\cdot\boldsymbol{\Xi}$$

The linearized system, with $\delta\boldsymbol{x} = \boldsymbol{x} \ominus \boldsymbol{\omega} = \boldsymbol{x}$ and $\delta\boldsymbol{u} = \boldsymbol{u} \ominus \boldsymbol{\omega} = \boldsymbol{u}$, is

$$\delta\boldsymbol{x}[t+1] = (1-p-q)\cdot\delta\boldsymbol{x}[t] \oplus (q-p)\cdot\delta\boldsymbol{u}[t]$$

where we note that $|1-q-p| < 1$ for $p \in (0,1)$, $q \in (0,1)$ which means when the controls are random, $\delta\boldsymbol{u}[t] = \boldsymbol{\omega}$, the system state, $\boldsymbol{x}$, will tend towards $\boldsymbol{\omega}$. Also note that when $p = q$ we no longer have any control over the system. □

In a general decentralized system, the global control, $\boldsymbol{u}$, will actually be a function of a number of decentralized controls, $\boldsymbol{u}_k$ $\forall k = 1 \ldots K$. For the linearized decentralized system we have

$$\delta\boldsymbol{x}[t+1] = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^*\\ \boldsymbol{u}=\boldsymbol{\omega}}} \otimes \delta\boldsymbol{x}[t] \bigoplus_{k=1}^{K} \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}_k}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^*\\ \boldsymbol{u}=\boldsymbol{\omega}}} \otimes \delta\boldsymbol{u}_k[t] \tag{3.115}$$

where $\delta x = x \ominus x^* \in {}^n\mathbb{S}$ and $\delta u_k = u_k \ominus \omega = u_k \in {}^{m_k}\mathbb{S}$. Note, the Jacobians must be evaluated at the equilibrium $(x^*, \omega)$. We might also wish to linearize the observation equation (3.22) which we may think of again as a nonlinear stochastic equation

$$y_k[t] = g_k(x[t]) \tag{3.116}$$

with $y \in {}^{p_k}\mathbb{S}$, $g(x) \in {}^{p_k}\mathbb{F}^n$ $\forall k = 1 \ldots K$. We further assume that when $x = \omega$ we have $y_k = \omega$ $\forall k = 1 \ldots K$. Then let $\delta y_k = y \ominus \omega = y$ and linearize so that

$$\delta y_k[t] = \left.\frac{\partial g_k}{\partial x}\right|_{x=\omega} \otimes \delta x[t] \tag{3.117}$$

Combining (3.115) and (3.117) we have a system of equations of the form

$$x[t+1] = A \otimes x[t] \bigoplus_{k=1}^{K} B_k \otimes u_k[t] \tag{3.118}$$

$$y_k[t] = C_k \otimes x[t] \qquad \forall k = 1 \ldots K \tag{3.119}$$

where $x \in {}^n\mathbb{S}$, $A \in {}^n\mathbb{S}^n$, $u_k \in {}^{m_k}\mathbb{S}$, $B_k \in {}^m\mathbb{S}^{m_k}$, $y_k \in {}^{p_k}\mathbb{S}$, $C_k \in {}^{p_k}\mathbb{S}^n$ $\forall k = 1 \ldots K$. This system is exactly of the (discrete-time) form examined in [Wang and Davison, 1973] but cast in stochastic algebra rather than matrix algebra. To complete the connection to the well known linear system we may simply employ the isomorphism, $\varphi$, described in Section 2.9, Page 24. We select orthonormal bases for ${}^n\mathbb{S}$, ${}^{m_k}\mathbb{S}$, and ${}^{p_k}\mathbb{S}$ and use the isomorphic transformation to arrive at

$$\mathsf{x}[t+1] = \mathsf{A}\mathsf{x}[t] + \sum_{k=1}^{K} \mathsf{B}_k \mathsf{u}_k[t] \tag{3.120}$$

$$\mathsf{y}_k[t] = \mathsf{C}_k \mathsf{x}[t] \qquad \forall k = 1 \ldots K \tag{3.121}$$

where

$$\begin{gathered} \mathsf{x} = \varphi(x)^{-1} \in {}^{n-1}\mathbb{R} \qquad \mathsf{u}_k = \varphi(u_k)^{-1} \in {}^{m_k-1}\mathbb{R} \qquad \mathsf{y}_k = \varphi(y_k)^{-1} \in {}^{p_k-1}\mathbb{R} \\ \mathsf{A} = \varphi(A)^{-1} \in {}^{n-1}\mathbb{R}^{n-1} \qquad \mathsf{B}_k = \varphi(B_k)^{-1} \in {}^{n-1}\mathbb{R}^{m_k-1} \qquad \mathsf{C}_k = \varphi(C_k)^{-1} \in {}^{p_k-1}\mathbb{R}^n \end{gathered} \tag{3.122}$$

From a notational point of view, this system appears fairly similar to the one we started with in equations (3.20) through (3.22) but there is a major difference. The state, control and observations vectors are now in ${}^{n-1}\mathbb{R}$, ${}^{m_k-1}\mathbb{R}$, ${}^{p_k-1}\mathbb{R}$ rather than ${}^n\mathbb{S}$, ${}^{m_k}\mathbb{S}$, ${}^{p_k}\mathbb{S}$. We must stress that a linearized system will likely only be useful in a neighbourhood around the equilibrium point $(x^*, \omega)$. However, in that neighbourhood we may employ all of the immense resources at hand for linear systems to better our understanding of the system. Furthermore, in the previous discussion we have been assuming that we are linearizing the system in equations (3.20) through (3.22) but these techniques are quite general in nature and may be applied to any system of the form

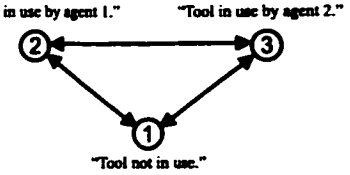
$$x[t+1] = f(x[t], u_k[t]) \tag{3.123}$$

$$y_k[t] = g_k(x[t], u_k[t]) \tag{3.124}$$

with $x \in {}^n\mathbb{S}$, $u_k \in {}^{m_k}\mathbb{S}$, $y \in {}^{p_k}\mathbb{S}$, $f(x, u_k) \in {}^n\mathbb{F}^{n,m}$, $g_k(x, u_k[t]) \in {}^{p_k}\mathbb{F}^{n,m}$ $\forall k = 1 \ldots K$.

## 3.10 A Tool of Two Agents

In this section we investigate the usefulness of the above linearization approach in the context of a simple example. The example is of two agents that must share a common resource. Possiblities for this common resource include a computer file (e.g., a report), a block of shared memory, even "the television remote control". For our purposes, we will assume that the resource is some physical tool. The tool can be in one of three states: not held by either agent, held by agent 1, or held by agent 2.




Thus the state vector will be $\boldsymbol{x} \in {}^3\mathbb{S}$ and will correspond to the aforementioned possible states for the tool. We will further assume that each agent has at its disposal two possible actions: stay, switch. 'Stay' will correspond to an agent trying to keep the tool if it holds it or do nothing if it does not hold it. 'Switch' will correspond to an agent trying to pick up the tool if it does not hold it and put down the tool if it does hold it. The action vectors for the agents will be $\boldsymbol{u}_1 \in {}^2\mathbb{S}$ and $\boldsymbol{u}_2 \in {}^2\mathbb{S}$. We thus have

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{matrix} \leftarrow \text{tool not held} \\ \leftarrow \text{tool held by agent 1} \\ \leftarrow \text{tool held by agent 2} \end{matrix} \quad \boldsymbol{u}_1 = \begin{bmatrix} u_{11} \\ u_{12} \end{bmatrix} \begin{matrix} \leftarrow \text{stay} \\ \leftarrow \text{switch} \end{matrix} \quad \boldsymbol{u}_2 = \begin{bmatrix} u_{21} \\ u_{22} \end{bmatrix} \begin{matrix} \leftarrow \text{stay} \\ \leftarrow \text{switch} \end{matrix}$$

We will furthermore build into our model a stochastic element such that actions are not always successful. For example, if an agent holds the tool and wants to put it down, there will be a small probability, $2q$, that this does not happen. Likewise, if an agent has the tool and wants to keep it there will be a small probability, $2p$, that it is unable to do so. With this in mind we can construct our transition equation to be the following

$$\boldsymbol{x}[t+1] = \Bigg( \underbrace{\begin{bmatrix} 1-2p & p & p \\ p & 1-2p & p \\ p & p & 1-2p \end{bmatrix}}_{\substack{\text{agent 1: stay} \\ \text{agent 2: stay}}} \underbrace{\begin{bmatrix} q & p & 1-2q \\ q & 1-2p & q \\ 1-2q & p & q \end{bmatrix}}_{\substack{\text{agent 1: stay} \\ \text{agent 2: switch}}} \underbrace{\begin{bmatrix} q & 1-2q & p \\ 1-2q & q & p \\ q & q & 1-2p \end{bmatrix}}_{\substack{\text{agent 1: switch} \\ \text{agent 2: stay}}} \underbrace{\begin{bmatrix} 1-2p & q & q \\ p & q & 1-2q \\ p & 1-2q & q \end{bmatrix}}_{\substack{\text{agent 1: switch} \\ \text{agent 2: switch}}} \Bigg) \Big( \boldsymbol{x}[t], \boldsymbol{u}_1[t], \boldsymbol{u}_2[t] \Big)$$

where the underbracing indicates which transition matrix corresponds to which combination of actions by the agents. Notice that the allowable values for $p$ and $q$ are $p \leq 1/2$ and $q \leq 1/2$. We may linearize this equation using the method from the previous section. If we let $\boldsymbol{f} \in {}^3\mathbb{F}$ represent the above transition equation (i.e., the right hand side) then the Jacobians with respect to $\boldsymbol{x}$, $\boldsymbol{u}_1$, and $\boldsymbol{u}_2$ are

$$\left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\substack{x=\omega \\ u_1=\omega \\ u_2=\omega}} = \frac{1}{4}(1+3q-6p)\cdot\boldsymbol{\Xi} \qquad \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}_1}\right|_{\substack{x=\omega \\ u_1=\omega \\ u_2=\omega}} = \frac{3}{2}(p-q)\cdot\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ e^{-1} & e \end{bmatrix} \qquad \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}_2}\right|_{\substack{x=\omega \\ u_1=\omega \\ u_2=\omega}} = \frac{3}{2}(p-q)\cdot\begin{bmatrix} 1 & 1 \\ e^{-1} & e \\ 1 & 1 \end{bmatrix}$$

where we have chosen $(\boldsymbol{x}, \boldsymbol{u}_1, \boldsymbol{u}_2) = (\boldsymbol{\omega}, \boldsymbol{\omega}, \boldsymbol{\omega})$ as the point at which the linearization takes place. The linearized transition equation is thus

$$\delta\boldsymbol{x}[t+1] = \frac{1}{4}(1+3q-6p)\cdot\boldsymbol{\Xi}\otimes\delta\boldsymbol{x}[t] \oplus \frac{3}{2}(p-q)\cdot\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ e^{-1} & e \end{bmatrix}\otimes\delta\boldsymbol{u}_1[t] \oplus \frac{3}{2}(p-q)\cdot\begin{bmatrix} 1 & 1 \\ e^{-1} & e \\ 1 & 1 \end{bmatrix}\otimes\delta\boldsymbol{u}_2[t]$$

We can notice right away that when $p = q$ we have no way to control this model. A reasonable goal in this problem might be to try and equally partition the use of the tool between the two agents and minimize the time when neither agent has the tool. This corresponds to trying to achieve a state of the form

$$\delta \boldsymbol{x}^* = \lambda \cdot \begin{bmatrix} 1 \\ e \\ e \end{bmatrix}$$

where $\lambda > 0$. Naturally $\lambda$ cannot be arbitrarily large as the linearized system will likely only be valid near the linearization point. We may first check that this system is controllable by the two agents. If the decentralized controls are combined (independently) to a single vector

$$\delta \boldsymbol{u}[t] = \begin{bmatrix} e & 1 \\ e & 1 \\ 1 & e \\ 1 & e \end{bmatrix} \otimes \delta \boldsymbol{u}_1[t] \oplus \begin{bmatrix} e & 1 \\ 1 & e \\ e & 1 \\ 1 & e \end{bmatrix} \otimes \delta \boldsymbol{u}_2[t]$$

then our transition equation would be

$$\delta \boldsymbol{x}[t+1] = \boldsymbol{A} \otimes \delta \boldsymbol{x}[t] \oplus \boldsymbol{B} \otimes \delta \boldsymbol{u}[t]$$

where

$$\boldsymbol{A} = \frac{1}{4}(1 + 3q - 6p) \cdot \boldsymbol{\Xi} \qquad \boldsymbol{B} = \frac{3}{4}(p - q) \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ e^{-1} & e & e^{-1} & e \\ e^{-1} & e^{-1} & e & e \end{bmatrix}$$

It is not hard to verify that the rank of $\boldsymbol{B}$ is 2 (as long as $p \neq q$) so that the system is indeed controllable [Kalman, 1960] using the decentralized controls. This should be expected since each agent has one degree of freedom in its action choice and the state exists in a space of dimension 2. Also notice that if one of the agents behaves randomly (e.g., $\boldsymbol{u}_1 = \boldsymbol{\omega}$) the system becomes

$$\delta \boldsymbol{x}[t+1] = \boldsymbol{A} \otimes \delta \boldsymbol{x}[t] \oplus \boldsymbol{B}_2 \otimes \delta \boldsymbol{u}_2[t]$$

where

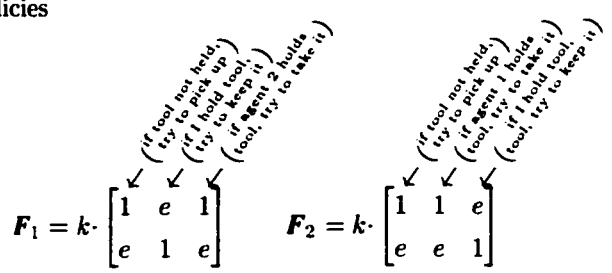
$$\boldsymbol{A} = \frac{1}{4}(1 + 3q - 6p) \cdot \boldsymbol{\Xi} \qquad \boldsymbol{B}_2 = \frac{3}{2}(p - q) \cdot \begin{bmatrix} 1 & 1 \\ e^{-1} & e \\ 1 & 1 \end{bmatrix}$$

so that now for controllability we must construct the full controllability matrix

$$[\, \boldsymbol{A} \otimes \boldsymbol{A} \otimes \boldsymbol{B}_2 \;\; \boldsymbol{A} \otimes \boldsymbol{B}_2 \;\; \boldsymbol{B}_2 \,] = \left[ \frac{1}{16}(1 + 3q - 6p)^2 \cdot \boldsymbol{B}_2 \;\; \frac{1}{4}(1 + 3q - 6p) \cdot \boldsymbol{B}_2 \;\; \boldsymbol{B}_2 \right]$$

which will have the same rank as $\boldsymbol{B}_2$ alone which is 1. Thus, if one agent behaves randomly the other cannot control the system through state feedback. This should be obvious since when agent 2 has the tool, agent 1 cannot be forced to put it down. If we would like to achieve the state $\delta \boldsymbol{x}^*$ above then we need to design a set of decentralized control laws that use state feedback. Two control laws that work

are the following "selfish" policies




$$\boldsymbol{F}_1 = k\cdot\begin{bmatrix} 1 & e & 1 \\ e & 1 & e \end{bmatrix} \qquad \boldsymbol{F}_2 = k\cdot\begin{bmatrix} 1 & 1 & e \\ e & e & 1 \end{bmatrix}$$

where $k > 0$ is a tunable gain. Let us also assume that $q > p$ for the remainder of the example. With these control laws in place the closed loop system that results is

$$\begin{aligned} \delta\boldsymbol{x}[t+1] &= \left(\boldsymbol{A} \oplus \boldsymbol{B}_1 \otimes \boldsymbol{F}_1 \oplus \boldsymbol{B}_2 \otimes \boldsymbol{F}_2\right) \otimes \delta\boldsymbol{x}[t] \\ &= \left(\frac{1}{4}(1+3q-6p)\cdot\boldsymbol{\Xi} \oplus \frac{3}{2}k(q-p)\cdot\begin{bmatrix} 1 & 1 & 1 \\ e & e & e^{-1} \\ e & e^{-1} & e \end{bmatrix}\right) \otimes \delta\boldsymbol{x}[t] \end{aligned}$$

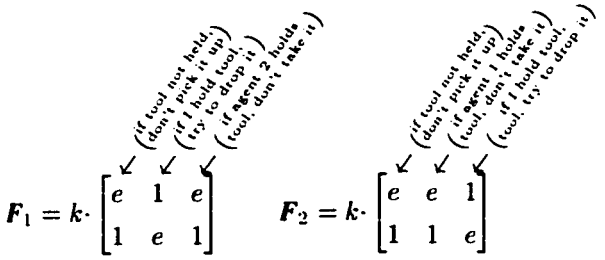
We can see that if the system starts at the point $\delta\boldsymbol{x}[t] = \delta\boldsymbol{x}^* = \lambda\cdot\begin{bmatrix} 1 & e & e \end{bmatrix}^T$ with $\lambda > 0$ at time $t$ then at the next time-step it will be at

$$\delta\boldsymbol{x}[t+1] = \left(\frac{1}{4}(1+3q-6p) + k(q-p)\right)\lambda\cdot\begin{bmatrix} 1 \\ e \\ e \end{bmatrix} = \left(\frac{1}{4}(1+3q-6p) + k(q-p)\right)\cdot\delta\boldsymbol{x}^*$$

In other words, the system state vector will point in the same direction and will be larger in magnitude when

$$k > \frac{1 - \frac{1}{4}(1+3q-6p)}{q-p}$$

Interestingly, we may also have "considerate" policies with the control laws




$$\boldsymbol{F}_1 = k\cdot\begin{bmatrix} e & 1 & e \\ 1 & e & 1 \end{bmatrix} \qquad \boldsymbol{F}_2 = k\cdot\begin{bmatrix} e & e & 1 \\ 1 & 1 & e \end{bmatrix}$$

where again $k > 0$ is a tunable gain. These "considerate" policies are just the (stochastic) negative of the "selfish" policies which is quite a nice interpretation to make. With the "considerate" policies we need to have

$$k < \frac{1 - \frac{1}{4}(1+3q-6p)}{q-p}$$

to increase the magnitude of $\delta\boldsymbol{x}[t]$. This is impossible for $k > 0$ since $\frac{1}{4}(1+3q-6p) < 1$ for the allowable values of $p$ and $q$. In summary, when $q > p$ the "selfish" policies are able to achieve the desired sharing of the tool between the two agents (with the gain, $k$, large enough) while the "considerate" policies are not. When $q < p$ the situation is reversed and when $q = p$ the system is uncontrollable entirely.

## 3.11 Another Example

This section provides another example of linearizing a DecPOMDP. We will be able to apply the controllability and observability tests of Kalman [1960] to see that the process of decentralization can change a system from controllable to uncontrollable and from observable to unobservable.

We consider the system given by

$$\begin{aligned} \boldsymbol{x}[t+1] &= \left[\boldsymbol{A}_1\,\boldsymbol{A}_2\,\boldsymbol{A}_3\,\boldsymbol{A}_4\right](\boldsymbol{x}[t],\boldsymbol{u}[t]) \\ \boldsymbol{u}[t] &= \boldsymbol{B}_1^T\boldsymbol{u}_1[t] \oplus \boldsymbol{B}_2^T\boldsymbol{u}_2[t] \\ \boldsymbol{y}[t] = \mathbf{1}\boldsymbol{x}[t] \qquad \boldsymbol{y}_1[t] &= \boldsymbol{C}_1\boldsymbol{y}[t] \qquad \boldsymbol{y}_2[t] = \boldsymbol{C}_2\boldsymbol{y}[t] \end{aligned}$$

where $\boldsymbol{x},\ \boldsymbol{u},\ \boldsymbol{y} \in {}^4\mathbb{S},\ \boldsymbol{u}_k,\ \boldsymbol{y}_k \in {}^2\mathbb{S}\ \forall k = 1\ldots2,\ \boldsymbol{A}_i \in {}^4\mathbb{S}^4\ \forall i = 1\ldots4$, and $\boldsymbol{B}_k, \boldsymbol{C} \in {}^2\mathbb{S}^4\ \forall k = 1\ldots2$. We let

$$\boldsymbol{A}_1 = \begin{bmatrix} 1-3p & q & q & q \\ p & 1-3q & q & q \\ p & q & 1-3q & q \\ p & q & q & 1-3q \end{bmatrix} \qquad \boldsymbol{A}_2 = \begin{bmatrix} 1-3q & p & q & q \\ q & 1-3p & q & q \\ q & p & 1-3q & q \\ q & p & q & 1-3q \end{bmatrix}$$

$$\boldsymbol{A}_3 = \begin{bmatrix} 1-3q & q & p & q \\ q & 1-3q & p & q \\ q & q & 1-3p & q \\ q & q & p & 1-3q \end{bmatrix} \qquad \boldsymbol{A}_4 = \begin{bmatrix} 1-3q & q & q & p \\ q & 1-3q & q & p \\ q & q & 1-3q & p \\ q & q & q & 1-3p \end{bmatrix}$$

$$\boldsymbol{B}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad \boldsymbol{B}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{C}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad \boldsymbol{C}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

with $p, q \in (0, \frac{1}{4})$. Notice that each of the $\boldsymbol{A}_i$ are regular transition matrices. In fact, they have equilibria, $\boldsymbol{x}_i^* \in {}^4\mathbb{S}\ \forall i = 1\ldots4$, given by

$$\boldsymbol{x}_1^* = \downarrow\begin{bmatrix} q \\ p \\ p \\ p \end{bmatrix} \qquad \boldsymbol{x}_2^* = \downarrow\begin{bmatrix} p \\ q \\ p \\ p \end{bmatrix} \qquad \boldsymbol{x}_3^* = \downarrow\begin{bmatrix} p \\ p \\ q \\ p \end{bmatrix} \qquad \boldsymbol{x}_4^* = \downarrow\begin{bmatrix} p \\ p \\ p \\ q \end{bmatrix}$$

We first linearize the centralized system arriving at the following linear system.

$$\begin{aligned} \delta\boldsymbol{x}[t+1] &= \boldsymbol{A}\otimes\delta\boldsymbol{x}[t] \oplus \boldsymbol{B}\otimes\delta\boldsymbol{u}[t] \\ \delta\boldsymbol{y}[t] &= \boldsymbol{C}\otimes\delta\boldsymbol{x}[t] \end{aligned}$$

where

$$\boldsymbol{A} = \underbrace{(1-p-3q)}_{\text{mag} < 1}\cdot\boldsymbol{\Xi} \qquad \boldsymbol{B} = (q-p)\cdot\boldsymbol{\Xi} \qquad \boldsymbol{C} = \boldsymbol{\Xi}$$

Note we have linearized about the point $(\boldsymbol{x},\boldsymbol{u}) = (\boldsymbol{\omega},\boldsymbol{\omega})$ so that $\delta\boldsymbol{x} = \boldsymbol{x}$, $\delta\boldsymbol{u} = \boldsymbol{u}$. Also $\delta\boldsymbol{y} = \boldsymbol{y}$. We may see immediately that when $\delta\boldsymbol{u} = \boldsymbol{\omega}$ the system will head to $\boldsymbol{x} = \boldsymbol{\omega}$ since the magnitude of $(1-p-3q)$ is

less than 1 for the allowable values of $p$ and $q$[10]. If we now select an orthonormal basis for ${}^4\mathbb{S}$

$$\phi_1 = \downarrow \begin{bmatrix} e \\ e \\ 1 \\ 1 \end{bmatrix} \qquad \phi_2 = \downarrow \begin{bmatrix} e \\ 1 \\ e \\ 1 \end{bmatrix} \qquad \phi_3 = \downarrow \begin{bmatrix} e \\ 1 \\ 1 \\ e \end{bmatrix}$$

we may use our isomorphism to convert this to the following system in matrix algebra.

$$\mathsf{x}[t+1] = \mathsf{A}\mathsf{x}[t] + \mathsf{B}\mathsf{u}[t] \qquad \mathsf{y}[t] = \mathsf{C}\mathsf{x}[t]$$
$$\mathsf{A} = (1-p-3q)1 \qquad \mathsf{B} = (q-p)1 \qquad \mathsf{C} = 1$$

where $\mathsf{x},\ \mathsf{u},\ \mathsf{y} \in {}^3\mathbb{R}$ and $\mathsf{A},\ \mathsf{B},\ \mathsf{C} \in {}^3\mathbb{R}^3$ are determined through the relations in (3.122). For controllability [Kalman, 1960] [11] we require

$$\operatorname{rank}\left[\mathsf{A}^2\mathsf{B}\ \ \mathsf{A}\mathsf{B}\ \ \mathsf{B}\right] = \operatorname{rank}\left[(q-p)(1-p-3q)^2 1\ \ (q-p)(1-p-3q)1\ \ (q-p)1\right] = 3$$

which is satisfied for the allowable values of $p$ and $q$ as long as $p \neq q$. For observability[12] we require

$$\operatorname{rank}\left[(\mathsf{A}^{\mathsf{T}})^2\mathsf{C}^{\mathsf{T}}\ \ \mathsf{A}^{\mathsf{T}}\mathsf{C}^{\mathsf{T}}\ \ \mathsf{C}^{\mathsf{T}}\right] = \operatorname{rank}\left[(1-p-3q)^2 1\ \ (1-p-3q)1\ \ 1\right] = 3$$

which is always satisfied. We now turn to the decentralized case. When we now linearize we end up with

$$\begin{aligned} \delta\boldsymbol{x}[t+1] &= \boldsymbol{A}\otimes\delta\boldsymbol{x}[t] \oplus \boldsymbol{B}_1\otimes\delta\boldsymbol{u}_1[t] \oplus \boldsymbol{B}_2\otimes\delta\boldsymbol{u}_2[t] \\ \delta\boldsymbol{y}_1[t] &= \boldsymbol{C}_1\otimes\delta\boldsymbol{x}[t] \\ \delta\boldsymbol{y}_2[t] &= \boldsymbol{C}_2\otimes\delta\boldsymbol{x}[t] \end{aligned}$$

where

$$\boldsymbol{A} = (1-p-3q)\cdot\Xi \qquad \boldsymbol{B}_1 = (q-p)\cdot\begin{bmatrix} e & 1 \\ e & 1 \\ 1 & e \\ 1 & e \end{bmatrix} \qquad \boldsymbol{B}_2 = (q-p)\cdot\begin{bmatrix} e & 1 \\ 1 & e \\ e & 1 \\ 1 & e \end{bmatrix}$$

$$\boldsymbol{C}_1 = \frac{1}{2}\cdot\begin{bmatrix} e & e & 1 & 1 \\ 1 & 1 & e & e \end{bmatrix} \qquad \boldsymbol{C}_2 = \frac{1}{2}\cdot\begin{bmatrix} e & 1 & e & 1 \\ 1 & e & 1 & e \end{bmatrix}$$

which, using the orthonormal basis from before, may be expressed in matrix algebra as

$$\mathsf{x}[t+1] = \mathsf{A}\mathsf{x}[t] + \mathsf{B}_1\mathsf{u}_1[t] + \mathsf{B}_2\mathsf{u}_2[t] \qquad \mathsf{y}_1[t] = \mathsf{C}_1\mathsf{x}[t] \qquad \mathsf{y}_2[t] = \mathsf{C}_2\mathsf{x}[t]$$
$$\mathsf{A} = (1-p-3q)1 \qquad \mathsf{B}_1 = \begin{bmatrix}(q-p) & 0 & 0\end{bmatrix}^{\mathsf{T}} \qquad \mathsf{B}_2 = \begin{bmatrix}0 & (q-p) & 0\end{bmatrix}^{\mathsf{T}}$$
$$\mathsf{C}_1 = \begin{bmatrix}1 & 0 & 0\end{bmatrix} \qquad \mathsf{C}_2 = \begin{bmatrix}0 & 1 & 0\end{bmatrix}$$

[10] We must comment that typically in linear control the system wants to head away from the origin and the goal is to stabilize the origin through feedback. Here the origin is naturally stable and we are trying to move away from it. Thus, the notion of stabilizing the origin does not make a lot of sense but the concepts of controllability and observability do not change. If we wanted to try to get to an arbitrary point other than the origin we still need the system to be controllable and observable.

[11] Controllability tests involving rank are simply to check we are able to affect all of the degrees of freedom, not necessarily perfectly.

[12] Observability tests involving rank are to check that we may obtain at least some information about each degree of freedom in the system, not necessarily contructing a perfect estimate.

where $\mathsf{x} \in {}^3\mathbb{R}$ $u_k$, $y_k \in \mathbb{R}$ $\forall k = 1 \ldots 2$ and $\mathsf{A} \in {}^3\mathbb{R}^3$, $\mathsf{B}_k \in {}^3\mathbb{R}$, $\mathsf{C}_k \in \mathbb{R}^3$ $\forall k = 1 \ldots 2$ are determined through the relations in (3.122). We let $\mathsf{B} = \begin{bmatrix} \mathsf{B}_1 & \mathsf{B}_2 \end{bmatrix} \in {}^3\mathbb{R}^2$ and $\mathsf{C}^{\mathrm{T}} = \begin{bmatrix} \mathsf{C}_1^{\mathrm{T}} & \mathsf{C}_2^{\mathrm{T}} \end{bmatrix} \in {}^3\mathbb{R}^2$ and test for controllability

$$\operatorname{rank}[\mathsf{A}^2\mathsf{B} \;\; \mathsf{AB} \;\; \mathsf{B}] = \operatorname{rank} \begin{bmatrix} (q-p)(1-p-3q)^2 & 0 & (q-p)(1-p-3q) & 0 & (q-p) & 0 \\ 0 & (q-p)(1-p-3q)^2 & 0 & (q-p)(1-p-3q) & 0 & (q-p) \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = 2 < 3$$

and observability

$$\operatorname{rank}[(\mathsf{A}^{\mathrm{T}})^2\mathsf{C}^{\mathrm{T}} \;\; \mathsf{A}^{\mathrm{T}}\mathsf{C}^{\mathrm{T}} \;\; \mathsf{C}^{\mathrm{T}}] = \operatorname{rank} \begin{bmatrix} (1-p-3q)^2 & 0 & (1-p-3q) & 0 & 1 & 0 \\ 0 & (1-p-3q)^2 & 0 & (1-p-3q) & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = 2 < 3$$

neither of which is satisfied. Thus the system is neither controllable nor observable using the decentralized controls and observations defined above. We may see that in both cases it is the third dimension, spanned by $\phi_3$ in the stochastic version, which causes the trouble. This may be interpreted as the "dependent" dimension, which the independent controllers may not control nor observe. This leads to a *fixed mode* which Wang and Davison [1973] showed is the natural extension of Kalman's [1960] uncontrollable and unobservable modes to decentralized systems. Thus we may again see that decentralization can enforce constraints on the input and output of the system which are equivalent to actually removing sensors and actuators. As stressed many times so far, it may be possible to overcome these deficits using communication between the controllers. Sometimes, communication may not be necessary, however. Consider the exact same system above with

$$\boldsymbol{A}_1 = \begin{bmatrix} p & q & q & 1-3q \\ 1-3p & q & q & q \\ p & 1-3q & q & q \\ p & q & 1-3q & q \end{bmatrix} \qquad \boldsymbol{A}_2 = \begin{bmatrix} q & p & q & 1-3q \\ 1-3q & p & q & q \\ q & 1-3p & q & q \\ q & p & 1-3q & q \end{bmatrix}$$

$$\boldsymbol{A}_3 = \begin{bmatrix} q & q & p & 1-3q \\ 1-3q & q & p & q \\ q & 1-3q & p & q \\ q & q & 1-3p & q \end{bmatrix} \qquad \boldsymbol{A}_4 = \begin{bmatrix} q & q & q & 1-3p \\ 1-3q & q & q & p \\ q & 1-3q & q & p \\ q & q & 1-3q & p \end{bmatrix}$$

so that when we linearize we have

$$\boldsymbol{A} = (1-p-3q)\cdot \begin{bmatrix} 1 & 1 & 1 & e \\ e & 1 & 1 & 1 \\ 1 & e & 1 & 1 \\ 1 & 1 & e & 1 \end{bmatrix} \qquad \boldsymbol{B}_1 = (q-p)\cdot \begin{bmatrix} 1 & e \\ e & 1 \\ e & 1 \\ 1 & e \end{bmatrix} \qquad \boldsymbol{B}_2 = (q-p)\cdot \begin{bmatrix} 1 & e \\ e & 1 \\ 1 & e \\ e & 1 \end{bmatrix}$$

where $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are unchanged. Converting to matrix algebra using the same orthonormal basis as before we have

$$\mathsf{A} = (1-p-3q) \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \qquad \mathsf{B}_1 = (p-q) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \mathsf{B}_2 = (p-q) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

with $\mathsf{C}_1$ and $\mathsf{C}_2$ unchanged. We let $\mathsf{B} = \begin{bmatrix} \mathsf{B}_1 & \mathsf{B}_2 \end{bmatrix} \in {}^3\mathbb{R}^2$ and $\mathsf{C}^{\mathrm{T}} = \begin{bmatrix} \mathsf{C}_1^{\mathrm{T}} & \mathsf{C}_2^{\mathrm{T}} \end{bmatrix} \in {}^3\mathbb{R}^2$ and test for controllability

$$\text{rank}[\mathbf{A}^2\mathbf{B}\ \mathbf{AB}\ \mathbf{B}] = \text{rank}\begin{bmatrix} 0 & 0 & (p-q)(1-p-3q) & 0 & 0 & 0 \\ 0 & (p-q)(1-p-3q)^2 & 0 & (q-p)(1-p-3q) & 0 & (p-q) \\ (q-p)(1-p-3q)^2 & 0 & 0 & 0 & (p-q) & 0 \end{bmatrix} = 3$$

and observability

$$\text{rank}[(\mathbf{A}^{\mathrm{T}})^2\mathbf{C}^{\mathrm{T}}\ \mathbf{A}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}\ \mathbf{C}^{\mathrm{T}}] = \text{rank}\begin{bmatrix} -(1-p-3q)^2 & 0 & 0 & 0 & 1 & 0 \\ 0 & (1-p-3q)^2 & 0 & -(1-p-3q) & 0 & 1 \\ 0 & 0 & (1-p-3q) & 0 & 0 & 0 \end{bmatrix} = 3$$

which are both satisfied for $p \neq q$. Thus it is at least feasible to use decentralized controls in the neighbourhood of the equilibrium. We see that due to the nature of the transition matrix, the first controller is able to affect two degrees of freedom while the second affects only one. Here communication may not be necessary as there are no fixed modes. Note, however, that although the system may be controllable using decentralized controls, it does not mean that the decentralized controllers may implement the exact same control as a centralized controller, only that they may be able to get the job done by interacting through the transition matrix instead of by communicating directly.

To conclude this example a simulation of the first system is presented in order to see how well the linearized model approximates the nonlinear Markov system. Values of $p = 0.02$ and $q = 0.03$ were used and simulated for 400 time-steps. The following controls were applied in sequence, each for 100 time-steps.

$$\boldsymbol{u} = \begin{bmatrix} 1-3s \\ s \\ s \\ s \end{bmatrix} \quad \boldsymbol{u} = \begin{bmatrix} s \\ 1-3s \\ s \\ s \end{bmatrix} \quad \boldsymbol{u} = \begin{bmatrix} s \\ s \\ 1-3s \\ s \end{bmatrix} \quad \boldsymbol{u} = \begin{bmatrix} s \\ s \\ s \\ 1-3s \end{bmatrix}$$

with $s = 0.02$. An initial condition of $\boldsymbol{x}[0] = \boldsymbol{\omega}$ was used. Figure 3.4 depicts the time series for the nonlinear Markov system and the linearized system. Note, there are four time series for each of the four components of $\boldsymbol{x}[t], \delta\boldsymbol{x}[t] \in {}^4\mathbb{S}$. Figure 3.5 depicts the difference, $\boldsymbol{x}[t] \ominus \delta\boldsymbol{x}[t] \in {}^4\mathbb{S}$, which as expected becomes larger as the system moves away from $\boldsymbol{x}[t] = \boldsymbol{\omega}$, the operating point. Figure 3.6 plots the *information*[13] in each of $\boldsymbol{x}$, $\delta\boldsymbol{x}$, and $\boldsymbol{x} \ominus \delta\boldsymbol{x}$ over time. The information in $\boldsymbol{x} \ominus \delta\boldsymbol{x}$ may be thought of as the error between the Markov and linearized models.

## 3.12 Summary

Stochastic matrices have been used in the study of uncontrolled Markov chains for a very long time. Here we see that the extension of this to a stochastic algebra allows some interesting results to be derived for Markov systems. Specifically, the system model for the Decentralized Partially Observable Markov Decision Problem (DecPOMDP) was treated, which is a nonlinear system in this new matrix formulation. The focus of study was on the actual process of decentralization and its effect on the type of control possible using decentralized controllers.

First, reactive (or static) control laws were studied for the nonlinear DecPOMDP. It was shown that the act of decentralization introduces constraints both on the input (observation) and output (control) of the system as a whole. These two constraints naturally define four classes of centralized reactive control

[13] See Section 2.18 (Page 37) for the definition of information.

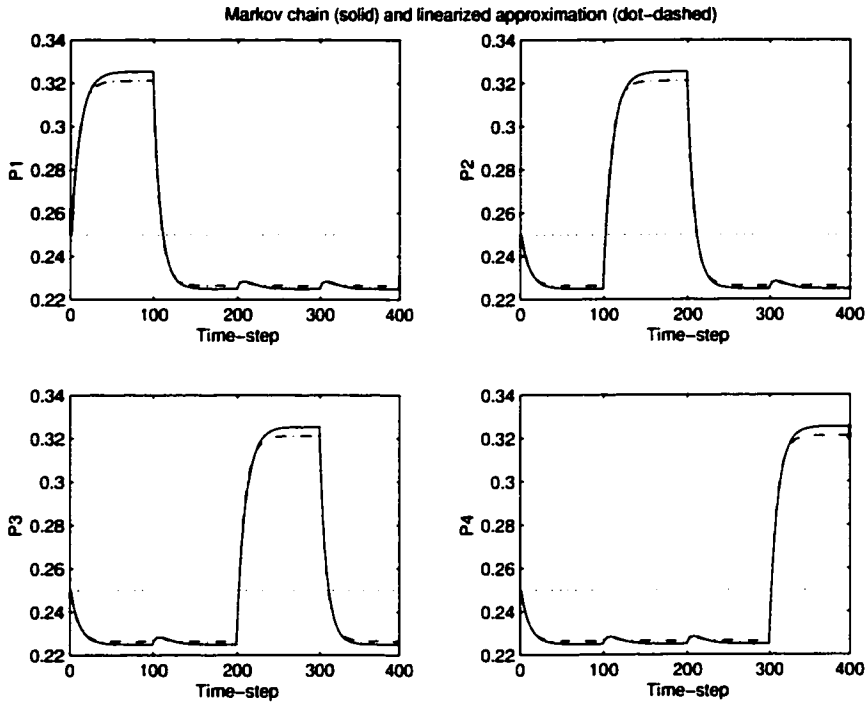



Figure 3.4: Time series for the Markov (solid) and linearized (dot-dashed) models. Note, there are four time series for each of the four components of $\boldsymbol{x}[t], \delta\boldsymbol{x}[t] \in {}^4\mathbb{S}$. Also, $\omega$, has been included in the plot which is the constant (dotted) line at 0.25.

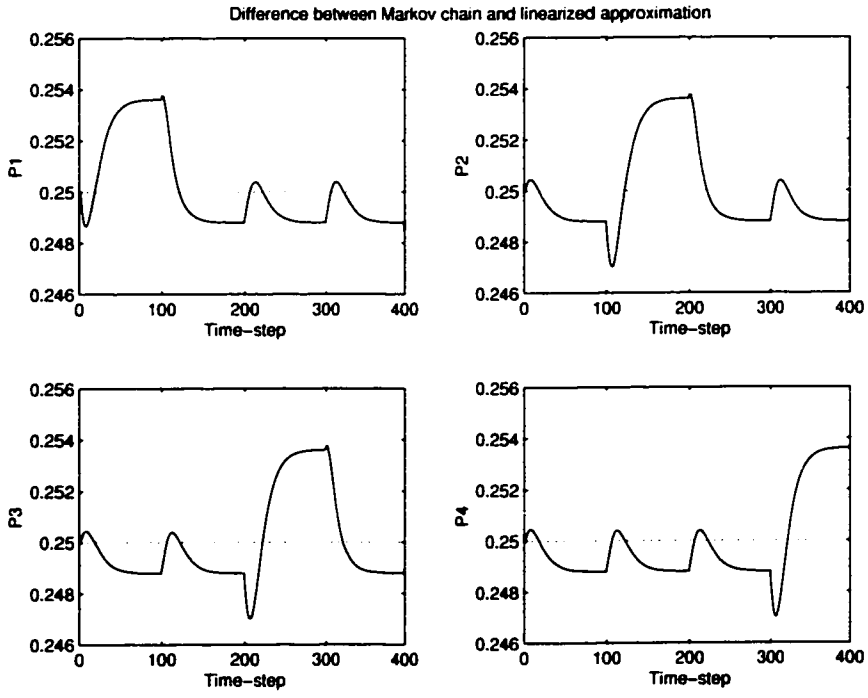



Figure 3.5: Difference, $\boldsymbol{x}[t] \ominus \delta\boldsymbol{x}[t] \in {}^4\mathbb{S}$, between the Markov and linearized models.

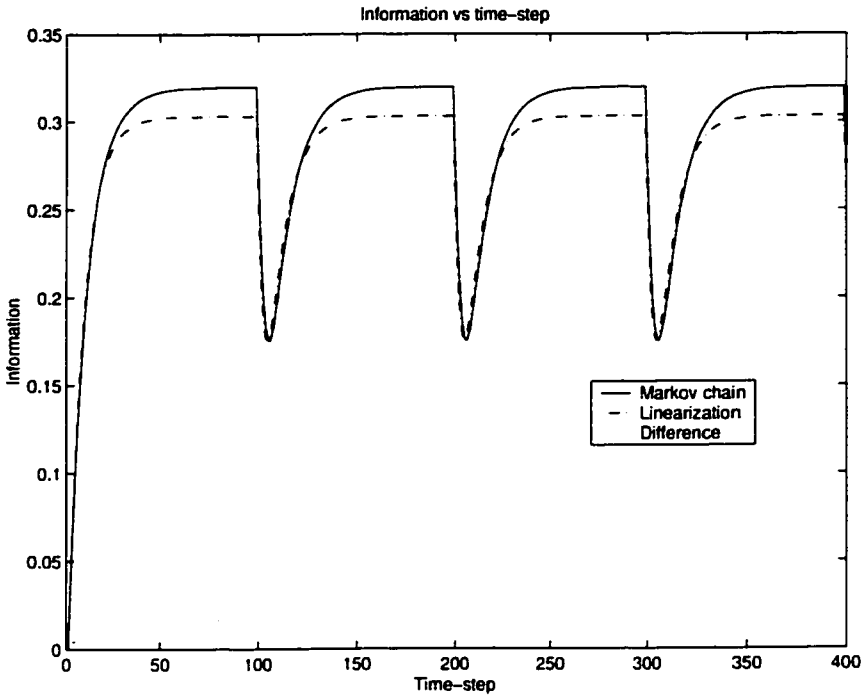



Figure 3.6: Information in each of $\boldsymbol{x}$, $\delta\boldsymbol{x}$, and $\boldsymbol{x} \ominus \delta\boldsymbol{x}$ over time. The information in $\boldsymbol{x} \ominus \delta\boldsymbol{x}$ may be thought of as the error between the Markov and linearized models.

laws. Only those of Class I may be exactly implemented using decentralized controllers which do not communicate. However, communication can aid in several aspects of decentralized control, assuming that it can take place at a rate much faster than the speed at which controls need to be applied. First, communication of observations, and thus state information, can help build better state-estimators for the decentralized controllers. Second, communication of controls can help coordinate the effects of the different controllers. In reactive control, communication of controls is often necessary when the global control matrix is nondeterministic. This corresponds to the situation where the system must be *self-organizing* (internally choosing between more than one equally good global control). Reactive decentralized controllers that communicate may exactly implement any reactive centralized controller.

Communication of controls can play another role, namely in improving the state-estimator of each controller when a dynamic state estimate is used. Centralized control typically makes the assumption that past controls are known and may be exploited in the state-estimator. With decentralized control this is not necessarily true. The communication of past actions avoids this problem. Furthermore, it separates the design of controllers from the design of the state-estimators. With communication of both observations and controls, a decentralized system with $K$ clusters of sensors and actuators is no worse than solving a POMDP problem. If the combination of observations from all controllers results in a perfect state estimate, it is no worse than solving an MDP problem. This is a large improvement over solving the DecPOMDP problem directly as it involves much less in the way of computation, particularly for the infinite horizon case mentioned here. Furthermore, the quality of solution obtained will be much better (as more information is being exploited). We believe that communication is a vital aspect of

decentralized control systems and that it is necessary for such systems to achieve interesting results.

Secondly, it was shown that it is actually possible to linearize a DecPOMDP model about an operating point using the new stochastic calculus, thereby converting it to a linear decentralized system in stochastic algebra. Through an isomorphism, this system may be rewritten in matrix algebra thereby creating a new link between DecPOMDPs and the discrete-time version of the system treated in [Wang and Davison, 1973]. This means that within some neighbourhood of the operating point, it is possible to apply the powerful (and plentiful) techniques that have been developed for linear systems. Through examples it was again shown that decentralization imposes constraints on the input and output which can switch the system from controllable to uncontrollable and observable to unobservable in the sense of Kalman [1960] (but in the new stochastic framework). Communication may again overcome these difficulties thereby restoring the centralized behaviour.

The notion of the uniform probability distribution as the *zero vector* is very appealing. Typically in control theory the idea is to try to get to the zero vector from an initial random point using appropriate feedback, to forget the initial conditions. Furthermore, the system would like to move away from that zero. For example, an inverted pendulum wants to fall one way or the other and we would like to prevent this from occurring; the system is naturally unstable and we are trying to stabilize it. In the Markovian control framework discussed here, the natural tendency of the system is to head towards zero (it is stable). The initial conditions are naturally forgotten. We are actually trying to destabilize the zero point in a very particular way through our control sequences in order to achieve distributions other than the uniform one. It is hoped the stochastic algebra will allow many more applications to be made of classic control results to stochastic systems.

A solitary ant, afield, cannot be considered to have much of anything on his mind. Four ants together, or ten, encircling a dead moth on a path, begin to look more like an idea. But it is only when you watch the dense mass of thousands of ants, blackening the ground that you begin to see the whole beast, and now you observe it thinking, planning, calculating. It is an intelligence, a kind of live computer, with crawling bits for its wits.
—Lewis Thomas
THE LIVES OF A CELL, 1974

# Chapter 4

# COLLECTIVE ROBOTICS

This chapter examines the possibilities for communication in a practical network of cooperating mobile robots. This is only one example of a real world decentralized system (with communication) but hopefully one that is instructive. It will be shown that communication can be used both to spatially extend sensing capabilities and to help coordinate actions. This dual role is demonstrated through various tasks on a group of mobile robots which communicates via radio. The focus here is to provide a methodology for building a collective robotics control architecture rather than to describe the detailed construction of the individual robots. Refer to [Earon et al., 2001] or [Barfoot et al., 2001] for further information in this regard.

The potential applications for a group of mobile robots are many, ranging from planetary space exploration to environmental cleanup to recreation. Cooperation is key to collective robotics. It is possible to design such systems so that they work together in parallel, carrying out tasks no one robot may accomplish alone. Here a methodology is described for doing this in a completely decentralized bottom-up manner. Some of the most appealing attributes of a group of robots (as opposed to a single monolithic vehicle) are redundancy, modularity, risk-taking, and cooperation. Redundancy is helpful in that if constructed in a completely decentralized manner, a network of robots can afford to lose a small number of individuals and yet still accomplish the desired task. This is a result of *modularity* which means any number of robots may be added or removed from the collective seamlessly. The flip side to this is risk-taking. By making the system modular and thus redundant, it is possible to take more risks in the design of each robot as no one robot has the potential to be a single point failure.

*Collective robotics* refers to groups of robots which interact to some degree [Cao et al., 1997]. Often

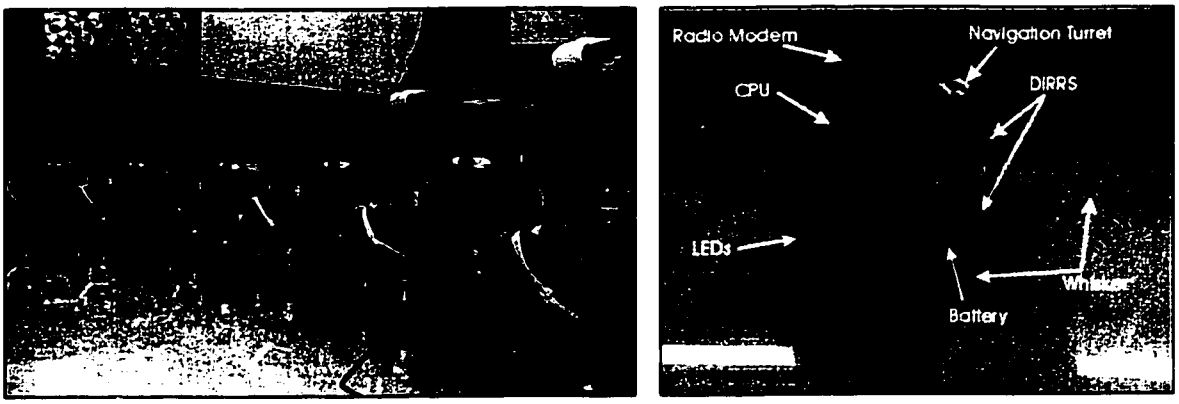



Figure 4.1: (left) Six mobile robots making up the RISE network. (right) Features of a single robot.

in collective robotics, the goal is to design the interactions between robots such that tasks requiring *cooperation* may be successfully completed. It might be more accurate to refer to this situation as *cooperative robotics* as there are many conceivable situations in which groups of robots are not trying to cooperate (e.g., competition). This chapter assumes that the robots are trying to cooperate, not compete.

The types of possible interaction between robots are many. There is in fact an entire spectrum of interaction possibilities. For example, at one end of the spectrum robots randomly bumping into each other may be considered interaction while a well established communication protocol which is employed frequently and leaves little room for misinterpretation (e.g., a direct radio link between robots) lies at the other end. Interactions should be categorized not by the transmission medium (as is sometimes done) but rather by the informational content transmitted.

Social insects are striking examples of biological organisms which have very well established communication protocols that are transmitted through the environment. A single species of ant may have as many as 20 different chemical signals called *pheromones* which enable complex inter-species communication. Pheromones are secreted from glands on one ant and are in turn detected by the highly sensitive tongues and, more commonly, antennae of other nearby ants. The term *superorganism* has been used to describe colonies of ants (i.e., up to millions of individuals) due to the incredible colony wide coordination afforded by pheromone communication. The "waggle dance" of some bees is another example of a well established protocol. The intricate motion of this dance can communicate both the distance and angle to a food source from the hive.

It is well known in nature, that groups of mobile agents (organisms) often use well established communication protocols in order to interact[1]. We suggest that the nature of the transmission medium for communication is irrelevant when designing cooperative robots and that the most convenient medium should be used. To this end, radio modems have been used in the design of a small group of cooperating mobile robots (see Figure 4.1). In this system short message packets are continually broadcast from all robots with a frequency of approximately 1 Hz. These messages allow the robots to both share sensory information and coordinate their action choices in order to be successful on cooperative tasks.

[1]Note, it is not within the scope of this thesis to debate whether biological organisms are cooperating or competing and thus we simply state that they are interacting.

This chapter will advocate a methodology for building up communications between robots which is a natural extension of the *behaviour-based* design philosophy [Brooks, 1986]. Each robot broadcasts messages at regular intervals over its radio modem. The messages are available to any robot within communication range (approximately 30 metres). However, robots do not stop to wait for a particular message from a specific robot. If information from one robot is available, it may be used in various parts of another robot's internal control system but if not, that robot continues on with what information it does have. This allows the system to operate in a robust and parallel manner. Should the battery die or the computer crash on one robot (both of which happen from time to time), this will not bring the entire system to a halt. Furthermore, the algorithms running on each robot (which are themselves decentralized behaviour-based controllers), have been designed to seamlessly accommodate different numbers of robots. Thus if a robot is incapacitated for any reason, the others compensate within seconds. This bottom-up approach to collective robotics is fast, modular, and reliable.

## 4.1 Related Work

There is a vast amount of literature for multiagent mobile robotics working in two dimensional environments as well as simulations (of varying complexity) of such robots. The simulations span from the very elaborate (with dynamics models) to the very simple (grid worlds). This review cannot possibly cover the entire range in depth so it will attempt to give a good cross section of results. Furthermore, it will be arranged more by the investigator and the tasks the groups of robots are to accomplish than by chronology. It describes only work that uses decentralized, bottom-up approaches to collective robotics.

Brooks [1986,1990, 1997, 1994] put down the foundations for multiagent mobile robotics in his repopularization of behaviour-based robotics. His *subsumption architecture* works in stark contrast to the logic-based artificial intelligence approaches. It is a robust decentralized control architecture for any type of robot. Lower priority behaviours yield control of the robot to higher priority ones when they become active. Some very interesting examples of robot behaviour such as wall following and obstacle avoidance were demonstrated in real-time, a goal that had eluded logic-based approaches for decades. Behaviour-based control provides a perfect substrate for mobile robotics and naturally accommodates decentralized control which is distributed over multiple robots. Communication may be easily incorporated and is used by iRobot [2000] which has been looking at behaviour-based control of groups of small mobile robots for military and other applications. They demonstrate similar types of tasks to the present work.

Matarić [1992, 1994a] was amongst the first to focus on multiagent control of real robots. Her early work consisted mostly of developing elaborate control structures for groups of mobile robots by hand. Flocking, herding, following and other behaviours were demonstrated. Matarić [1995] and Matarić and Cliff [1996] outline systematic approaches to programming single and groups of robots to work together. Later work focuses on learning group behaviours [Matarić, 1994b, Matarić, 1997b, Matarić, 1997a, Matarić, 1997c, Schneider-Fontán and Matarić, 1998] for about 4-10 robots. Simsarian and Matarić [1995, 1998] looked learning tightly coupled tasks such pushing a box towards a light using two robots. Here communication is used in a sensory sharing capacity to make the task easier to solve. Recent work [Goldberg and Matarić, 1999b, Goldberg and Matarić, 1999a] builds up a transition model

between different behaviour layers. This allows the robot to compare its present behaviour with this model to decide if it is functioning properly.

Werner and Dyer [1992] use genetic techniques to evolve the weights of artificial neural networks (ANNs). These ANNs act as control systems for simulated agents. Flocking, herding and schooling behaviours are evolved. Hodgins and Brogan [1994] provide an example of a herding behaviour in a simulated three-dimensional world incorporating simple dynamics (physics). Watson et al. [1999] describes an embodied evolutionary algorithm to search in policy space for simple neural network controllers for a light seeking task.

Kube and Zhang [1992, 1993, 1996] also work primarily in hardware. Their collective box pushing experiments are important as the robots are able to accomplish a difficult task in a matter of seconds rather than hours (as some experiments require).

Robotic soccer [Kitano et al., 1997] is an up and coming focal point in collective robotics research. RoboCup is the principal organization in this area. Stone [2000] has dominated the simulator league in this competition. There are also several leagues involving real robots. It is interesting to note that in the simulator league, messages may be passed between players through a simulation of noisy vocal communication (just as with humans playing soccer). This anthropomorphic addition is in contrast to the view taken here which is to use the most convenient transmission medium possible. However, it does have merits if the eventual goal of RoboCup is to have humans playing against robots, or perhaps mixed teams of both humans and robots in which the rules must be the same for all players. The situation we address is different in that we are simply trying to achieve a task the easiest way possible (while maintaining decentralized control). Balch [1997a, 1997b] also describes another soccer simulator, *Javabots*. His agents are all provided a common set of skills; during a learning phase they specialize their skills and take on different roles on the field (offence, defence, goal keeper). The *Swarm Simulation System*[2] [Minar et al., 1996] also now has a soccer toolkit available [Hochstein et al., 2000]. Robotic soccer provides a common framework for collective robotics researchers from all over the world to directly compare their results.

Deneubourg et al. [1990] describes a system of simulated grid world agents that are able to form heaps of objects as well as separate two different types of object into two heaps. Beckers et al. [1994], Maris and te Boekhorst [1996], and Pfeifer and Maris [1994] also consider the problem of heap formation. This entails a group of robots pushing pucks into a pile in an arena. They rely on a fortuitous mechanical design to do most of the difficult work thus making their control system conveniently simple. Melhuish et al. [1998] describe a more sophisticated version of this experiment in which different spatial distributions of objects can be achieved. Barfoot and D'Eleuterio [1999] also look at heap formation in a grid world simulation. They use genetic algorithms to evolve successful rules and demonstrate that their solution works on groups of hundreds of agents. Dagaeff et al. [1997] explore the interplay of cooperation and antagonism in another multiagent heap formation grid world. Bonabeau et al. [1994, 1998] describe a three-dimensional grid world in which groups of agents have been designed to construct elaborate structures similar to wasp nests. Most of these heap formation experiments involve subtle interactions through manipulation of the objects in the environment which is referred to as *stigmergy*. This is certainly

[2] *Swarm* is a general tool for the simulation of many simple agents that was started at the Sante Fe Institute.

an equally valid communication protocol but can be somewhat limited in its applicability to other tasks. However, it does show that successful communication protocols need not be in a representation that humans may easily interpret [Brooks, 1991].

Balch et al. [2001] has developed technology to automatically track the movements of thousands of live social insects in order to study their interactions and movement as a whole. They hope to apply this knowledge to the design of decentralized robotics systems.

Barfoot, Earon, and D'Eleuterio [1999, 2001a, 2001b] describe a decentralized robotics system for planetary space exploration. The focus of this work is to provide a general technology that will facilitate planetary network science. A network of robots could be used to deploy a very-low frequency array (VLFA) on the Moon to allow for collection of astrophysical observations using radio astronomy. Such an observatory will require a number of dipole units deployed over a region of a few hundred square kilometres. This concept was in fact studied by the International Space University [1993]. It could also enable seismology studies of an alien body which require sending a signal from one point on the surface to be read at several other points in order to analyze the material characteristics of the body. These and other examples of network science could be facilitated by a network of small mobile robots, similar to a colony of ants.

Certainly collective robotics is being looked at from a number of very different perspectives. We have detailed only a small portion of the current work in this field but hope these examples are representative in their complexity and depth. This is very much an active research field whose popularity seems to be growing very quickly.

## 4.2 A Decentralized Robotics System

A testbed facility (see Figure 4.1), the RISE (Robotics In Space Exploration) Network has been constructed[3]. This facility consists of six mobile robots which communicate with each other and a desktop computer through radio communications. The focus of RISE is on the autonomous control of such a network. Each robot has its own local computing facilities yet the group must work together to accomplish tasks.

Four DIRRSs (Digital InfraRed Range Sensors) point forward and return a distance to objects up to 1 metre ahead of each robot. Six whisker-style sensors (on/off) are located on the front, back and sides of each robot to sense collisions with obstacles.

Landmark navigation is used based on triangulation of position by observation of three lights. Each robot computes its own position using a navigation turret rather than simply having its position provided from an external source. A second navigation system based on odometry (there is an encoder on the rear axle) is used to update the position in between uses of the navigation turret. This local system provides fairly good data for a few metres whereupon the global system is employed once more.

Electromechanical components have been constructed using LEGO Technic$^{TM}$ pieces to enable rapid prototyping. Each robot has two DC motors, one to drive the back axle through an open differential and one to drive the front axle to enable four-wheel drive. There is a passive suspension system that

[3]See acknowledgements at the beginning of this document for construction credits.

enables all four wheels to remain in contact with the ground on slightly curved surfaces. The steering and navigation turret are driven by Airtronics model airplane servomotors.

The computational facilities of each robot consist of a 20 MHz Infineon *C*164 with 1 Mb of RAM and 1 Mb of flash-ROM (for program storage). The total power draw of the electronics is under 400 mA on average. The battery package consists of 8 NiMH (Nickel Metal Hydride) cells, size AA (1600 mAh). The robots can run for about 2 hours on a single charge.

Each robot is equipped with a radio modem (Radio Packet Controller (RPC) from Radiometrix). This allows the robots to communicate with each other and a base station desktop computer with bandwidth of 10 kbps[4]. The range of these modems is approximately 30 metres. All broadcasts are heard by all modems (each robot must "decide" autonomously whether or not to make use of the incoming information). The robots communicate with the base station for two reasons: issuing high-level commands (such as start, stop, pause) which affect the entire group and gathering any data that is measured by the robots both from their sensors and any potential payload instruments. The robots communicate with each other to share information about their environment and thus make it possible (or much easier) to solve cooperative tasks. Furthermore, they communicate in order to come to common decisions about what course of action to take.

## 4.3 Communication

In the RISE system, communication has been used in a variety of ways. There are protocols to allow the base station to issue high level commands to the network of robots. This allows the operator to select which task the group should attempt, to start/stop/pause the system, and to collect telemetry data for later analysis. There are also protocols to allow messages to be bounced from one robot to the next when out of range of the base station thus greatly extending the operating range beyond the range of a single radio modem[5]. Although these are crucial concerns for a practical design, the RISE robots do not require the base station to operate. The most important communications are *interrobot* (occurring between robots).

It is interesting to consider whether a centralized controller (running on the base station) could be used instead of the decentralized controllers (running on the robots). As discussed at great length in the previous chapter, this is theoretically possible, in fact equivalent in the world of mathematics, but here the finite communication bandwidth of the existing radio modems does not permit this to happen. Table 4.3 summarizes the total raw information flow to and from the decentralized controller of each robot. It is approximately 32.5 kbps. With 6 robots we require about 195 kbps of bandwidth even to consider using a centralized control model which is far beyond the measured capabilities of the existing communication system. Clearly the bottleneck is the data from the navigation sensor but to process this data locally presupposes some form of decentralization. Even if we did not require sending the navigation sensor data, we would require 3.0 kbps for 6 robots. This is close enough to the maximum of 10 kbps that serious packet collisions, bottlenecks, and delays would occur. In many control situations, tight

[4]Although the manufacturer claims 40 kbps (thousands of bits per second) as the theoretical maximum bandwidth, experimental testing found that in this application the number is closer to 10 kbps.

[5]Typically these protocols are not needed as the size of the workspace is smaller than the range of a single modem.

coordination between sensors and actuators is required (this is the essence of behaviour-based control). Delays of a few seconds here and there (since all robots much share the communication network) would seriously reduce performance of this system.

By contrast, the decentralized model (which is the one actually used) requires[6] only about 0.04 kbps of bandwidth per robot, or with 6 robots a total of 0.24 kbps which certainly is well below the experimental maximum for the radio modems. The most that robots must share (for the tasks considered here) are their locally computed $(x, y)$ position coordinates (32 bps) and another generic coordination variable, $u$ (8 bps). Everything else in Table 4.3 need not be communicated. Thus the decentralized model uses the communication facilities much more efficiently than the centralized model would for these tasks. Furthermore, communication is task dependent; only the data needed to solve the task is communicated. In the extreme, no communication is necessary. The centralized model is not able to take advantage of situations when data need not be shared between robots; it is task independent, always requiring communication of everything. These simple calculations precluded implementing the centralized model for performance comparisons.

As in Chapter 3, it will be useful to put interrobot communication into two categories for explanation purposes but we must recognize that at the implementation level these categories do not necessarily require different approaches. Both require the transmission of information by one robot and reception by one or more other robots. A communication channel may be treated both as a sensor [Matarić, 1998] and an actuator from a control point of view. When these mechanisms are represented as part of a behaviour-based controller, it is not always possible to distinguish them, nor need we [Brooks, 1991]. The two categories are

> SENSORY SHARING: Robots communicate their sensor inputs to one another in order to spatially extend the sensing capabilities of each robot.
>
> ACTUATOR COORDINATION: Robots communicate in order to "vote" on possible actions to take thus enabling coordination of actuator outputs.

It may again be more useful to think of these categories as the two ends of a spectrum of possibilities. It is unlikely that robots will share raw sensor data but rather information that has been preprocessed to some degree. For example, $(x, y)$ positions are shared rather than the raw navigation sensor data. Similarly, robots will likely not vote on what motor voltages to apply but something at a higher level (e.g., clustering location). When represented in a behaviour-based controller, where communication channels can exist throughout, it can become difficult to interpret these categories. Still, we maintain they are a useful construction when explaining the role of communication in decentralized mobile robotics.

Sensory sharing is the more basic of the two categories. In its simplest form, it involves relaying information directly from a sensor to a communication channel. Once the information is sent, the role of the sender is complete. The receiver(s) then treat(s) the communication channel as just another sensor.

Actuator coordination can be more involved. It is necessary when the operator does not provide enough information in the task description, resulting in more than one equally good solution to the problem (e.g., "Cluster in a corner, I don't care which corner." or "Move that object out of the way, I

[6] It requires this amount of bandwidth to perform optimally. No additional bandwidth would improve performance.

Table 4.1: Approximate flow of information occurring to and from the controller in each robot of the RISE system is 32.5 kbps.

| ITEM | WIDTH [bits] | RATE [Hz] | BANDWIDTH [bps] |
|---|---|---|---|
| DIRRS | 32 | 5 | 160 |
| Whiskers | 8 | 5 | 40 |
| Navigation Sensor | 160000 | 0.2 | 32000 |
| Battery Level | 8 | 5 | 40 |
| Odometer | 32 | 5 | 160 |
| Navigation Servo | 8 | 0.4 | 3.2 |
| Steering Servo | 8 | 5 | 40 |
| Front DC Motor | 2 | 5 | 10 |
| Rear DC Motor | 2 | 5 | 10 |
| TOTAL | | | ~ 32500 |

don't care where you put it."). Resolving this dilemma can involve repeated communication until some form of consensus is achieved as there is no central location to "count the ballots and announce the result". This presupposes that communications can occur fast enough to make the decisions before the robots must act. This is a classic example of a *two-timescale* system, a common concept in control engineering. The fast part of the system is that of the communication network and the slow part is comprised of the control of the electromechanical parts of the robots. If the communications can be made fast enough, the slow part of the system may be designed by assuming the fast part occurs instantaneously. This will be referred to as the *two-timescale assumption*[7]. The efficiency of communications afforded by decentralization often makes this assumption valid where it would not be for the comparable centralized control model. As more responsibility is shifted to the decentralized controllers, the efficiency of the communications increases along with the validity of the assumption of instantaneous communications (for a fixed bandwidth). For the robotics system and tasks (to be described) here, the two-timescale assumption is a very good one, thus greatly simplifying the slow timescale control design. However, it must be stressed that this assumption is highly dependent on the task at hand and the number of robots involved. The available bandwidth of communication enforces a constraint on the types of tasks that may be solved by a decentralized control system.

It is a valid argument that action coordination can be made unnecessary by providing more information in the task description. This was discussed in Section 3.6.2 on deterministic reactive controllers (Page 60). This point is not debated and in fact, providing more information should be encouraged as a general practice as it removes the need for some communication, thus allowing robots to behave more independently. However, we also recognize that a situation can conceivably arise where a more detailed task description is *not* possible. For example, different robots may be required to take on different roles to successfully complete the task but the operator may not wish to assign these roles explicitly; the roles should be assigned by the robots themselves. Having the capability to leave some decisions up to the robots is very desirable if the system is to be at all flexible. This in fact goes to the very heart of

[7]The assumption of a centralized control model is essentially this very assumption, as discussed in the introduction. Here, however, it is generalized to the decentralized case as well.

"intelligence". If every detailed action must be given a priori, how can we expect the system to ever be intelligent? The process of decentralized decision making is very much a part of the intelligence we are expecting from these types of systems. We find the term *self-organizing* to be appropriate in describing a decentralized system which is able to coordinate its actions in the face of more than one equally good solution. Decentralized decision making will be discussed in depth in the next chapter.

Based on the two types of interrobot communication described above, we have four classes of collective robotics tasks which were introduced in Section 3.6.1 (Page 59). These classes allow us to categorize all collective robotics tasks based on the type of communication necessary for success.

**Definition.** COLLECTIVE ROBOTICS TASK CLASSES:

- **I** Tasks requiring neither *sensory sharing* nor *actuator coordination.*
- **II** Tasks requiring only *sensory sharing.*
- **III** Tasks requiring only *actuator coordination.*
- **IV** Tasks requiring both *sensory sharing* and *actuator coordination.*

## 4.4 Generic Communication Model

This section presents two generic tools which may be incorporated into most types of control but in particular behaviour-based control. The first module enables sensory sharing and may be seen in Figure 4.2 (left). The second, depicted in Figure 4.2 (right), builds on the first to enable action coordination.

The sensory sharing module is very simple and may be used to share any type of datum. It should be run on each of the robots in a network. In the diagram, $(x, y)$ position information is shared as an example. Robots broadcast a data packet consisting of their current position and their identification number at a rate of approximately 1 Hz. The modem receives packets of data which are immediately placed in a receiving buffer. A routing algorithm takes the packets from this buffer and writes them to the appropriate data slot based on the robot identification number. Old data is overwritten by new data. This data is then used by various control modules (not to be described here). In the case of navigation data, each robot's position information is written to its own data slot not by the modem but by the various navigation modules. This data is read every second and placed as a packet (with the identification number) in the transmission buffer. The modem then sends out this packet. It should be stressed that there is no particular sequence of operations required. All submodules operate in a decentralized, concurrent manner. If new data is available, it is used. Otherwise, the old data is used. The robot does not stop and wait for data in any circumstance.

The action coordination module adds one simple module to the sensory sharing tool, a stochastic coordination mechanism. The purpose of this box is to try and make a coordination variable, $u$, the same for all robots. It inputs the $u$'s from the other robots (and itself) and outputs a new value for its own $u$. Let the size of $u$ be $m$ bits. This module allows a group of robots to generate a common piece of information of $m$ bits in length in a completely decentralized fashion. Chapter 5 will describe a stochastic coordination mechanism for a sparsely[8] connected decentralized system. For the purposes of

[8]Each robot only communicates with those robots within a certain radius. This situation arises due to finite communication range.

this chapter, in which we have a fully[9] connected decentralized system, the following simple algorithm works well (it is a simple case of the algorithm to be presented in Chapter 5).

**Definition.** (FULLY CONNECTED) STOCHASTIC COORDINATION MECHANISM:

STEP 1: Define set of possible values for $u$.

STEP 2: Randomly pick a new $u$ from set and transmit selection.

STEP 3: Receive votes ($u$'s) from others, and determine winner(s) based on most common value. Note, there can be a tie thus multiple winners.

STEP 4: Redefine set of possible values to be those of the winner(s).

STEP 5: Return to step 2.

This algorithm may be run ad infinitum and can exist as a small part of each robot's behaviour-based controller (or any other type of controller). It should continually provide as output the action the robot is to follow (selected from the instantaneous set of possible actions which shrinks over time). Thus voting and action-following occur concurrently not sequentially, making the mechanism compatible with decentralized/behaviour-based control. Robots may join or leave the task and the algorithm will adjust automatically.

As indicated in the name, this coordination mechanism is stochastic in nature. Essentially when there is a tie between two possible results, another vote must occur[10]. It is not possible to guarantee (with probability 1) that a consensus has been reached after a certain time. However, we may compute what the probability of consensus is after $t$ updates. For example, consider the case where two voters are trying to agree on a single bit. Let $\boldsymbol{u}_1$, $\boldsymbol{u}_2 \in {}^2\mathbb{S}$ represent the decisions of two voters. Then let $\boldsymbol{u} \in {}^4\mathbb{S}$ be their combined (global) decision such that

$$\boldsymbol{u} = \mathbf{B}_1{}^T\boldsymbol{u}_1 \oplus \mathbf{B}_2{}^T\boldsymbol{u}_2 \qquad \mathbf{B}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad \mathbf{B}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \tag{4.1}$$

with $\mathbf{B}_1$, $\mathbf{B}_2 \in {}^2\mathbb{D}^4$. For consensus we would like to have either $\boldsymbol{u} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$ or $\boldsymbol{u} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$ such that $\mathbf{u}_1 = \mathbf{u}_2$ where the individual decisions are now deterministic. Note that $\boldsymbol{u} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^T$ or $\boldsymbol{u} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T$ corresponds to $\mathbf{u}_1 \neq \mathbf{u}_2$, or no consensus. At the ensemble level, the global decision updates according to the following Markov chain.

$$\boldsymbol{u}[t+1] = \begin{bmatrix} 1 & \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} & 1 \end{bmatrix} \boldsymbol{u}[t] \tag{4.2}$$

If we assume that $\boldsymbol{u}_1[0] = \boldsymbol{u}_2[0] = \omega$ and thus $\boldsymbol{u}[0] = \omega$ we have

$$\boldsymbol{u}[1] = \downarrow \begin{bmatrix} 3 \\ 1 \\ 1 \\ 3 \end{bmatrix} \quad \boldsymbol{u}[2] = \downarrow \begin{bmatrix} 7 \\ 1 \\ 1 \\ 7 \end{bmatrix} \quad \boldsymbol{u}[3] = \downarrow \begin{bmatrix} 15 \\ 1 \\ 1 \\ 15 \end{bmatrix} \quad \cdots \quad \boldsymbol{u}[t] = \downarrow \begin{bmatrix} 2^{t+1}-1 \\ 1 \\ 1 \\ 2^{t+1}-1 \end{bmatrix} \quad \cdots \quad \lim_{t\to\infty} \boldsymbol{u}[t] = \begin{bmatrix} \frac{1}{2} \\ 0 \\ 0 \\ \frac{1}{2} \end{bmatrix} \tag{4.3}$$

[9]Each robot communicates with all other robots.

[10]This is analogous to the "rock, paper, scissors" method from everyday experience.

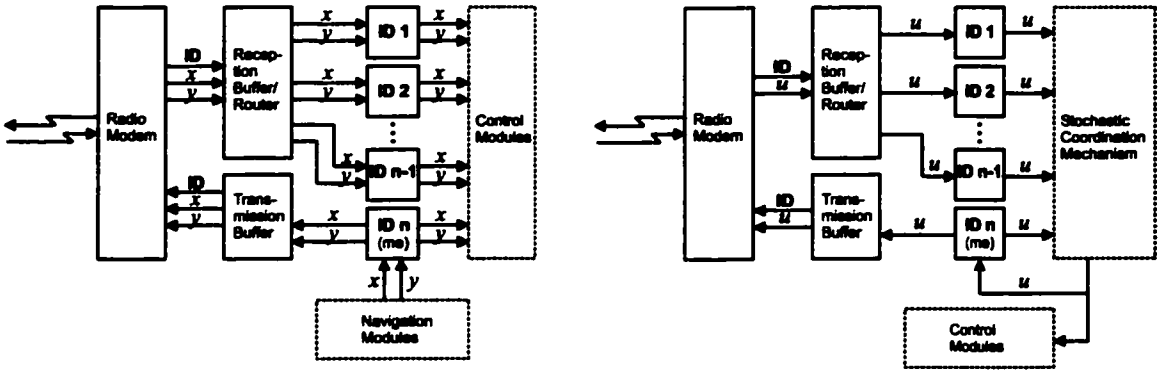



Figure 4.2: (left) SENSORY SHARING MODULE: Incoming data packets consisting of some data to be shared (e.g., $(x, y)$ coordinates) and the sender's ID number are buffered and then routed to the appropriate storage slot for that ID. This data may then be used in other modules such as those for control. The robot running the algorithm writes to its own storage slot (marked "me") not from the modem but a navigation module. (right) ACTION COORDINATION MODULE: Virtually identical to sensory sharing but a *stochastic coordination mechanism* is used which allows robots to generate a piece of common information, $u$, in a decentralized manner. The agreed upon $u$ may then be used in other modules such as those for control.

so that the probability of consensus after $t$ time-steps is

$$(2^{t+1} - 1) + (2^{t+1} - 1) = 1 - \left(\frac{1}{2}\right)^t \tag{4.4}$$

If this algorithm runs at 1 Hz then after 10 seconds there is a consensus probability of 0.999. After 20 seconds it is 0.999999. With more voters the consensus will (on average) actually be reached sooner as the probability of an exact tie gets smaller. With 4 voters trying to agree on a single bit the probability of a consensus after $t$ updates is $1 - \left(\frac{3}{8}\right)^t$. With 6 voters it is $1 - \left(\frac{5}{16}\right)^t$. Similar analyses may be employed for decisions larger in size than a single bit.

## 4.5 Experimental Examples

Table 4.5 gives a summary of tasks currently implemented on the RISE system with the type and content of communication required to solve the task. In each case the sensory sharing and action coordination modules described above were used as part of a behaviour-based controller. Robots shared $(x, y)$ position data with one another and coordinated such actions as: in which of 4 corners to cluster (2 bits) or which of the 5 robots to become flock leader (3 bits).

There are four main task categories (some having subtasks). All tasks involve spatial configurations of the robots. *Clustering*, *distributing*, and *formation* involve achieving a static configuration then stopping while *flocking* requires a dynamic configuration (changes with time). Each main category is described briefly.

CLUSTERING: Requires robots to pack themselves tightly into one of the four corners of the workspace from an initially random configuration. In the various subtasks the active corner can be

Table 4.2: Summary of collective robotics tasks currently implemented on RISE system.

| TASK | | CLASS | DESCRIPTION | SENSORY SHARING | ACTION COORDINATION | FIG |
|---|---|---|---|---|---|---|
| Clustering | (a) | I | Cluster in prespecified corner | — | — | 4.3 |
| | (b) | II | Cluster in mean closest corner | coordinates | — | — |
| | (c) | III | Cluster in random corner | — | vote on corner | — |
| | (d) | IV | Cluster in either of two mean closest corners | coordinates | vote on corner | — |
| Distributing | (a) | I | Distribute over two prespecified corners, division prespecified | — | — | — |
| | (b) | II | Distribute over two prespecified corners, division based on proximity | coordinates | — | 4.4 |
| | (c) | III | Distribute over random two corners, division prespecified | — | vote on corners | — |
| | (d) | IV | Distribute over random two corners, division based on proximity | coordinates | vote on corners | — |
| Formation | | IV | Form an $K$-sided regular polygon with robots at vertices, center and radius prespecified, any orientation | coordinates | vote on orientation | 4.5 |
| Flocking | (a) | I | Follow a prespecified leader, leader drives in large clockwise circle | — | — | — |
| | (b) | II | Follow robot who has biggest gap ahead, leader drives in large clockwise circle | coordinates | — | — |
| | (c) | III | Follow an elected leader, leader drives in large clockwise circle | — | vote on leader | 4.6 |
| | (d) | IV | Same as (b) but vote between clockwise/counterclockwise direction | coordinates | vote on direction | — |

(a) prespecified, (b) based on mean distance (of all robots) to corners, (c) voted upon, (d) voted upon from two mean closest corners.

DISTRIBUTING: Similar to clustering except roughly half the robots pack tightly into one corner and the other half into another. In subtasks (a) and (b) the two active corners are prespecified while in (c) and (d) they are voted upon. In subtasks (a) and (c) the division into two groups is prespecified while in (b) and (d) it is based on proximity to the two active corners (tries to minimize travel distance of robots).

FORMATION: Requires robots to form an $K$-sided regular polygon (where $K$ is the number of robots) with robots at vertices, from an initially random configuration. The center and radius of the polygon are prespecified but the orientation must be decided upon amongst robots.

FLOCKING: Requires robots to travel in a tight pack around a large circle in the workspace. In the various subtasks the leader is (a) prespecified, (c) voted upon, (b) or (d) based on which robot is ahead of the pack. The leader drives around the large circle. Robots who are not the leader drive towards the leader. In subtasks (a), (b) and (c) the direction of travel is clockwise while in (d) robots vote between clockwise and counterclockwise.

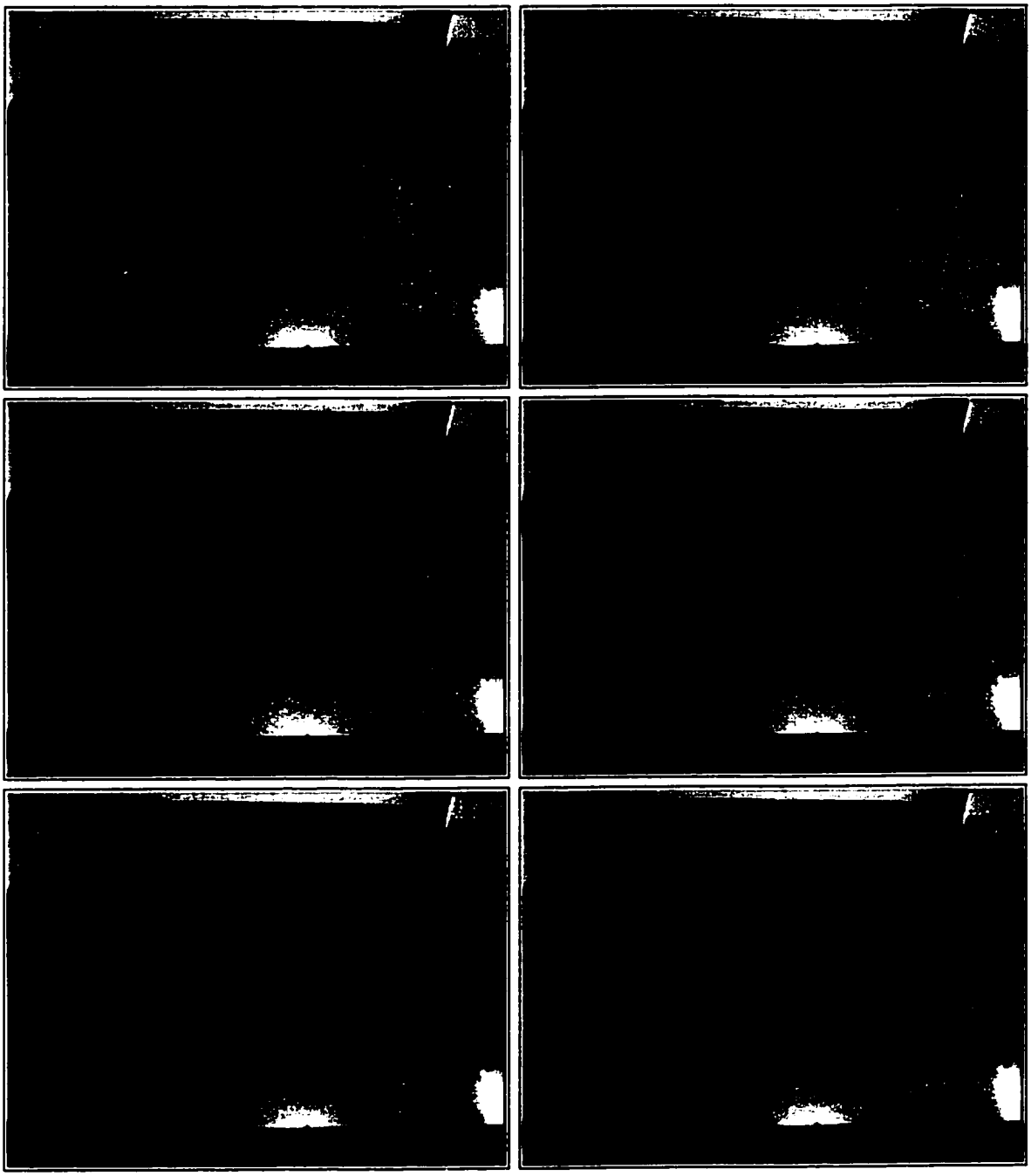


Figure 4.3: CLUSTERING (a): Six time-lapsed frames of 5 robots forming a cluster in the marked box. In this version of the task, no communication was necessary but in others the robots must come to a common decision as to the cluster location.

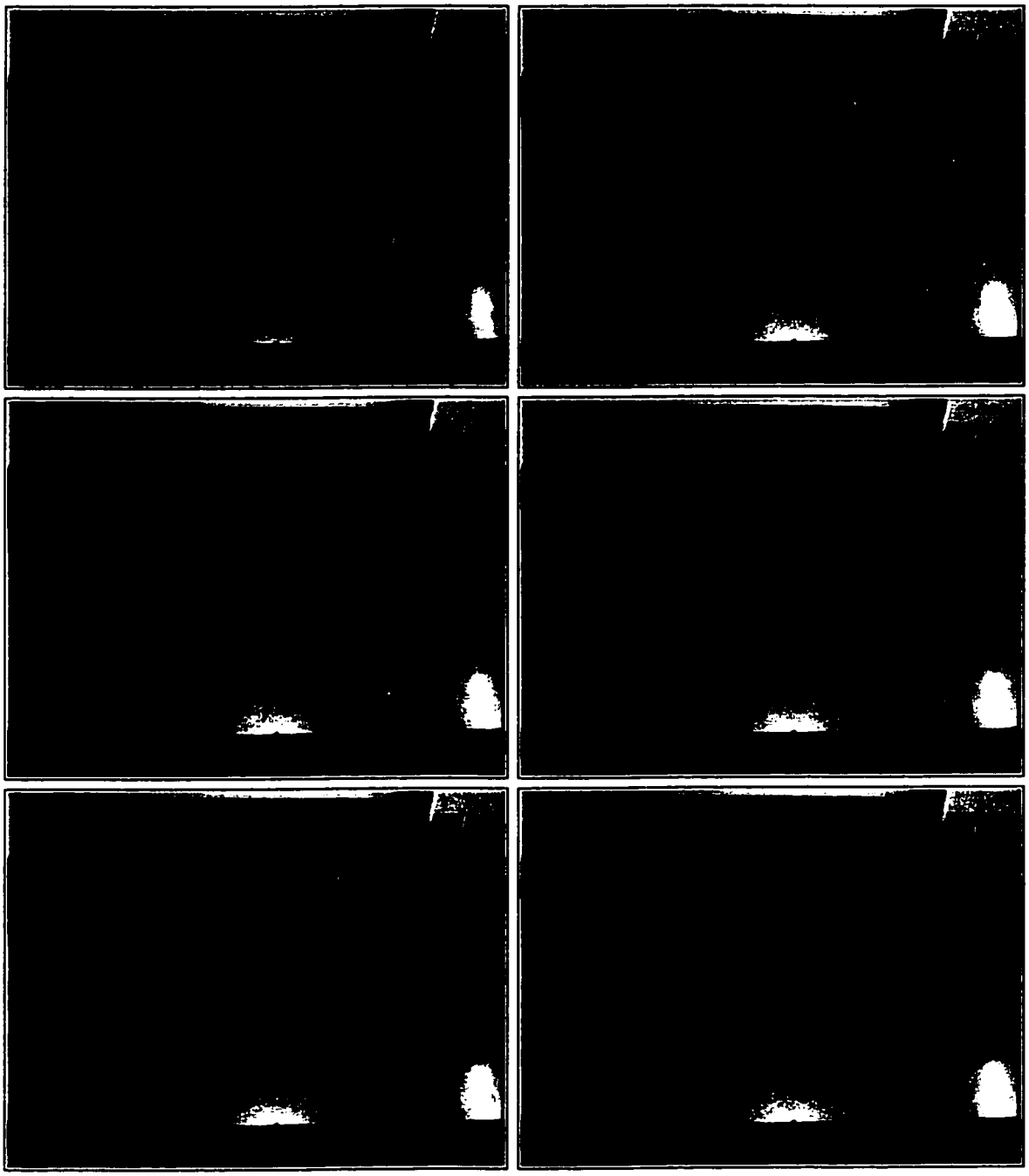


Figure 4.4: DISTRIBUTING (b): Six time-lapsed frames of 5 robots distributing themselves into the two marked boxes. The robots must share their position information with one another in order to coordinate which robots will head to which boxes (the division is based on proximity).

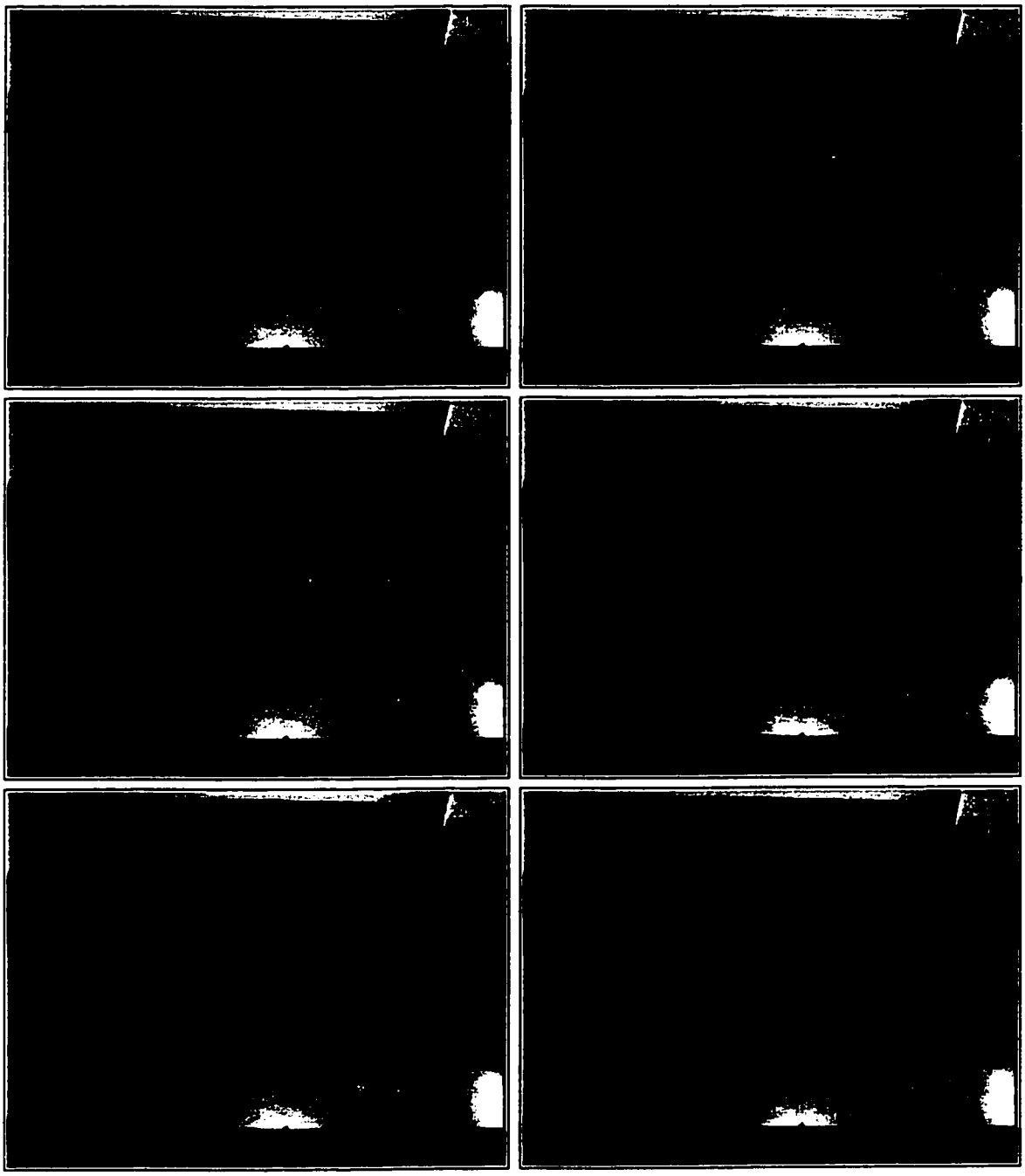


Figure 4.5: FORMATION: Six time-lapsed frames of 5 robots forming a pentagon with vertices on the marked circle. Here the robots must continually share position information in order to space themselves out relatively on the circle. Neither the specific location of each robot nor the orientation of the resulting pentagon is prespecified. The robots "decide" how to form the shape.

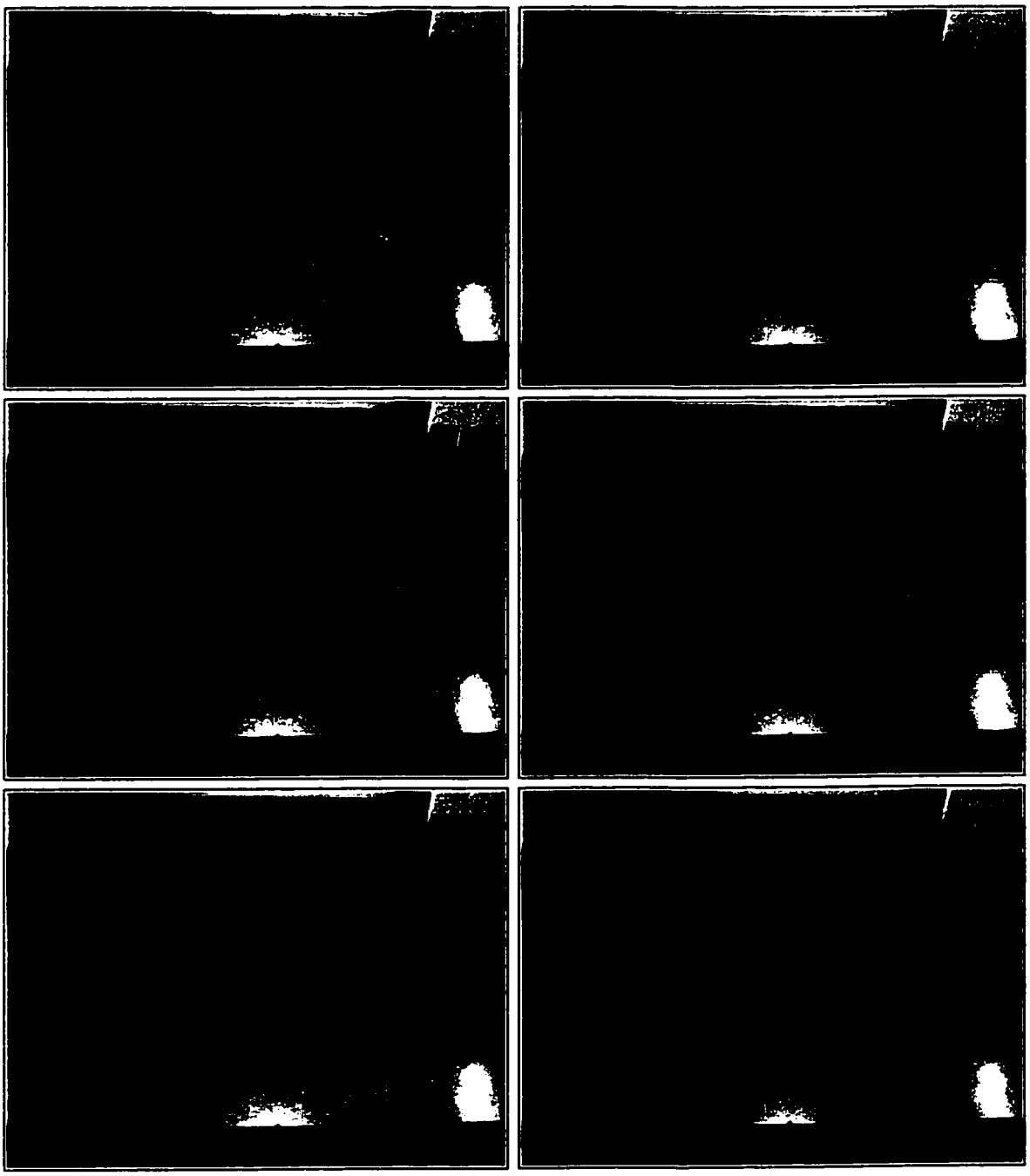


Figure 4.6: FLOCKING (c): Six time-lapsed frames of 5 robots flocking around the marked circle (1 revolution shown). Here the robots must vote on who amongst them should be the leader of the flock and then follow that elected leader around the circle. This task is very dynamic and difficult. Notice the flock spreads out quite a bit (frames 3,4) but recovers gracefully.

Figures 4.3 through 4.6 depict one subtask from each of the main task categories. All tasks take on the order of a few minutes to complete from an initially random configuration of robots. In the case of flocking, completion means achieving steady-state behaviour.

In Figure 4.3, robots cluster in a corner prespecified by the operator. Here no interrobot communication is required. Other versions of the tasks have the robots voting on the corner in which to cluster. Figure 4.4 shows the robots distributing themselves between two different corners. The corners are again prespecified by the operator but the robots decide amongst themselves which should go to each corner. This was done based on distance to the corners and thus communication of $(x, y)$ positions was required. The two robots starting on the left side move to left-hand corner while the three starting on the right side move to the right-hand corner. In Figure 4.5 the robots deploy themselves in a pentagonal formation. They decide amongst themselves the order of robots on the circle and what orientation the resulting pentagon should have. This is again done based on various distances which require the communication of $(x, y)$ positions. The algorithm is designed to accommodate any number of robots and will form an $K$-sided polygon with $K$ robots.

Figure 4.6 shows the robots flocking around a circle. Here they have chosen a leader using the decentralized voting scheme described above. The elected robot drives around the circle and the others drive towards it. This is an example of ***spontaneous task division***, one robot assuming the role of leader and the others following. Having the robots assign themselves different roles is very important capability in collective robotics. In this way as robots join or leave the group, perhaps through malfunction, the different roles may be reassigned dynamically such that the task always gets completed. In the flocking task here if the current leader is removed or malfunctions, another robot will be elected leader and the group will continue to perform the task.

Clustering and distributing are the most reliably successful tasks, followed by formation and finally flocking. Formation is a highly coupled task; a slight movement in the position of one robot causes the others to compensate in order to maintain the regular polygon shape. In some cases it can take a long time for the robots to settle into a stable configuration but usually it happens quite quickly (a few minutes).

The inherent difficulty in flocking arises from the fact that it is dynamic. It was found that the battery levels played a large role in the success of this task. When batteries drained unevenly (which typically does happen), the robots travelled at different maximum speeds, making it difficult to remain in a tight pack. Spreading of the pack may be seen in Figure 4.6 but the system is able to gracefully recover.

## 4.6 Summary

Radio communication (or any other well established protocol) allows very tightly coupled collective robotics tasks to be accomplished with a reduced burden on traditional sensors and computational facilities. This was demonstrated through various tasks involving five mobile robots communicating with radio modems.

The tasks in Table 4.5 from classes II, III, and IV would not be possible (or would be a great deal more difficult) without well established interrobot communication protocols. As discussed above, to resort to

some form of centralized control would require a communication network to input sensor information to and output actions from the centralized controller. Here we find that once a communication network is in place, there is no reason to use centralized control. This verifies (in this context) the claim that decentralized controllers with the two types of interrobot communication described above can perform exactly the same tasks as a centralized controller. In fact, in many cases (e.g., class I tasks), it is a great deal more efficient to use decentralized control as less data needs to be broadcast over the communication network. For the RISE system, a purely centralized controller would not be viable due its inefficient use of the limited bandwidth communications.

Interrobot communication can be roughly placed into two categories: *sensory sharing* and *action coordination*. The first allows the sensing capabilities of each robot to be spatially extended while the second allows robots to globally coordinate actions through some form of voting. Both types of interrobot communication are compatible with a decentralized philosophy and can easily be implemented in behaviour-based controllers using generic communication modules. The difficulty of collective robotics tasks can be judged by the type and amount of interrobot communication required for success.

It is convenient to speak of the classes of collective robotics tasks defined above but we must stress that it will not always be possible to clearly place a task into one of these categories. The tasks described here were deliberately designed to make them easy to categorize. The classes are based on the type of communication required which, as discussed earlier, can become difficult to judge at the implementation level. For example, at a high level the communicated information between robots was in a representation easily recognized by a human (e.g., $(x, y)$ coordinates) but at the digital radio packet level it is unrecognizable. Despite this, we find that the general notion of gauging the difficulty of collective robotics tasks by the nature of the interactions required between robots to have great merit.

One of the most reliable parts of the RISE system is the communication framework. The digital message packets are rarely missed and have not been observed to convey noisy information to date. Radio modem appears to be a very robust method of implementing interrobot communication. At a transmission rate of approximately 1 Hz, even if a packet were to be missed by one rover, it will most likely receive the next one. The robots are not able to move far enough in such a short time period for this to have much effect. The *two-timescale* assumption appears to be valid for this system allowing for highly reliable sensory sharing and action coordination on all tasks.

One concern in collective robotics is the stopping issue. If control is decentralized, how do the robots know when the task is finished? There are a few ways to look at this. First, it may not be necessary for them to actually stop executing the algorithm; the task would carry on indefinitely. Secondly, stopping can be built into the algorithm as in the formation task described above; once the desired shape is made the robots stop moving even though they continue to execute the algorithm. The important point to make is that communication allows any centralized task to be carried out in a decentralized fashion. If stopping is itself considered to be a task then this can also be achieved by a collection of robots which communicate.

The design of collective behaviours in a group of robots is easily facilitated by the robust, bottom-up, decentralized approach advocated here. It is hoped this study of interrobot communication will push the boundaries of collective robotics research while at the same time helping to provide insights into the biological communication protocols by which we are inspired.

The field of Artificial Life is unabashedly mechanistic and reductionist. However, the new mechanism—based as it is on multiplicities of machines and on recent results in the fields of nonlinear dynamics, chaos theory, and the formal theory of computation—is vastly different from the mechanism of the last century.
—Christopher G Langton
ARTIFICIAL LIFE, 1989

# Chapter 5

# SELF-ORGANIZATION

This chapter describes a coordination mechanism for a system of communicating agents with sparse connections [Barfoot and D'Eleuterio, 2001]. This is a very general tool which may be used to make decentralized decisions. The mechanism is based on a stochastic version of cellular automata (SCA). A parameter analogous to a "temperature" can be tuned to vary the behaviour of the system. It is found that the best temperature for coordination occurs near a phase transition between order and chaos. By design, coordination does not rely on any particular structure of the connections between agents, thus it may be applicable to a large array of sparsely communicating mobile robots or software agents.

As we have seen in previous chapters, the term *decentralized system* encompasses large bodies of work from engineering, computer science, and mathematics. Examples include networks of mobile robots [Matarić, 1992], software agents [Bonabeau et al., 1994], and cellular automata [Wolfram, 1984]. A common thread in all decentralized systems is the issue of *coordination*. How is a large number of *sparsely* communicating agents able to produce a coherent global behaviour using simple rules? Answering this question will not only permit the construction of interesting and useful artificial systems but may allow us to understand more about the natural world. Ants and the other social insects are perfect examples of local interaction producing a coherent global behaviour. It is possible for millions of ants to act as a *superorganism* through local pheromone communication. We seek to reproduce this ability on a fundamental level in order to coordinate artificial systems.

It can be argued that *cellular automata* (CA) are the simplest example of a mathematical decentralized system. Originally studied by von Neumann [1966], the term CA is used to describe systems of sparsely coupled difference equations. Despite their simple mechanics, some extremely interesting

behaviours have been catalogued (e.g., Conway's *Game of Life*). The word *self-organization* is used in many contexts when discussing decentralized systems, which can lead to confusion. Here we use it to mean *global coordination in the face of more than one alternative.* We will be describing a stochastic version of cellular automata. The goal will be to have all cells choose the same symbol from a number of possibilities using only sparse communication. We maintain that rules able to succeed at this task are self-organizing because the cells are not told which symbol to choose, yet they must all coordinate their choices to produce a globally coherent decision. If we told the cells which symbol to choose, the task would be very easy and no communication between cells would be necessary. This can be dubbed *centralized organization* and is in stark contrast to *self- or decentralized organization.* This notion is paralleled in nonlinear physics. At the risk of an over-simplified explanation, in Rayleigh-Bénard convection regular hexagonal convection cells are able to form in a thin fluid heated from below. However, the orientation of these cells is not coordinated by any central agency. We believe that coordination in the face of more than one alternative is at the heart of all decentralized systems. This chapter seeks to reproduce this phenomenon in as simple a model as possible.

The difference between centralized and decentralized decision making is already familiar to most people but likely not thought of in this way. It is essentially the difference between flipping a coin and using "rock, paper, scissors"[1] when two people are trying to resolve a dilemma. In flipping a coin, both parties rely on the coin (a centralized agency) to make the decision and then abide by its ruling. In "rock, paper, scissors" a decision is also quickly achieved if both people follow the same rules, but it occurs in a decentralized manner. Note, however, in "rock, paper, scissors" it can take repeated trials to finally come to a consensus as there can be a tie. In fact, there is no guarantee that a consensus will ever be achieved. But, we may compute the (high) probability with which a decision is made after a certain time (as was done with the fully connected voting scheme of Chapter 4). Decentralized decision-making is unavoidably stochastic in nature.

## 5.1 Related Work

To avoid repetition, this review will not address the control and robotics literatures described in Chapters 3 and 4 but bear in mind their relevance on the current discussion. To contrast these views, relevant studies from artificial life (AL) are described. These studies typically operate on large numbers (hundreds to thousands) of agents which must coordinate themselves in some manner. Some mention is also made of other work in decentralized coordination.

In the following, note that typically cellular automata do not operate in a stochastic but rather a deterministic manner. Unless explicitly stated (e.g., stochastic cellular automata), the term cellular automata (CA) will imply determinism.

von Neumann [1966] originally studied cellular automata in the context of self-reproducing mechanisms. The goal was to devise local rules which would reproduce and thus spread an initial pattern over a large area of cells, in a tiled fashion. The current work can be thought of as a simple case of this where the tile size is only a single cell but there are multiple possibilities for that tile. Furthermore, we wish

[1] On the off chance that the reader has never heard of this, please turn to the person next to you and ask what this means.

our rules to work starting from any random initial condition of the system.

Cellular automata were categorized by the work of Wolfram [1984] in which four *universality classes*[2] were identified. All rules were shown to belong to one of class I (fixed point), class II (oscillatory), class III (chaotic), or class IV (long transient). These universality classes can also be identified in SCA and we will show that in our particular model, choosing a parameter such that the system displays long transient behaviour (e.g., class IV) results in the best performance on our decentralized coordination task.

Langton [1990, 1991] has argued that natural computation may be linked to the universality classes. It was shown that by tuning a parameter to produce different CA rules, a phase transition was exhibited. The relation between the phase transition and the universality classes was explored. It was found that class IV behaviour appears in the vicinity of the phase transition. The current work is very comparable to this study in that we also have a parameter which can be tuned to produce different CA rules. However, our parameter varies the amount of randomness that is incorporated into the system. At one end of the spectrum, completely random behaviour ensues while at the other completely deterministic behaviour ensues (which is simple voting). We also relate the universality classes to particular ranges of our parameter and find a correlation between performance on our decentralized coordination task and class IV behaviour. We attempt to use similar statistical measures to Langton [1990] to quantify our findings.

Mitchell et al. [1993] and Das et al. [1995] study the same coordination task as will be examined here in the case of deterministic CA. However, their approach is to use a genetic algorithm to evolve deterministic rules successful at the task whereas here hand-coded stochastic rules are described. They found that the best solutions were able to send long range *particles* (similar to those in the *Game of Life*) [Andre et al., 1997] in order to achieve coordination. These particles rely on the underlying structure of the connections between cells, specifically that each cell is connected to its neighbours in an identical manner. The current work assumes that no such underlying structure may be exploited and that the same mechanism should work for different connective architectures. The cost for this increased versatility is that the resulting rules are less efficient (in terms of time to coordinate) than their particle-based counterparts.

Tanaka-Yamawaki et al. [1996] study the same problem to that considered here. They use *totalistic* [Wolfram, 1984] rules which do not permit exploitation of the underlying structure of the connections between cells but rather rely on the intensity of each incoming symbol. They also vary a parameter to produce different rules and find that above a certain threshold, "global consensus" occurs but below it does not. The connectivity between cells is regular and success was found to depend in part on the connective architecture used. However, they consider large clusters of symbols to be a successful global consensus. We do not and thus turn to a stochastic version of their totalistic rules in an attempt to destroy these clusters and complete the job of global coordination. Barfoot and D'Eleuterio [2001] describe a subset of the results presented here.

The essential idea used in the model to be described here has been borrowed from nonlinear physics. To force the system away from a uniform distribution, $\omega$, of symbols (which may be viewed as a stable

[2]These are entirely different from the classes described in Chapter 3.

equilibrium), an instability is introduced in the local update rule of each decision maker. This local instability drives the system away from a uniform distribution and as we will see towards global consensus. The notion of an instability forcing a system far-from-equilibrium thus creating "order" has been seen in nonlinear physics [Nicolis and Prigogine, 1977] and chemistry [Nicolis and Baras, 1984]. At the single trajectory level, an instability may be seen as breaking symmetry between more than one alternative while at the ensemble level, symmetry is once again restored as depicted in Figure 5.2.

The use of instabilities in the coordination of artificial systems is not new. Haken et al. [1973, 1983, 1984, 1987, 1994] lead a movement to describe all structure in nature in these terms and exploit these ideas in the design of artificial systems[3]. Instabilities have even been used to describe coordination in the wavelength population dynamics of lasers [Graham and Wunderlin, 1987].

## 5.2 Stochastic Cellular Automata

In deterministic cellular automata there is an *alphabet* of $m$ symbols, one of which may be adopted by each cell. Each incoming connection provides a cell with one of these symbols. The combination of all incoming symbols uniquely determines which symbol the cell will display as output. Stochastic cellular automata (SCA) work in the very same way except at the output level. Instead of there being a single unique symbol which is adopted with probability 1, there can be multiple symbols adopted with probability less than 1. Based on this *outgoing probability distribution* over the $m$ symbols, a single unique symbol is drawn to be the output of the cell. This is done for all cells simultaneously. It should be noted that deterministic CA are a special case of SCA.

We consider a specific subcase of SCA in this paper which corresponds to the totalistic rules of CA. Assume that cells cannot tell which symbols came from which connections. In this case, it is only the intensity of each incoming symbol which becomes important. Furthermore, we desire that our rules work with any number of incoming connections thus rather than using the number of each of the incoming $m$ symbols, we use this number normalized by the number of connections which can be thought of as an *incoming probability distribution* over the $m$ symbols. In summary the model we consider is as follows.

**Definition.** TOTALISTIC SCA: Consider a system of $K$ cells, each of which is connected to a subset of the other cells. Let ${}^m\mathbb{D}$ represent the alphabet of $m$ symbols. The state of Cell $k$ at time-step $t$ is $\mathbf{u}_k[t] \in {}^m\mathbb{D}$. The input probability distribution, $\boldsymbol{p}_{\text{in},k} \in {}^m\mathbb{S}$, for Cell $k$ is given by

$$\boldsymbol{p}_{\text{in},k}[t] = \boldsymbol{\sigma}_k\big(\mathbf{u}_1[t], \ldots, \mathbf{u}_K[t]\big) \tag{5.1}$$

where $\boldsymbol{\sigma}_k \in {}^m\mathbb{F}^{mK}$ accounts for the connections of Cell $k$ to the other cells. The output probability distribution $\boldsymbol{p}_{\text{out},k} \in {}^m\mathbb{S}$ is given by the map, $\boldsymbol{\pi}_k \in {}^m\mathbb{F}^m$,

$$\boldsymbol{p}_{\text{out},k}[t] = \boldsymbol{\pi}_k\big(\boldsymbol{p}_{\text{in},k}[t]\big) \tag{5.2}$$

The probability distributions $\boldsymbol{p}_{\text{in},k} \in {}^m\mathbb{S}$ and $\boldsymbol{p}_{\text{out},k} \in {}^m\mathbb{S}$ are stochastic columns. The new state of Cell $k$ at time-step $t+1$ is randomly drawn according to the distribution $\boldsymbol{p}_{\text{out},k}[t]$ and is represented by $\boldsymbol{u}_k[t+1] \in {}^m\mathbb{D}$.

[3]For some reason these references are rarely cited but in the opinion of this author, undeservedly so.

It should be noted that in (5.1) if the connections between the cells are not changing over time then the functions, $\sigma_k(\cdot)$, will not be functions of time. However, we could allow these connections to change which would make them functions of time. Once the connections are described through the $\sigma_k(\cdot)$ functions, the only thing that remains to be defined is the $\pi$-map. We assume that each cell has the same $\pi$-map but this need not be the case.

## 5.3 Self-Organization as a Control Problem

It should be no surprise that the above self-organization problem (a.k.a., the coordination problem, decentralized decision making problem) may be cast as a (nonlinear) control problem in the following way

$$\begin{aligned} \boldsymbol{x}[t+1] &= (\boldsymbol{u}_1[t], \boldsymbol{u}_2[t], \ldots, \boldsymbol{u}_K[t]) && (5.3) \\ \boldsymbol{y}_k[t] &= \sigma_k(\boldsymbol{x}[t]) \quad \forall k = 1 \ldots K && (5.4) \\ \boldsymbol{u}_k[t] &= \pi_k(\boldsymbol{y}_k[t]) \quad \forall k = 1 \ldots K && (5.5) \end{aligned}$$

where $\boldsymbol{x} \in {}^{m^K}\mathbb{S}$ is the complete (global) system state, $\boldsymbol{u}_k \in {}^{m}\mathbb{S}$ are the local controls, $\boldsymbol{y}_k \in {}^{m}\mathbb{S}$ are the local observations, $\sigma_k \in {}^{m}\mathbb{F}^{m^K}$ are the (nonlinear) observation projection functions (as described in the previous section), and $\pi_k \in {}^{m}\mathbb{F}^{m}$ are the (possibly nonlinear) feedback control laws. Note in the system state equation (5.3), $\boldsymbol{x}[t+1]$, is a joint distribution (see Section 2.14) of the local controls, $\boldsymbol{u}_k[t]$. Everything in this system is given except the control laws, $\pi_k$, which we must design. The goal of this control problem is to make the $m$ points

$$\boldsymbol{x}_1^* = \left( \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right), \ldots, \boldsymbol{x}_m^* = \left( \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \right) \tag{5.6}$$

stable while making all others unstable. Thus beginning from a random initial condition the system will tend towards one of these $m$ points and stay there. The next section discusses the design of control laws for the decentralized decision making problem.

## 5.4 Control Laws

We have already seen the successful application of a control law to the decision making problem in Chapter 4. It is tempting to apply the (fully connected) stochastic coordination mechanism of Section 4.4 to the more general problem of *sparsely* connected cells. In the following example, we will see why this does not always work.

**Example.** Consider a network of $K = 4$ cells sparsely connected to one another as in Figure 5.1. The alphabet size will be taken to be $m = 2$. There are $2^K = 2^4 = 16$ possible global states, $\boldsymbol{x} \in {}^{16}\mathbf{S}$, only two of which we find to be satisfactory (i.e., when all cells take on the same symbol). That is,

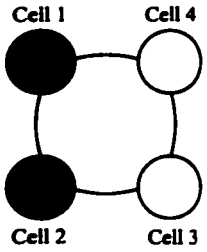



Figure 5.1: Example of a SCA model with $K = 4$ cells. Lines indicate (sparse) connections between cells. In this example the $m = 2$ symbols of the alphabet are represented by the 2 colours. In this state we have two clusters and hence no global consensus.

we want to make the two points

$$x_1^* = \left(\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}\right) \qquad x_2^* = \left(\begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}\right)$$

stable but no others. This will ensure only a single cluster will form and a global consensus is reached. The local controls are

$$u_1 = \begin{bmatrix}u_{11}\\u_{12}\end{bmatrix} \qquad u_2 = \begin{bmatrix}u_{21}\\u_{22}\end{bmatrix} \qquad u_3 = \begin{bmatrix}u_{31}\\u_{32}\end{bmatrix} \qquad u_4 = \begin{bmatrix}u_{41}\\u_{42}\end{bmatrix}$$

The observation projection functions are given as

$$\sigma_1(u_1, \ldots, u_K) = \downarrow\begin{bmatrix}u_{11} + u_{21} + u_{41}\\u_{12} + u_{22} + u_{42}\end{bmatrix} \qquad \sigma_2(u_1, \ldots, u_K) = \downarrow\begin{bmatrix}u_{11} + u_{21} + u_{31}\\u_{12} + u_{22} + u_{32}\end{bmatrix}$$

$$\sigma_3(u_1, \ldots, u_K) = \downarrow\begin{bmatrix}u_{21} + u_{31} + u_{41}\\u_{22} + u_{32} + u_{42}\end{bmatrix} \qquad \sigma_4(u_1, \ldots, u_K) = \downarrow\begin{bmatrix}u_{11} + u_{31} + u_{41}\\u_{12} + u_{32} + u_{42}\end{bmatrix}$$

We then consider the state (with two clusters) depicted in Figure 5.1,

$$x[t] = \left(\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}\right)$$

which gives the observations

$$y_1[t] = \begin{bmatrix}\frac{2}{3}\\\frac{1}{3}\end{bmatrix} \qquad y_2[t] = \begin{bmatrix}\frac{2}{3}\\\frac{1}{3}\end{bmatrix} \qquad y_3[t] = \begin{bmatrix}\frac{1}{3}\\\frac{2}{3}\end{bmatrix} \qquad y_4[t] = \begin{bmatrix}\frac{1}{3}\\\frac{2}{3}\end{bmatrix}$$

Using the (fully connected) stochastic coordination mechanism of Section 4.4 as feedback for each cell (i.e., as the $\pi_k$-maps), we get the following controls

$$u_1[t] = \begin{bmatrix}1\\0\end{bmatrix} \qquad u_2[t] = \begin{bmatrix}1\\0\end{bmatrix} \qquad u_3[t] = \begin{bmatrix}0\\1\end{bmatrix} \qquad u_4[t] = \begin{bmatrix}0\\1\end{bmatrix}$$

and thus the system state at the next time-step is

$$x[t+1] = (u_1[t], u_2[t], u_3[t], u_4[t]) = \left(\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}\right) = x[t]$$

which means the state has not changed and never will. There is a system state involving two clusters that is stable. The (fully connected) coordination mechanism has failed in this extremely simple case of sparse connections and thus it seems unlikely that it will work for $K = 100$ or $K = 1000$ cells. A different control law is required. □

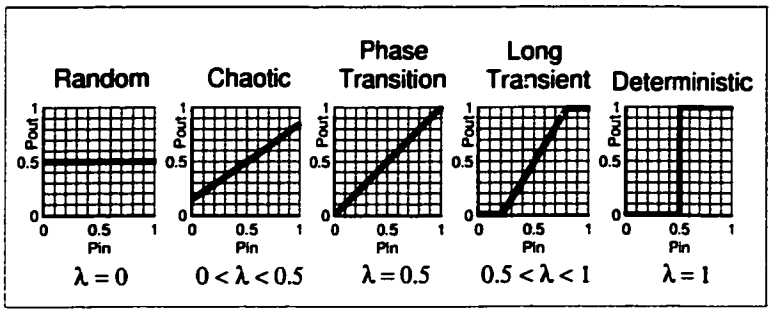



Figure 5.2: The *piecewise-π* rule for different values of $\lambda$ and $m = 2$.

In the above example we will see that the "gain" on our feedback control law is essentially too high, which results in undesirable clusters and boundaries. The possibilities for the $\pi$-map are infinite and thus we discuss parameterized subsets of these possibilities. One possibility, denoted *piecewise-π*, is defined as follows.

**Definition.** PIECEWISE-$\pi$: Let $\boldsymbol{p}_{\text{in}} = [p_{j,\text{in}}] \in {}^m\mathbb{S}$ be the input probability column. The (unnormalized) output probabilities are given by

$$p_{j,\text{out}} = \begin{cases} 1, & \text{if } \frac{1}{m} + \beta(p_{j,\text{in}} - \frac{1}{m}) \geq 1 \\ 0, & \text{if } \frac{1}{m} + \beta(p_{j,\text{in}} - \frac{1}{m}) \leq 0 \\ \frac{1}{m} + \beta(p_{j,\text{in}} - \frac{1}{m}), & \text{otherwise} \end{cases} \qquad \forall j = 1 \ldots m \tag{5.7}$$

where $\beta$ is derived from the tunable parameter $\lambda$ as follows:

$$\beta = \begin{cases} 2\lambda, & \text{if } 0 \leq \lambda \leq \frac{1}{2} \\ \frac{1}{2}(1-\lambda)^{-1}, & \text{if } \frac{1}{2} \leq \lambda < 1 \end{cases} \tag{5.8}$$

The (normalized) output probability column is

$$\boldsymbol{p}_{\text{out}} = \downarrow[p_{j,\text{out}}] \tag{5.9}$$

where $\boldsymbol{p}_{\text{out}} \in {}^m\mathbb{S}$.

Note that in (5.8), the tunable parameter $\lambda$ acts in a similar manner to a temperature parameter. As $\lambda \to 1$ the map approaches the (fully connected) rule of Section 4.4 which is deterministic (except in the event of a tie). When $\lambda = 0$ we have a completely random rule. Figure 5.2 shows what the new rule looks like for different $\lambda$ when $m = 2$.

An equilibrium point, $\boldsymbol{p}^* \in {}^m\mathbb{S}$, in a $\pi$-map is one for which the following is true

$$\boldsymbol{p}^* = \pi(\boldsymbol{p}^*) \tag{5.10}$$

The idea behind the $\pi$-map is to create an instability in the probability map at the uniform distribution equilibrium point, $\boldsymbol{p}^*_{\text{uni}} = \boldsymbol{\omega}$, such that a small perturbation from this point would drive the probability

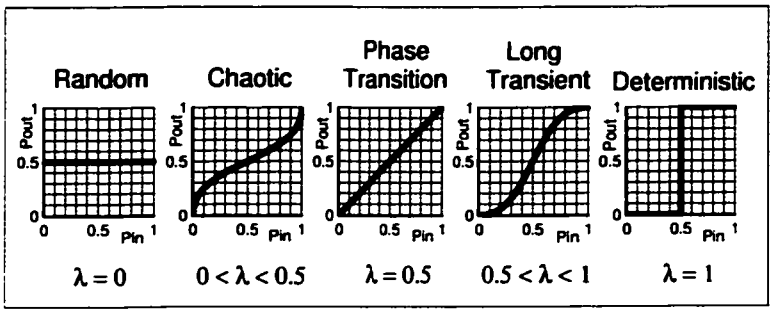



Figure 5.3: The *linear-π* rule for different values of $\lambda$ and $m = 2$.

towards one of the stable equilibria

$$\mathbf{p}_1^* = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \mathbf{p}_2^* = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad \cdots \quad \mathbf{p}_m^* = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{5.11}$$

where $\mathbf{p}_j^* \in {}^m\mathbb{D}\ \forall j = 1 \ldots m$. For the *piecewise-π* map, when $0 \leq \lambda < \frac{1}{2}$, the equilibrium point, $\boldsymbol{p}_{\text{uni}}^*$, is the only stable equilibrium. However when $\frac{1}{2} < \lambda \leq 1$, $\boldsymbol{p}_{\text{uni}}^*$ becomes unstable and the other equilibria[4], $\mathbf{p}_1^*, \ldots, \mathbf{p}_m^*$, become stable. This is similar to the classic pitchfork bifurcation as depicted in Figure 5.4 for $m = 2$. However, with $m$ symbols in the alphabet the pitchfork will have $m$ tines. The point, $\lambda = 0.5$ or $\beta = 1$, will be called the *critical point.* Note that in stochastic algebra, *piecewise-π* is a nonlinear

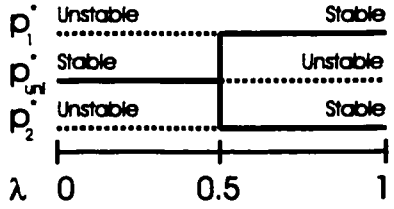



Figure 5.4: Pitchfork stability of the *piecewise-φ* rule for $K = 2$. $\lambda$ is a parameter analogous to a temperature.

function. We may construct another $\pi$-map by linearizing about the equilibrium point, $\boldsymbol{\omega}$. Using the Jacobian approach described in Chapter 2 results in the following map.

**Definition.** LINEAR-$\pi$: Let $\boldsymbol{p}_{\text{in}} \in {}^m\mathbb{S}$ be the input probability column. The output probability column, $\boldsymbol{p}_{\text{out}} \in {}^m\mathbb{S}$, is

$$\boldsymbol{p}_{\text{out}} = \beta \cdot \boldsymbol{p}_{\text{in}} \tag{5.12}$$

where $\beta$ is derived from the tunable parameter $\lambda$ as follows:

$$\beta = \begin{cases} 2\lambda, & \text{if } 0 \leq \lambda \leq \frac{1}{2} \\ \frac{1}{2}(1-\lambda)^{-1}, & \text{if } \frac{1}{2} \leq \lambda < 1 \end{cases} \tag{5.13}$$

[4]Note: There are other equilibria (e.g., $[\frac{1}{2} \quad \frac{1}{2} \quad 0]^T$ when $m = 3$). We would like these, too, to be unstable.

From the linear equation (5.12) it is easy to verify that the desired equilibria are present. Furthermore, we know from the eigenvalues of (5.12) that when $\beta < 1$ the probability distribution will move closer to $\omega$ and when $\beta > 1$ it will mover further away. Thus Figure 5.4 applies also to *linear-$\pi$*. Figure 5.3 depicts the *linear-$\pi$* rule for different $\lambda$ and $m = 2$. Again, as $\lambda \to 1$ we have $\beta \to \infty$ and the simple voting mechanism of Section 4.4 results.

There certainly are other stochastic functions which may work as $\pi$-maps but two will be enough for our purposes here. It is important to stress that we have designed the stability of our system at a local level. The question of global stability and success on the decentralized coordination problem does not follow directly from the local stability of each cell (as we saw for $\lambda = 1$ in the above example). It might be possible to study the global stability of a large system of cells (e.g., $K = 100$) with a *piecewise-$\pi$* or *linear-$\pi$* rule analytically. The problem is that there is an explosion in the number of global states as $K$ is increased. For example, with $K = 100$ and $m = 2$ there are $2^{100} \approxeq 1.3 \times 10^{30}$ possible global states, $\boldsymbol{x} \in {}^{2^{100}}\mathbb{S}$. The approach for the rest of this chapter is to study them through simulation and statistical analysis. This is an important issue for stochastic decentralized systems. If it is computationally intractable to study large systems analytically and prove they will work, then will they still be useful? The hope is that by designing large decentralized systems from the bottom up, the interactions that we design on a small scale will still work on a very large scale. This is typically called *scaling up* and will be investigated here through simulation.

## 5.5 Simulation

We now present simulations of cells running the *piecewise-$\pi$* and *linear-$\pi$* rules. In order to ensure that the connections between cells are not regular, we consider each cell to exist in a Cartesian box (of size 1 by 1). The $K$ cells are randomly positioned in this box and symmetrical connections are formed between two cells if they are closer than a threshold Euclidean distance, $d$, from one another. Figures 5.5 through 5.7 show example connections for $K = 100$ cells (with $d = 0.2$) and $K = 400$ cells (with $d = 0.1$). Figure 5.8 shows example time series for different values of $\lambda$ for the *piecewise-$\pi$* rule. When $\lambda < 0.5$, chaotic global behaviour arises, with $0.5 < \lambda < 1$ fairly successful behaviour results but with $\lambda = 1$ clusters form. The formation of clusters means that the global system has stable equilibria which we did not predict from the local rule. However, as $\lambda$ is decreased towards 0.5, these equilibria are no longer stable and the system continues to coordinate.

It would seem that there is a good correlation between the stability on the local level and the behaviour type of the global system. As $\lambda$ moves from below 0.5 to above, it appears there is a dramatic phase transition in the behaviour of the system. In the neighbourhood of 0.5 there is long transient behaviour. It turns out that the best value for $\lambda$ (for the simulation parameters considered here) from the point of view of decentralized coordination, is approximately $\lambda = 0.6$.

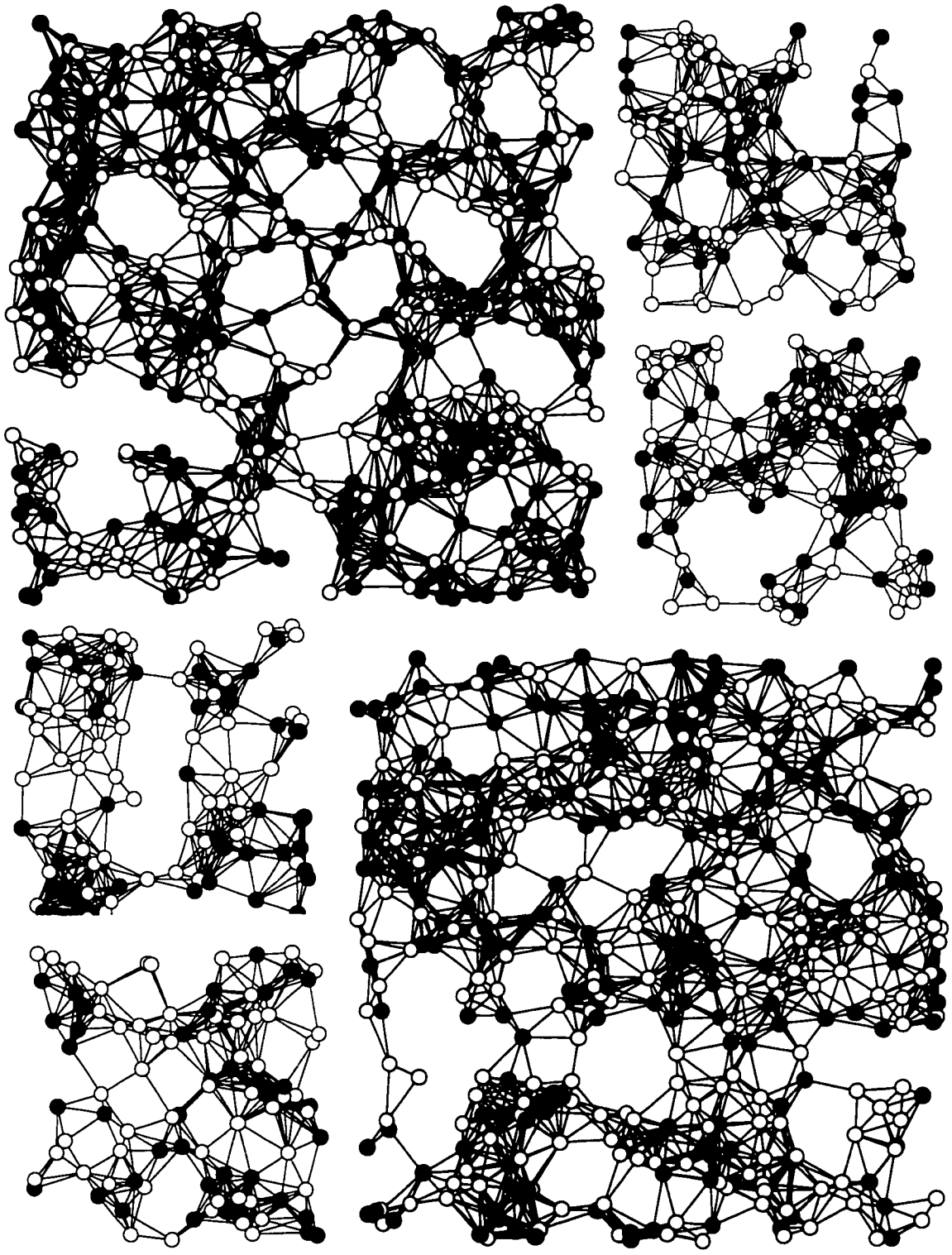


Figure 5.5: Six examples of random initial conditions for alphabet size, $m = 2$. The two colours represent the two symbols of the alphabet. The larger examples have $K = 400$ and $d = 0.1$ while the smaller ones have $K = 100$ and $d = 0.2$.

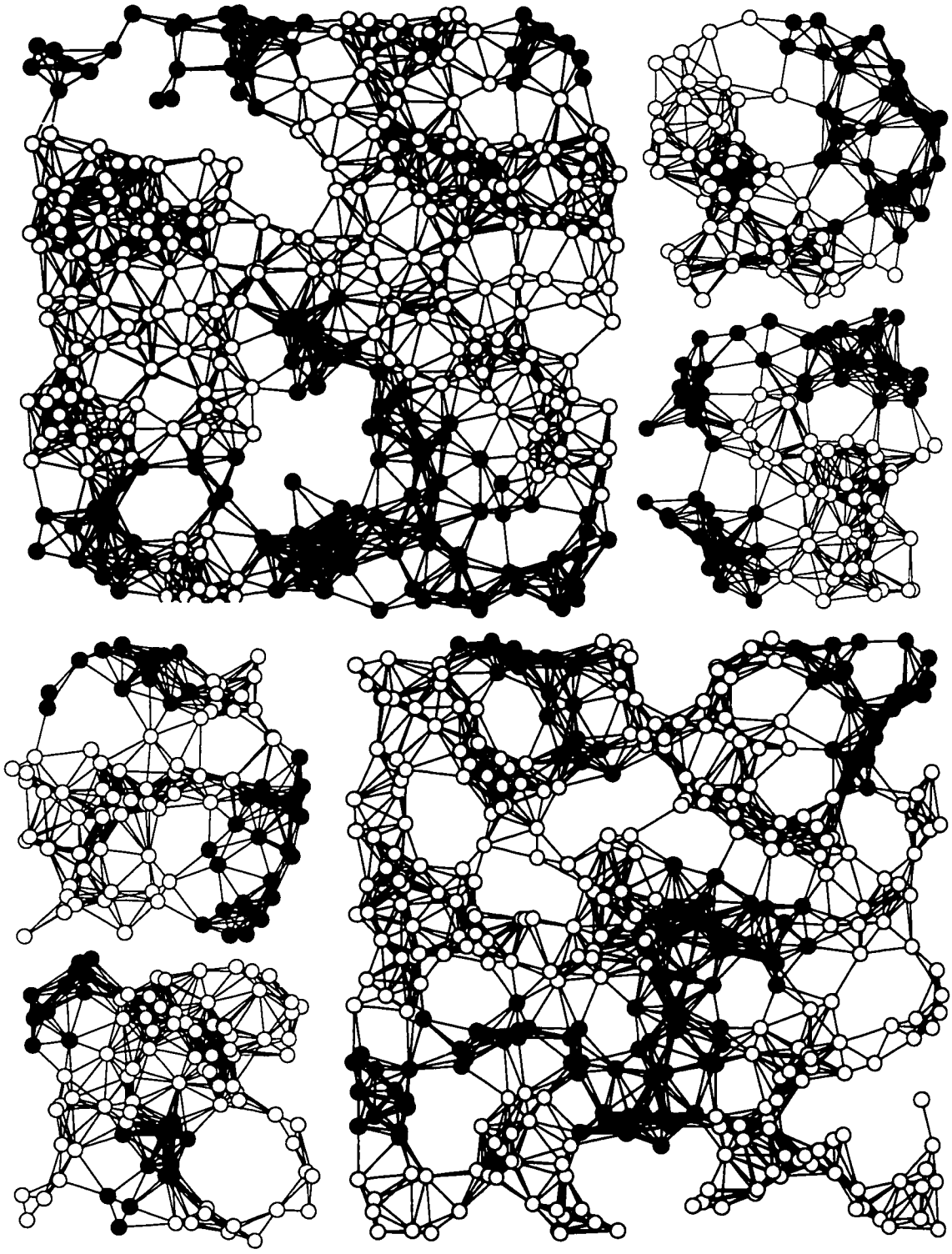


Figure 5.6: Six examples of undesirable clusters forming for alphabet size, $m = 2$. The two colours represent the two symbols of the alphabet. The larger examples have $K = 400$ and $d = 0.1$ while the smaller ones have $K = 100$ and $d = 0.2$.

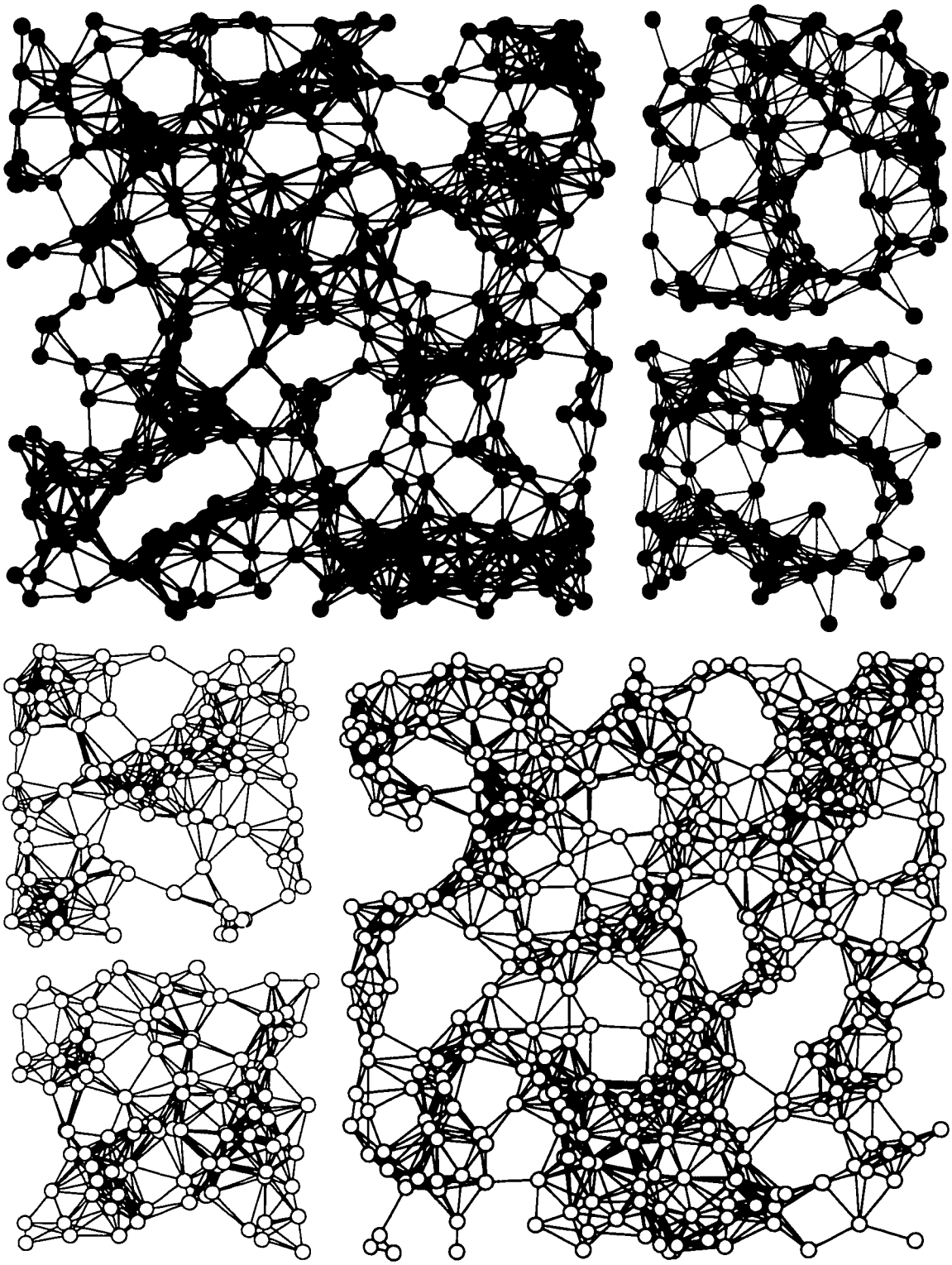


Figure 5.7: Six examples of consensus for alphabet size, $m = 2$. The two colours represent the two symbols of the alphabet. The larger examples have $K = 400$ and $d = 0.1$ while the smaller ones have $K = 100$ and $d = 0.2$.

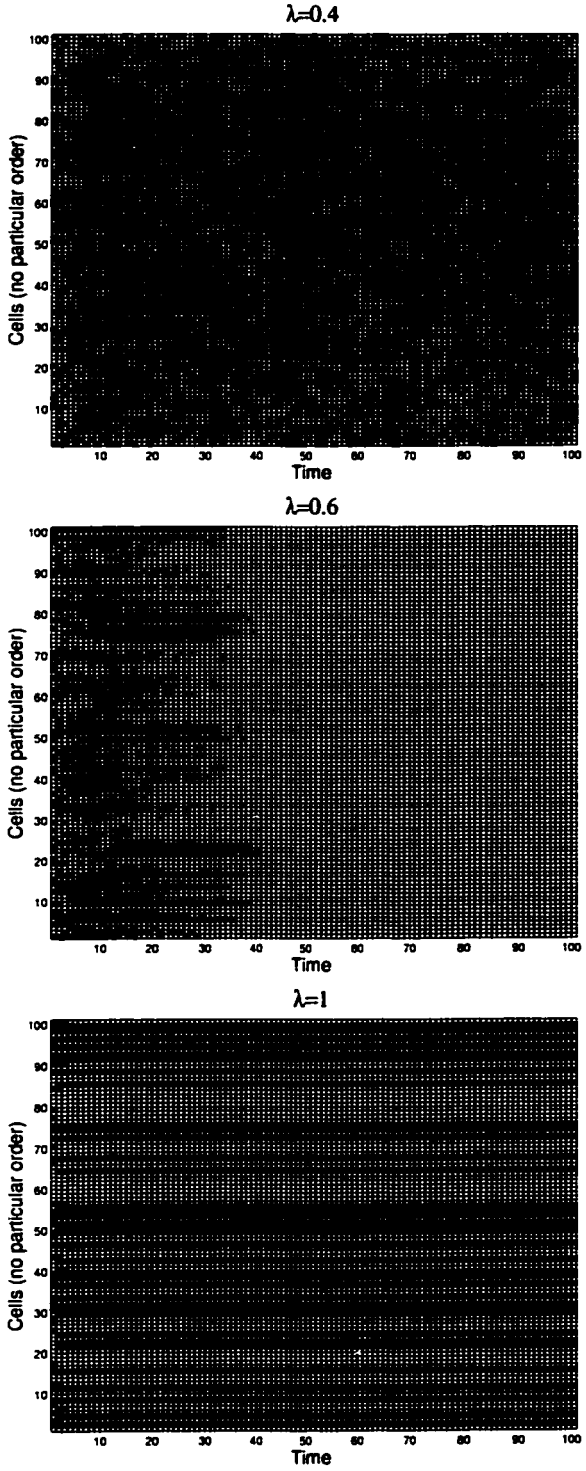



Figure 5.8: Example time series for different values of $\lambda$ and $K = 100$, $m = 2$, $d = 0.2$. (top) chaotic behaviour, (middle) successful coordination, (bottom) clusters. The two colours represent the two symbols of the alphabet. The *piecewise-$\pi$* rule was used but plots are qualitatively the same for *linear-$\pi$*.

## 5.6 Statistical Analysis

In an attempt to quantify the qualitative observations of the previous section, a number of statistical measures were employed in the analysis of the SCA time series. These were used also by Langton [1990]. The first measure is taken from Shannon [1948] and will be referred to as *entropy* ($H$). It is defined as follows.

**Definition.** ENTROPY[5]: Given a sequence of $n$ symbols

$$\mathbf{s} = [\, s_1 \;\; s_2 \;\; \cdots \;\; s_n]^T \tag{5.14}$$

from an alphabet of size $m$, the *entropy* of the sequence may be computed as follows. First compute the frequency, $n_j$, of each of the $m$ symbols $\forall j = 1 \ldots m$ which is simply the number of occurrences of symbol $j$ in the sequence, $\mathbf{s}$. From the frequencies, compute the probability, $p_j$, of each of the $m$ symbols $\forall j = 1 \ldots m$ as

$$p_j = \frac{n_j}{n_\Sigma} \tag{5.15}$$

where $n_\Sigma = \sum_{l=1}^{m} n_l$. Finally, the entropy of sequence, $H(\mathbf{s})$, is defined as

$$H(\mathbf{s}) = \frac{-\sum_{j=1}^{m} p_j \ln(p_j)}{\ln(m)} \tag{5.16}$$

where the $\ln(m)$ denominator is a normalization constant to make $H(\mathbf{s}) \in [0, 1]$.

This entropy function produces a value of 0 when all the symbols in $\mathbf{s}$ are identical and a value of 1 when all $m$ symbols are equally common. The second measure is based on the first and will be referred to as *mutual information* ($I$). It is defined as

**Definition.** MUTUAL INFORMATION: Given two sequences of $n$ symbols each

$$\mathbf{s}_1 = [\, s_{1,1} \;\; s_{1,2} \;\; \cdots \;\; s_{1,n}]^T \tag{5.17}$$

$$\mathbf{s}_2 = [\, s_{2,1} \;\; s_{2,2} \;\; \cdots \;\; s_{2,n}]^T \tag{5.18}$$

from an alphabet of size $m$, the *mutual information* of the sequence, $I(\mathbf{s}_1, \mathbf{s}_2)$, may be defined as

$$I(\mathbf{s}_1, \mathbf{s}_2) = H(\mathbf{s}_1) + H(\mathbf{s}_2) - H(\mathbf{s}_1, \mathbf{s}_2) \tag{5.19}$$

where $H(\mathbf{s}_1, \mathbf{s}_2)$ is the entropy of the two sequences considered as a joint process (i.e., with an alphabet of size $m \times m$).

These two measures may be computed on any sequence of symbols. They were tested on *spatial* sequences (e.g., time series *columns* from Figure 5.8) and *temporal* sequences (e.g., time series *rows* from Figure 5.8). The most interesting measures were *average spatial entropy* (average of entropies computed from all columns in a time series) and *average temporal mutual information* (average of all $I$s computed from all rows in a time series. $I$ was computed between a row and itself shifted by one time-step).

[5] See also the section on information theory in Chapter 2.

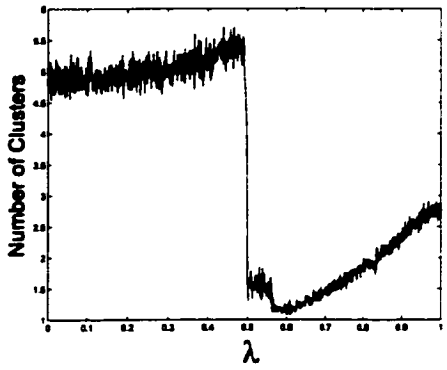



Figure 5.9: Average number of clusters at final time-step for 1000 values of $\lambda$. Plot shows average of 100 simulations at each value of $\lambda$. Number of clusters was computed by considering the SCA as a Markov chain with connections deleted between cells displaying different symbols. The number of clusters is then the number of eigenvalues equal to 1 from the Markov transition matrix. *Piecewise-$\pi$* rule was used.

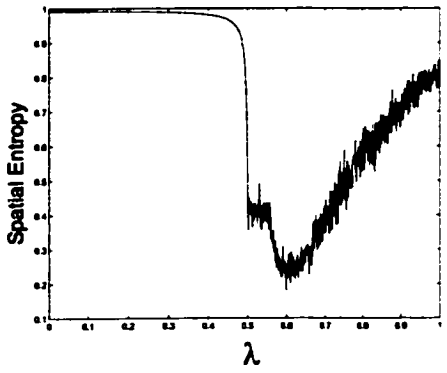



Figure 5.10: Average spatial entropy for 1000 values of $\lambda$. Plot shows average of 100 simulations at each value of $\lambda$. *Piecewise-$\pi$* rule was used.

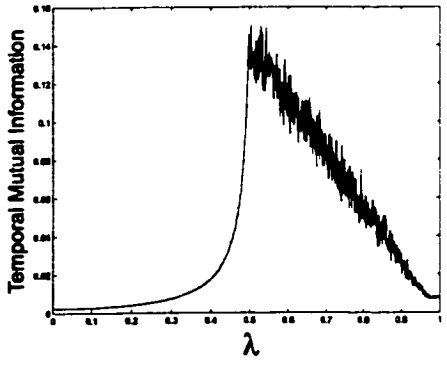



Figure 5.11: Average temporal mutual information for 1000 values of $\lambda$. Plot shows average of 100 simulations at each value of $\lambda$. *Piecewise-$\pi$* rule was used.

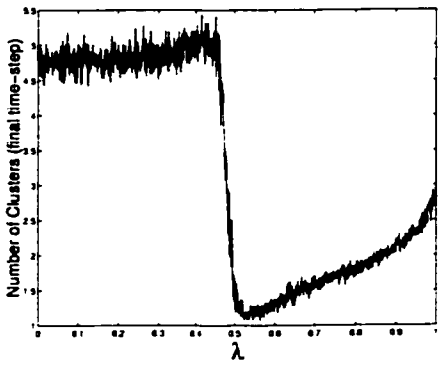



Figure 5.12: Average number of clusters at final time-step for 1000 values of $\lambda$. Plot shows average of 100 simulations at each value of $\lambda$. Number of clusters was computed by considering the SCA as a Markov chain with connections deleted between cells displaying different symbols. The number of clusters is then the number of eigenvalues equal to 1 from the Markov transition matrix. *Linear-$\pi$* rule was used.

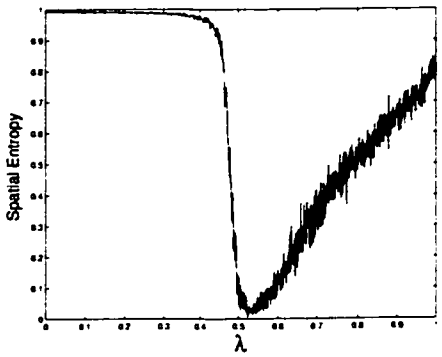



Figure 5.13: Average spatial entropy for 1000 values of $\lambda$. Plot shows average of 100 simulations at each value of $\lambda$. *Linear-$\pi$* rule was used.

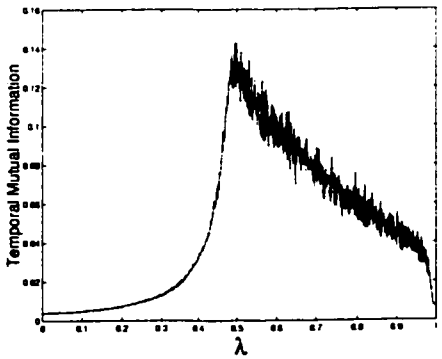



Figure 5.14: Average temporal mutual information for 1000 values of $\lambda$. Plot shows average of 100 simulations at each value of $\lambda$. *Linear-$\pi$* rule was used.

Figures 5.9, 5.10, 5.11 show various measures for *piecewise-π* and 1000 values of $\lambda$. At each value of $\lambda$, 100 simulations were done on different random connections between cells and initial conditions. Thus, all displayed measures are actually averaged over 100 simulations. Each simulation was run for 300 time-steps with $K = 100$, $m = 2$, and $d = 0.2$. Figures 5.12, 5.13, 5.14 show the same measures for *linear-π*.

Figures 5.9 and 5.12 show the average number of clusters at the final time-step for different values of $\lambda$. The phase transition is quite obvious at $\lambda = 0.5$. The optimal value (in terms of the fewest clusters formed on average) for $\lambda$ is near 0.6 for *piecewise-π* and 0.53 for *linear-π*. Figures 5.10 and 5.13 show average spatial entropy for different values of $\lambda$. This measure has a good correlation with average number of clusters. Again, there is a minimum occurring which corresponds to the best performance at multiagent coordination.

Figures 5.11 and 5.14 display average temporal mutual information for different values of $\lambda$. This is a very interesting plot. Temporal mutual information seems to capture the length of the global transient behaviour of the system. As discussed by Langton [1990], the random pattern in the chaotic region is not considered transient but rather the steady-state behaviour. The peak in temporal mutual information occurs at $\lambda = 0.5$ , the phase transition, and drops away on either side (for both rules). Langton [1990] has a similar plot.

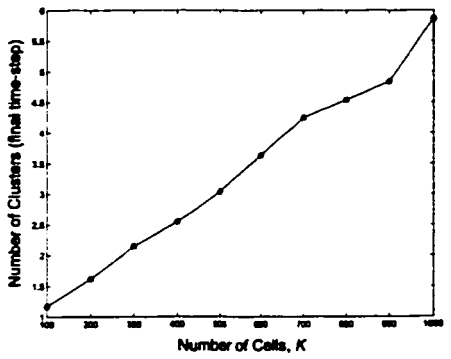



Figure 5.15: Average number of clusters (at the final time-step) as the number of cells, $K$, is varied from 100 to 1000. The parameters were: 300 time-steps, $m = 2$, $\lambda = 0.6$, $d = \frac{2}{\sqrt{K}}$. The *piecewise-π* rule was used. Plot shows average from 100 simulations at each value of $K$.

Figure 5.15 shows how the average number of clusters at the final time-step varies as the problem is scaled up from $K = 100$ cells to $K = 1000$ cells. The plot shows an average of 100 simulations, each run for 300 time-steps with $m = 2$, $\lambda = 0.6$, and $d = \frac{2}{\sqrt{K}}$. The parameter $d$ was made to depend on the number of cells in order to keep the average density of connections the same. This was required as the Cartesian box in which the cells live was always of size 1 by 1. As more cells are added they are closer together and thus to keep the density of connections between cells constant (on average), the factor of $\frac{10}{\sqrt{K}}$ was needed. The resulting relationship between number of clusters and number of cells is quite linear, about 0.47 clusters for every 100 cells added. Barfoot and D'Eleuterio [1999] show a qualitatively similar scaling plot for a heap formation problem.

Figure 5.16 shows how the average number of clusters, again at the final time-step, varies as the problem is scaled up from an alphabet size, $m$, of 2 (a single bit) to 256 (8 bits or 1 byte). The plot

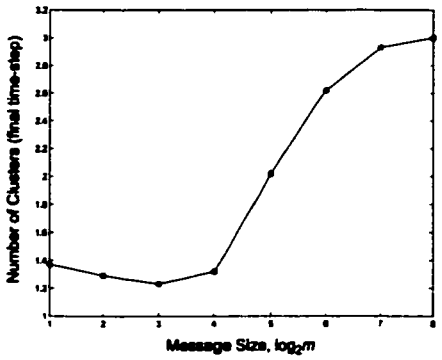



Figure 5.16: Average number of clusters (at the final time-step) as the alphabet size, $m$, is varied from 2 (1 bit) to 256 (8 bits). The parameters were: 300 time-steps, $K = 100$, $\lambda = 0.6$, $d = 0.2$. The *piecewise-$\pi$* rule was used. Plot shows average from 100 simulations at each value of $m$.

shows an average of 100 simulations, each run for 300 time-steps with $m = 2$, $\lambda = 0.6$, and $d = 0.2$.

## 5.7 Shifting the Instability

In the theory and simulations presented above, it was assumed that $K$ cells are trying to agree on a piece of information, $\mathbf{u} \in {}^m\mathbb{D}$ which is one of $m$ symbols from the allowable alphabet. The $\pi$-maps presented created an instability at the uniform distribution, $\boldsymbol{\omega}$. At the single trajectory level, a specific symbol will (if we wait long enough) be chosen with probability 1. However, if we run enough simulations and average the results, each symbol will be chosen with probability $\frac{1}{m}$. Thus, at the ensemble level we are still at the equilibrium, $\boldsymbol{\omega}$.

We may think of the unstable equilibrium as the distribution from which we would like the $K$ cells to select a single unique symbol. So far we have been assuming this is the uniform distribution, $\boldsymbol{\omega}$. However, we might like the group to select a symbol from a different distribution. This is necessary, for example, in the linear stochastic system of Chapter 3. To do this we must shift the unstable equilibrium in the $\pi$-map to the new desired distribution, $\boldsymbol{u}^* \in {}^m\mathbb{S}$. This is done by replacing (5.2) with

$$\boldsymbol{p}_{\text{out}}[t] = \pi\left(\boldsymbol{p}_{\text{in}}[t] \ominus \boldsymbol{u}^*\right) \oplus \boldsymbol{u}^* \tag{5.20}$$

such that we may continue to use the same $\pi$-maps, regardless of the desired distribution, $\boldsymbol{u}^*$. In the case of the *linear-$\pi$* map we have the following

$$\begin{aligned} \boldsymbol{p}_{\text{out}}[t] &= \pi\left(\boldsymbol{p}_{\text{in}}[t] \ominus \boldsymbol{u}^*\right) \oplus \boldsymbol{u}^* \\ &= \beta\cdot\left(\boldsymbol{p}_{\text{in}}[t] \ominus \boldsymbol{u}^*\right) \oplus \boldsymbol{u}^* \\ &= \beta\cdot\boldsymbol{p}_{\text{in}}[t] \ominus \beta\cdot\boldsymbol{u}^* \oplus \boldsymbol{u}^* \\ &= \beta\cdot\boldsymbol{p}_{\text{in}}[t] \oplus (1-\beta)\cdot\boldsymbol{u}^* \end{aligned} \tag{5.21}$$

which is interestingly in the form of the linear system in (3.115). Here $\boldsymbol{u}^*$ is the control input to the equation. In this form we may immediately use our coordination mechanism to effect any control distribution, $\boldsymbol{u}^* \in {}^m\mathbb{S}$, over an arbitrarily connected network of $K$ controllers.

To do this each controller takes as input the desired control distribution, $\boldsymbol{u}^*$, and runs the stochastic coordination mechanism of this chapter. After a sufficiently long amount of time the system will agree on a deterministic, $\mathbf{u}$, which has been selected from $\boldsymbol{u}^*$. As detailed in Section 3.6.2, a deterministic control sequence can always be carried out exactly by independent controllers.

This is a rather fundamental point so it is repeated for clarity. Independently acting controllers cannot exactly implement control from a desired (arbitrary) stochastic control distribution as it may require them to act in a statistically dependent manner. However, by communicating (using the coordination mechanism of this chapter) through sparse connections they can agree on a deterministic control from the desired distribution (without use of a centralized facility) which can then be exactly implemented by them in a decentralized manner. This occurs at the single trajectory level. At the ensemble level, the original stochastic control distribution is exactly implemented.

## 5.8 Discussion

The strong correlation between the local stability of the *piecewise-$\pi$* and *linear-$\pi$* rules and the type of global behaviour is quite interesting. It appears that $\lambda \gg 0.5$ corresponds to fixed point behaviour (Wolfram's class I), $\lambda \ll 0.5$ corresponds to chaotic behaviour (Wolfram's class III), and $\lambda$ near 0.5 corresponds to long transient behaviour (Wolfram's class IV). Local correlation has to do with the way in which the incoming probability distribution is computed in (5.1). This step delivers information averaged from all connected cells. This averaging serves to smooth out differences between connected cells. However, if this smoothing occurs too quickly (i.e., $\lambda = 1$) the system does not have time to smooth globally resulting in the formation of clusters. This has been called *critical slowing down* [Haken, 1983b] in other systems. As we approach the critical point ($\lambda = 0.5$ or $\beta = 1$) from above, the strength of the instability decreases which slows down the decision-making process. The third and vital ingredient in the recipe for self-organization is the fluctuations that occur at the single trajectory level. These fluctuations allow the system to begin the process of moving away from the unstable equilibrium, $\boldsymbol{\omega}$, in the $\pi$-maps. The nature of the $\pi$-maps is such that these fluctuations are largest when the system is near $\boldsymbol{\omega}$ and becoming smaller and smaller as a cell becomes more coordinated with its neighours. It is a balance of these three effects which seems to be the most effective at decentralized coordination. To summarize, self-organization in this model requires the following three mechanisms:

> INSTABILITY in the $\pi$-map which forces each cell to move away from $\boldsymbol{\omega}$ (behaving randomly) and towards one of a number of deterministic decisions.
>
> AVERAGING in the $\sigma$-map which serves to bias each cell to conform to the average behaviour of its immediate (connected) neighbours.
>
> FLUCTUATIONS at the single trajectory level to cause each cell to move away from the unstable equilibrium, $\boldsymbol{\omega}$. These fluctuations become smaller as the cell moves further away.

To properly balance these three effects the parameter, $\lambda$, was tuned. The optimal operating value of $\lambda$ is not right at the phase transition but a little bit towards the deterministic end of the $\lambda$ spectrum (approximately $\lambda = 0.6$ for *piecewise-$\pi$* and 0.53 for *linear-$\pi$*).

Note that we did not find any oscillatory behaviour (Wolfram's class II) which is because the connections between the cells are symmetrical. However, if the *piecewise-π* rule in Figure 5.2 is reflected (left-right) then the system "blinks" and global coordination corresponds to all cells blinking in phase with one another. The same may be said for the *linear-π* rule.

What is happening in the SCA model is that the boundaries between clusters are made unstable. This forces them to move randomly until they contact one another and annihilate, leaving a single cluster. This annihilation of boundaries is qualitatively the same method found to work in deterministic CAs by Mitchell et al. [1993] and Das et al. [1995]. In those studies the boundaries were made to move in very specific ways by exploiting the nature of the connections between cells. They found that the boundaries could be made to travel long distances. This allowed coordination to occur more quickly than the method presented here. However, their mechanism was not immediately portable to different connective architectures. By not exploiting the underlying connections between cells, the best we can do is to make the boundaries move randomly and wait for them to contact one another and annihilate. The benefit is that this method is independent of the connective architecture.

The results presented here used $K = 100$ cells and required on average 150 time-steps to get to a single cluster with $d = 0.2$, $m = 2$ and $\lambda = 0.6$. Clearly, the time required to form a single cluster will increase with the number of cells in the system. This is reflected in Figure 5.15 which shows how the number of clusters after 300 time-steps varies as the number of cells is increased. The larger systems are not able to finish coordinating in the allowed time, thus resulting in more clusters.

Figure 5.16 shows how the number of clusters at the final time-step varies with the alphabet size, $m$. This is more difficult to explain as the curve first goes down a little and then up as $m$ is increased from 2 to 256 in factors of 2. This would have been difficult to predict analytically. In some ways the $m = 2$ case is very difficult as there can be two equally large clusters whose boundary fluctuates but is never annihilated leaving a single cluster. There is effectively a stalemate, no further progress is being made. Having more clusters that are smaller in size can make the fluctuations relatively bigger, enabling boundaries to be annihilated more quickly. This explains the initial decline in Figure 5.16. The eventual rise in the plot (increasing from $m = 16$) is similar to that in Figure 5.15. Although the system is still making progress, it is not able to complete coordination in 300 time-steps. As the alphabet size becomes larger than the number of cells (here 100), the plot levels off. This may be explained by the fact that $K$ cells cannot represent more than $K$ different symbols in the random initial condition, regardless of how large we make the alphabet size, $m$. Note that if the system becomes too inefficient at very large $m$ (i.e., too time consuming), it is possible to use more than one coordination mechanism and combine the results. For example, two of the $m = 8$ mechanisms could be combined to produce messages of size 64. Depending on the parameters this may or may not improve coordination efficiency. The last point which should be mentioned is that the same value of $\lambda = 0.6$ was used at all values of $m$. There could, however, be different optimal values for this parameter for each $m$.

The *piecewise-π* and *linear-π* rules are not the only maps that can be used to achieve decentralized coordination in SCA. Replacing it with other monotonically increasing functions (i.e., in Figure 5.2) with the same equilibria will likely work. This was tried for $m = 2$ with the relation

$$p_{j,\text{out}} = 3(p_{j,\text{in}})^2 - 2(p_{j,\text{in}})^3 \quad \forall j = 1 \ldots m \tag{5.22}$$

which has equilibria at $[\ 1\ \ 0\ ]^T, [\ 0\ \ 1\ ]^T, [\ \frac{1}{2}\ \ \frac{1}{2}\ ]^T$. It is fairly successful but the *piecewise-$\pi$* and *linear-$\pi$* rules are easier to parameterize. The above rule is similar in form to equations studied by Haken et al. [1973, 1984], particularly the cubic term.

The equilibria are the most important features to consider in the design of $\pi$-maps. Creating an instability at the uniform distribution, $\omega$, is crucial. However, there are other features which can be incorporated. For example, the *linear-$\pi$* rule is appealing as it provides a smooth route to a deterministic decision. Essentially, less and less noise is added as the incoming probability gets closer to one of the deterministic equilibria. This is a form of proportional feedback (which is linear in stochastic algebra). The *piecewise-$\pi$* is nonlinear but this may have an additional useful feature in some situations. In the saturated regions of the curve (i.e., the horizontal flat parts) normal voting results. This feature greatly increases the strength of the stability of the deterministic equilibra. Thus any perturbation to the system is less likely to drive the system away from one of these regions. For example, if a group of 100 cells were already coordinated and a few more cells were introduced into the system at a later time, this would increase the likelihood of the new cells to conform. Similarly, if one cell were to begin malfunctioning it would be less likely to cause the other cells to uncoordinate. The size of the saturated region may be tuned as circumstances require.

Finding the optimal value for $\lambda$ for a particular set of parameters may not actually be necessary. As a future direction of research, a "cooling schedule" could be developed. We could start with $\lambda$ near the phase transition (e.g., $\lambda$ just larger than 0.5) and then slowly "cool" the system by bringing $\lambda$ gradually towards 1. The system would certainly pass through the optimal value for $\lambda$. This form of cooling schedule has been used, for example, in simulated annealing, a global optimization method. This would require each cell having some form of internal clock in order to time the cooling. Another possibility is to allow each cell to program its own $\lambda$ using feedback. $\lambda$ would get larger in periods of inactivity and smaller in periods of high activity. Removing the need for a centralized designer to program $\lambda$ is one more step towards fully autonomous decentralized decision-making. Another future direction of work is to consider the addition of noise to the communication between cells. This could be done, for example, using the state-projection matrices from Chapter 3. It is likely that a small amount of communication noise will not cause the system to catastrophically stop working as it has been built on fluctuation and noise to begin with.

The model considered here does not require knowledge of the underlying structure of the connections between cells. This was a design requirement as it was originally motivated by a network of communicating mobile robots whose connections might be changing over time and thus difficult to exploit. It is thus natural to question whether the model still works as the connections are varied over time. To this end, a small amount of Gaussian noise was added to the positions of the cells in the Cartesian box at each time-step. As the cells moved, the connections between them changed (since they are limited by the range, $d$). The SCA model was still able to form single clusters. This was possible even when $\lambda = 1$ which makes sense since there is still some noise being added. However, the nature of the noise is at the connection level rather than the signal level. This is fairly obvious. Over a long period of time it is as though the system were fully connected. However, the assumption of completely random movement is probably not a good one for a system of mobile robots. Consider, for example, the tasks described in the previous chapter. The coordination mechanism described here has tried to make as few assumptions

as possible about the nature of connections between cells.

Finally, it should be noted that the fully connected voting scheme from Chapter 4 is identical to either the *piecewise-$\pi$* or *linear-$\pi$* rules from this chapter with $\lambda = 1$. Because the robots are fully (not sparsely) connected, a long transient is not required to achieve global coordination and thus $\lambda = 1$ gets the job done as quickly as possible.

## 5.9 Summary

A mechanism for decentralized coordination has been presented based on stochastic cellular automata. This is an example of self-organizing behaviour in that global coordination occurs in the face of more than one alternative. It was shown that by using stochastic rules, sparsely communicating agents could come to a global consensus. A common piece of information may be generated to which each cell has access using a stochastic approach. A parameter in the coordination mechanism was tuned and it was found that coordination occurred best when the system was near a phase transition between chaotic and ordered behaviour (the optimum was a little bit towards the ordered side). It is hoped that this model will shed light on self-organization as a general concept while at the same time providing a simple algorithm to be used in practice.

Under what conditions [could, would, must, might] communication arise as a feature of interaction between individuals in groups? Can we build a gradualist bridge from simple amoeba-like automata to highly purposive intentional systems, with identifiable goals, beliefs, etc.? Under what conditions does the fate of groups as opposed to individuals play a decisive role in evolution? What is an individual?

—Daniel C Dennett

ARTIFICIAL LIFE AS PHILOSOPHY, 1994

# Chapter 6

# SYNTHESIS

This study of stochastic decentralized systems has wound its way downward from the whole to the parts and back. This chapter begins with a summary which serves to deepen the underlying connections between the ideas presented in the previous technical chapters. It furthermore tries to synthesize this work with the massive body of existing research on self-organization in general.

It was shown in the chapter on control that the assumption of a central locus of control is made unnecessary through the introduction of communication between decentralized controllers. Some might argue that this complicates matters more than it simplifies them. Although this may be true, the view here is that centralized control is a drastically simplified case of the more general problem of decentralized control. With this view in mind, the study's focus turned to building upward from the parts to the whole. By crafting specific types of communication between groups of decentralized controllers/robots/cells it was shown how to produce global behaviours that in the eye of the observer could just as easily appear to be centrally controlled. Sometimes if an observer looks hard enough, these systems can appear to be purposive, to have the semblance of an "idea"[1]. But if intelligence is in the eye of the observer, as claimed by new AI, then it is impossible to make any objective claim about its existence. This will be discussed further in the closing section. Tempting as these subjective notions may be, they have been avoided here. This study of decentralized systems has sought to objectively understand the fundamental interactions between autonomous agents which enable global coordination in a completely decentralized manner.

[1] Based on comments many observers of the robotics system have made.

Chapter 5 on self-organization sought to look for such fundamental features in a simple model of sparsely communicating cells which were trying to generate a common piece of information. The problem of coordination is an example of *self-organization* in that a common choice must be made in the face of more than one alternative. The study of a large number of sparsely communicating agents almost necessitates using some form of statistical analysis if any general properties are to be found. Even when the underlying dynamics are deterministic, statistical tools are needed to quantify the behaviour in an average sense. This may be seen throughout the study of artificial life and complex phenomena. For example, the deterministic cellular automata described by Mitchell et al. [1993] and Das et al. [1995] do not always form a global consensus; they do so with a high probability. The NK models of Kauffman and Johnsen [1991] examine the local coupling between genes in a DNA sequence using simple automata. The resulting ruggedness of the fitness landscapes they produce are quantified statistically. The sandpile model (and many others) of Bak [1996] shows the existence of a self-organized critical state in a locally coupled system of simple automata. Here the number and size of sand avalanches obey a beautiful statistical law. Analyses of real earthquake data have shown these phenomena to obey similar laws but it is still impossible to predict when and where avalanches and earthquakes will occur. Bak [1996] notes

> Self-organized critical systems evolve to the complex critical state without interference from any outside agent. The process of self-organization takes place over a very long transient period. Complex behaviour, whether in geophysics or biology, is always created by a long process.

In the stochastic cellular automata models described here this notion of a long transient was found to be very important. This quantity was captured well by the statistical measure, temporal mutual information. Das et al. [1995], Mitchell et al. [1993, 1996] found their solutions to display long transients. Langton [1990, 1991] found long transient behaviour to occur in the vicinity of a phase transition (between order and chaos) in the global behaviour of the system. This was also found here. He further suggested that long transient behaviour might be linked to natural computation in decentralized systems which was reexamined by Mitchell et al. [1993]. Hanson and Crutchfield [1995] attempted to formalize the connection between long transient behaviour and natural emergence of computation. Some cellular automata have even been shown to display all the necessary features of a universal Turing machine. Thus computation is at least theoretically possible in a system of sparsely communicating simple automata.

The 'computation' here was simply to form a consensus in a decentralized manner. It was found that a long transient *is* necessary in order to avoid the formation of small clusters in sparsely connected SCA. To achieve long transient behaviour, the $\lambda$ parameter needs to be lowered from $\lambda = 1$ (deterministic) towards the critical stability point, $\lambda = 0.5$, where a phase transition in the global behaviour of the system occurs. The best value for $\lambda$, in terms of global coordination, was found to be approximately 0.6. Lowering $\lambda$ is analogous to raising the temperature of the system. The transient is made longer by decreasing the strength of the instability (making the eigenvalues smaller in magnitude) in the $\pi$-map. By adding noise, which is analogous to heat (random energy), in this particular way, the decision making process is slowed down. This *critical slowing down* [Haken, 1983b] is necessary to allow the system enough time to globally coordinate. The local decisions made by the cells are thus drawn out long enough to allow information to be transmitted globally. This can also be thought of as adding noise (in just the right way) to avoid getting caught in local minima (clusters). Thus it is not the length of the transient itself that is important but rather the time it buys for communication of data. Spatial

entropy captures the degree of coordination very well.

It was found that balancing three important effects enabled global coordination in this model: an *instability* in the probability map to force each cell to make a decision, *averaging* of the incoming information from neighbouring cells, and *fluctuations* to start the ball rolling on the instability. These notions have been borrowed from nonlinear physics (not for the first time) where instabilities and fluctuations play important roles in many self-organizing systems [Nicolis and Baras, 1984], [Prigogine and Stengers, 1984], [Haken and Mikhailov, 1994, Haken, 1984] . The resulting SCA mechanism requires no adaptation to be used as a coordination mechanism for an arbitrarily connected network of decision-makers or controllers. The more connections between agents, the faster it coordinates. Put another way, longer transient behaviour is required for more sparsely connected cells. This mechanism (with $\lambda = 1$) was employed by the robots in Chapter 4 in order to make decentralized decisions and thus coordinate their behaviour. Note, this mechanism does not guarantee that a global decision will be reached, only that it will be reached with a high probability. The longer you wait, the higher that probability.

Being able to generate a common piece of information can actually be cast as a control problem (as was seen in Section 5.7). But how can we get a system of sparsely communicating agents to behave in an arbitrary way? To this end, Chapter 3 examined in some depth the effect decentralization has on systems in general. It was noted that there are actually two types of decentralization, that of control and that of observation. The very process of decentralization thus imposes two constraints on the system, one for each of control and observation. If we think of a centralized controller as a number of communicating decentralized controllers, then the actual process of 'decentralization' constrains the system by severing the communication links. This can have the same effect as removing both sensors and actuators from the system. Naturally this limits the possible behaviours of the decentralized system as a whole. It in fact naturally defines four classes of controllers: (requiring)/(not requiring) communication for (sensory sharing)/(action coordination). From a control point of view, the more communication bandwidth available, the better. There are two ways of summarizing these ideas. First, communication may be used to counter the effects of decentralization. Second, the illusion of centralized control may be created by using communication between decentralized controllers. When the communication bandwidth is considered to be infinite, these are the same.

These ideas were discussed in the framework of controlled Markov chains (DecPOMDPs). Markov systems were chosen for this study of control as they are inherently stochastic and have properties analogous to physical systems (e.g., they may be described in terms of entropy). It was hoped that connections could be made between existing control theory and Markov systems. Motivated by this possibility, the *stochastic algebra* of Chapter 2 was developed. This algebra allows stochastic dynamic equations (difference equations) which govern the temporal flow of probability distributions, to be rewritten in the familiar matrix algebra. In stochastic algebra, the zero vector is the uniform probability distribution. All interactions between controllers may be described in terms of statistical dependence. Linearity is akin to statistical independence. Causing decentralized controllers to behave in a coordinated manner then requires somehow making them behave in a statistically dependent (nonlinear) manner. It was shown that the reactive feedback controllers of Bellman [1957] for the centralized Markov system (MDP) fall into four classes as a result of decentralization. With the addition of communication, decentralized reactive controllers can achieve any behaviour a centralized reactive controller can. Similar arguments

were made for dynamic controllers. The more interesting cases, naturally, are those requiring communication for both sensory sharing and action coordination. It should be stressed that from the point of view of control, implementing an arbitrary 'centralized' controller on a group of decentralized controllers requires two *task independent* mechanisms: communication for *sensory sharing* and communication for *action coordination*. The first involves propagating existing information throughout the system while the second goes one step further to involve creating a *new* piece of information, common to all controllers in the system which may then be used for coordination. The generic modules presented in Chapter 4, for example, fill these roles. From this point of view, once the communication facilities are established, any global behaviour may be achieved (given enough bandwidth).

The control of a Markov chain is in fact a nonlinear problem in the new stochastic algebra. However, assuming that the transition matrices are *regular* under a uniformly random control, the system will gradually head to a stable equilibrium distribution. It is then possible, using the stochastic calculus from Section 2.12 to *linearize* the Markov system about this equilibrium. This is the connection between Markov systems and linear control theory originally sought. The resulting system is precisely the linear system [Kalman, 1960, Kalman, 1962] on which countless volumes of attention have been showered. Although it is cast in stochastic algebra rather than matrix algebra, there is an isomorphism between the two as shown in Section 2.9. In the neighbourhood of a Markov equilibrium our (control theoretic) feet are thus on familiar ground. The decentralized results of Wang and Davison [1973] may be immediately exploited, for example. Furthermore, we now have two types of evolution at play in a single mathematical model. First, the natural tendency of the system to head to disorder by decaying towards the uniform distribution, $\omega$, which is the (stable) zero vector. Second, the ability to create order by making the zero vector unstable through appropriate control sequences. Bushev [1994] describes these two mechanisms.

> Conventionally, we called the evolution from order to disorder thermodynamic evolution and opposed it to the evolution from chaos to order which, against convention, was called biological evolution.

In our linear stochastic model the evolution towards and away from the zero vector follows directly from the eigenvalues of the system. It may seem strange that order arises from purposely creating an instability; typically in control we are fighting an instability not creating one. This notion, too, has a parallel in nonlinear physics, Prigogine's theorem. Bushev [1994] writes

> Prigogine's theorem, known also as the principle of minimum entropy production, states: given the external conditions hindering the system to fall into equilibrium, the stationary state corresponds to minimum entropy production ... In a stationary state the entropy production is exactly compensated by an outflow of entropy into the surrounding medium, so that the total entropy of the body remains constant ... It should be pointed out that the proof of Prigogine's theorem rests completely on the principle of Onsager [which is] why the theorem holds only in the linear (Onsager) region.

Similar behaviours may be seen here. Applying a fixed control vector, the system state gradually creeps away from the (naturally stable) zero vector until there is a balance between the external condition (our control) and the natural tendency to decay whereupon it comes to rest (see Figure 3.4) some distance away from zero. If the system is controllable and observable, it is possible to use appropriate feedback to achieve any desired nonzero state, but beyond the linear approximation region the nonlinear stochastic equations of the original problem must be used.

The stochastic Markov systems examined here incorporate a number of ideas from control theory, nonlinear physics, and probability theory in a single mathematical model. Not all of the work presented

here is immediately applicable to a real world system. There are still several questions to be answered before that is possible. A major issue is generation of the mathematical models (i.e., transition and observation matrices). How do we create such models? How accurate do they need to be? Will simple decentralized controllers have the computational resources necessary to deal with such models? Perhaps the most useful aspect is the deeper understanding of decentralized systems they afford. This thesis finds communication to be a much more viable approach to decentralized control of real world systems than simply trying to behave optimally without communication (e.g., solving a DecPOMDP directly). The role of communication for both sensory sharing and action coordination is immediately applicable to real world problems, particularly when simple reactive decentralized controllers are to be used. At the very least, Markov systems seem a promising candidate for future studies of both stochastic decentralized systems and self-organization.

To demonstrate the usefulness of this knowledge, Chapter 4 described a small network of mobile robots that were able to communicate by radio modem. A number of tasks requiring the robots to communicate were implemented on this system (i.e., clustering, distributing, formation, flocking). The methodology of building up complex behaviours through interactions between robots was found to be fast, modular, and robust. It was furthermore argued that in the case of finite bandwidth communications, decentralized control may be the only alternative as its use of the communication facilities is often much more efficient (depending on the task) than centralized control. For this reason, strictly centralized control is not possible for the robotics system described here. Generic modules for sensory sharing and action coordination were presented. It was argued that if communication occurred quickly enough, the decentralized controllers could be designed by assuming instantaneous communication. This was called the two-timescale assumption[2]. It is a generalization of the assumption of centralized control to different amounts of assumed instantaneous communication. From this perspective it does not really make sense to distinguish between centralization and decentralization. It makes a great deal more sense to speak of the amount of information that can be quickly communicated throughout the system (centralized control being the limiting case of all possible information). Due to the potentially more efficient use of communication afforded by decentralized control (depending on the task), the two-timescale assumption may be valid for decentralized control but not centralized control. This assumption was found to be valid for the tasks carried out by the robotics system in Chapter 4.

The notion of *two-timescale behaviour* appears to be an absolutely fundamental ingredient in global coordination of decentralized systems. This theme may be seen throughout this thesis. In the chapter on control we needed to assume communication occurred much more quickly than the control in order for decentralized controllers to implement all types of centralized controllers. In the chapter on robotics we saw that in real world systems, when the two-timescale assumption is valid, design of decentralized controllers is greatly simplified as we can assume communication occurs instantaneously. This is in fact the very assumption of centralized control[3]. However, by using decentralized control the assumption is more likely to be valid as the communications are used more efficiently (depending on the task). Finally, in the chapter on stochastic cellular automata we saw that it was necessary to slow down the local decision-making process (decreasing $\lambda$ from 1 to about 0.6 in the $\pi$-map) so that the local averaging

[2] Analogous to the *adiabatic assumption* described in [Haken, 1983a] which is also based on two-timescales.
[3] Note, there has been much work on the issue of time delay in centralized systems.

(through the $\sigma$-maps) could occur quickly enough to achieve global coordination . Again, two timescales. This concept is not new. It is used, for example, in nonlinear physics to show the emergence of structure in Rayleigh-Bénard convection. It is used in chemistry to show spatio-temporal structures emerging in the Belousov-Zhabotinski reaction.

In physics such phenomena as strange attractors (e.g., Lorenz attractor) and large scale oscillations (e.g., Brusselator) are described by stiff[4] systems of equations. Having different timescales can introduce a phenomenon sometimes called *slaving* [Haken and Wagner, 1973, Haken, 1983b] in which the fast modes (usually stable) of the system are slaved to a small number of (usually unstable) slower modes. Thus the presence of an instability can also be important. From the point of view of an observer operating at the slow timescale, there is effectively a reduction in the degrees of freedom of the system, whence the illusion of an 'invisible hand' guiding all parts of the system at once. But it is only an illusion[5]. In the stochastic cellular automata, the slow, unstable modes are embodied by the $\pi$-maps (e.g., when $\lambda = 0.6$) at each cell. The fast, stable modes are embodied in the $\sigma$-maps which tries to reduce the differences between connected cells.

The mere presence of multiple timescales certainly does not imply interesting things will happen, that the system will *self-organize*, but it does appear to be a crucial ingredient. But how could it be otherwise? How else could sparsely communicating autonomous agents coordinate their behaviour in interesting ways? It seems obvious that they must communicate quickly enough to arrange their behaviour before being required to act. The speed at which information may be globally shared between autonomous agents imposes an insurmountable upper bound on the types of behaviour the group may exhibit. But simply sharing information is not enough. Instability and fluctuation are also helpful in a truly self-organizing system in order to break symmetries, to force the group, with no external guidance, to select one path over another. Although this field of research has only begun to mature, it would appear that there *are* general mathematical features of self-organizing systems that may be identified, understood, and harnessed. This is perhaps not surprising, as mathematics has, on countless occasions previously, revealed well guarded secrets of nature. Although we have only begun to understand, it seems likely (to this author), nay inevitable, that these basic mechanisms one day provide the palette from which engineers craft a whole new type of technology.

[4] 'Stiff' systems of equations are those in which there are both 'fast' (large) and 'slow' (small) eigenvalues. Typically complex systems are nonlinear; thus when we speak of eigenvalues we mean in the neighbourhood of some equilibrium.

[5] The careful observer of a large flock of birds, for example, may notice a change in direction of the flock does not occur with all birds at the same instant but rather in wave-like motions.

## 6.1 Conclusions

It is difficult to reduce the essence of this subject down to a few concise statements but in an effort to be clear, this will be attempted. The conclusions of this thesis may be stated as follows.

I Irreversibility (and hence stochasticity) plays an important role in self-organization and thus must be accounted for in a general theory of decentralized systems. The decentralized Markov control framework, cast in stochastic algebra, is a promising candidate for future studies of decentralized control and self-organization.

II Markov systems may be linearized about an equilibrium (using stochastic calculus) and thus connected to the classic linear system studied by Kalman [1960].

III The use of communication in developing solutions to DecPOMDPs is a much more promising avenue of research than trying to behave optimally without communication (e.g., solving a DecPOMDP directly). The resulting performance is higher with less computational effort expended. The cost is the implementation of communication.

IV Decentralized controllers can implement any 'centralized' behaviour using communication (assuming adequate bandwidth). The dual purpose of communication is to share sensory data and coordinate actions.

V Decentralized control can make much more efficient use of communication facilities than strictly centralized control as the amount of communicated information is task dependent.

VI Both sensory sharing and action coordination are viable mechanisms in a network of real mobile robots that eliminate any need to use centralized control.

VII Decentralized action coordination in a network of arbitrarily connected agents may be facilitated by a stochastic decision making process. Stochastic cellular automata serve well in this regard.

VIII The stochastic cellular automata (SCA) model is a simple model of self-organization in a network of arbitrarily connected decision makers. Decentralized decision making should be thought of in terms of an instability and is an inherently stochastic process (it requires fluctuations to work).

IX Successful decentralized decision making (in the SCA model) occurs in the vicinity of a critical phase transition where long transient behaviour ensues. Critical slowing down is vital as it allows information to be communicated globally before the local decision making processes are final. This is an example of two-timescale behaviour, a fundamental mechanism of self-organization.

X In self-organizing systems, an observer of slow timescale behaviour may conclude the system is controlled by a central agency but this is an illusion afforded by coordination occurring at the (unobserved) fast timescale.

XI There are fundamental aspects of self-organization that may be understood and exploited in the design of artificial decentralized systems.

## 6.2 Contributions

Having summarized the thesis and stated the conclusions, the main contributions of this work are now identified.

**STOCHASTIC ALGEBRA**: This mathematical framework which treats probability distributions as vectors is original work. It is built on stochastic matrices, an old concept. Its most important contributions are the following.

I Allows stochastic dynamic equations to be written in a well-known algebraic framework.

II Allows Markov systems to be cast in a rigourous matrix format.

III Makes connections between linearity and statistical independence, zero vector and uniform probability distribution.

**STOCHASTIC CONTROL SYSTEMS**:

I Markov systems were cast in a new stochastic algebra which is a matrix formulation[6].

II It was shown that decentralization of a Markov control problem places two constraints on the system, equivalent to removing sensors and actuators. Centralized control is equivalent to decentralized control with communication.

III Four classes of centralized Markov reactive controllers were defined based on type and amount of communication needed to be exactly implementable by decentralized controllers (Theorem 4).

IV A new stochastic calculus was used to linearize both Markov chains and Markov decision processes in the neighbourhood of an equilibrium probability distribution.

V A connection was made between Markov systems and the well known linear system (both centralized [Kalman, 1960] and decentralized [Wang and Davison, 1973]) through linearization and an isomorphism.

VI A connection was made between stability (using the stochastic eigen problem) of an equilibrium distribution and the creation/destruction of information (order, entropy) in a stochastic Markov system.

**COLLECTIVE ROBOTICS**:

I A network of 6 mobile robots was constructed to test various aspects of decentralized control with communication. Design and work done in partnership with E J P Earon with the help of several others (see Acknowledgements). The use of interrobot radio communications in a group of mobile robots is not entirely original but this is one in a handful of facilities with this capability.

II Various decentralized controllers were demonstrated using communication on the above facility. Tasks requiring both sensory sharing and action coordination were demonstrated thus validating the important role communications can play in a real world decentralized system.

[6] Although Markov chains have been written using stochastic matrices, the DecPOMDP model was not until this thesis.

**SELF-ORGANIZATION:**

I *Stochastic cellular automata* (SCA) were introduced as a very simple model of self-organization for a group of arbitrarily connected cells.

II Fundamental properties of self-organization were identified in SCA (i.e., averaging, instability, critical slowing down, fluctuation).

III A simple coordination mechanism was introduced which may be used in a practical system to make decentralized decisions in a group of arbitrarily connected decision-makers.

## 6.3 Open Questions

This section discusses possibilities for future work while the next deals with more philosophical issues.

We need to continue to identify and understand the basic mechanisms of self-organization. Stochastic algebra (and calculus) works with discrete states and discrete time. Extensions to both continuous states and time may be possible. Considering self-organization from the control perspective is quite natural as we may discuss both internal feedback and external pressures. This framework allows both reversible and irreversible trajectories, the forgetting of initial conditions and creation of new structures. Markov systems, cast in stochastic algebra, provide a solid framework for further discussion of these concepts.

How can we improve on the stochastic decentralized decision making process? We can first study different $\pi$-maps to see the possible variation in group behaviour. This will likely lead to incremental improvements. The method presented here assumes no particular connective architecture (i.e., the $\sigma$-maps) between decision makers; it is quite general. This was a design requirement. We know we can do better than this by exploiting the nature of the connections between agents [Das et al., 1995], [Andre et al., 1997]. We should further explore methods (e.g., genetic techniques) that automatically allow the connective architecture to be exploited in an online, adaptive fashion. This will likely again require consideration at different timescales (e.g., connections change more slowly than decisions need to be made). A better understanding of decentralized decision making is key to coordination of many agents, both software and robotic. Exploitation of the connective architecture will allow more complex spatio-temporal behaviours to be designed in a sparsely connected network of agents.

Two-timescale behaviour was seen to play an important role but this notion easily generalizes to an arbitrary number of timescales. Can we make use of an entire hierarchy of timescales? Along similar lines, this thesis almost exclusively studied what might be termed a "flat architecture" in which all agents exist on a single level. There are many organizations in nature (e.g., social insects, human society) which work with some form of hierarchical structure (e.g., workers, managers, vice-presidents, president). Does this type of structure arise in nature because it is more efficient or effective than a flat architecture? If so, can a hierarchical structure be specified from the bottom up rather than from the top down?

One exception to the flat architecture in this thesis was the robot flocking experiment of Chapter 4. In this task one robot was spontaneously elected as leader. This simple example shows that it may be possible to build up from a flat architecture to a hierarchy using some method of spontaneous task division. In this way, the system may be both self-organizing (it literally builds its own organization of leaders) while at once taking advantage of some form of hierarchical command (in which action

coordination might be simplified). In the event of a malfunction, new robots will automatically move up the ladder to replace their failing leaders. This seems a natural route to take for artificial societies and a major possibility for future work in collective robotics. Studies in this area may also provide insight into the prevalence of this type of organization in nature.

Scaling up, as discussed in Section 5.4, is a major issue in large decentralized systems. Can we design systems at one scale and then apply them at another? Are there general scaling relationships which may be understood and used to predict the scaled up behaviour? The robotics system from Chapter 4 was designed to allow a certain amount of scaling (e.g., it was designed for 1-15 robots) but will not scale past a certain level as the communication bandwidth is finite and it relies somewhat on each robot being able to transmit to every other robot. The use of sparse connections between agents is able to alleviate the communication bandwidth problem, but as was seen with stochastic cellular automata, this makes coordination more difficult. Larger systems will require either more bandwidth or a longer period of time to facilitate coordination. This is a common problem to all decentralized systems and a better understanding of scaling is needed.

This thesis almost exclusively dealt with systems that were designed to cooperate in order to achieve some task. However, it is possible that a number of agents might not be cooperative (e.g., due to a malfunction). We would certainly like decentralized systems to be robust to a small percentage of uncooperative members. This is definitely a game-theoretic issue which will become more important as these types of system scale up. It is inevitable that uncooperative agents will arise and robust methods need to be developed to deal with this problem.

Another issue which was beyond the scope of this thesis is design of decentralized systems. We have seen in a few contexts that decentralized systems can perform just as well as a centralized system if communication is allowed. But what about at the design level? Do we need to use some form of central design facility when constructing decentralized systems or will we again find that this is unnecessary? Can one decentralized system be designed by another decentralized system (with communication)? This may be investigated in the context of machine learning methods like reinforcement learning [Barfoot et al., 2000] and evolutionary algorithms [Earon et al., 2000].

We have seen that communication (in one form or another) is the backbone of decentralized systems. The establishment of standard communication protocols is thus key to enabling all kinds of interesting behaviours in large groups of agents. We should devote much effort to the design of common language representations for distributed autonomous agents. This will allow us to construct, from the bottom up, highly robust, modular, artificial decentralized systems.

## 6.4 Final Thoughts

I have tried to refrain from subjectivity until this point but feel I must offer the following opinions for debate. The opening quote of this chapter certainly poses a number of intriguing questions. "Can we build a gradualist bridge from simple amoeba-like automata to highly purposive intentional systems, with identifiable goals, beliefs, etc.?" [Dennett, 1994]. If we prescribe to the materialist philosophy or hold dear the tenets of new AI then we must answer this question with a yes. Are we close to being able to do this? No. We are probably a very long way from creating something the average person would say has "intelligence" comparable to a human. Does this mean we should give up? Certainly not. To assume that human level intelligence is the only worthwhile goal is anthropocentric arrogance. To believe in a gradualist bridge is to believe in an entire spectrum of intelligence from amoeba-like automata to ants to humans (although ants are probably closer to our end of the spectrum than we might like to admit).

From this point of view, the field of artificial intelligence is encompassed by the larger subject of artificial life which is in turn encompassed by the even larger field of self-organization. Historically, scientific research in these areas has grown in an outward direction ("whole to the parts") from the study of intelligence (humans) to life (biology) to general self-organization (physics, mathematics). The new view of building inward ("parts to the whole") has existed for a far shorter time. If we really want to create systems that harness abilities found in nature this new approach would seem to be much more compatible with the engineering objective. Furthermore, if we believe Darwin then we have an existence proof that gradually building up works. In the most general terms, natural selection itself *is* a stochastic decentralized process. But this means we must start at the most basic level of fundamental building blocks and begin constructing from there. The key mechanisms of self-organization should be carefully identified and then enabled artificially. Here we must stop to consider that if irreversibility and stochasticity are an inherent part of life, then will we even recognize our artificial creations as being alive? Schrödinger [1956] speculated that the essence of biological life is the evasion of thermodynamic equilibrium:

> What is the characteristic feature of life? When is a piece of matter said to be alive? When it goes on 'doing something', moving, exchanging material with its environment, and so forth, and that for a much longer period than we would expect an inanimate piece of matter to 'keep going' under similar circumstances... After that the whole system fades away into a dead, inert lump of matter. A permanent state is reached, in which no observable events occur. The physicist calls this the state of thermodynamic equilibrium, or of 'maximum entropy'.

If life or even intelligence can be recreated in an artificial medium will it be recognizable to us? Will it have meaning to us? Will it be useful to us? Will we be useful to it? Will we have meaning to it? Will it recognize us? There are certainly ethical questions which may be raised. Should we proceed? Perhaps not, but we will.

> ... there is a tendency to forget that all science is bound up with human culture in general, and that scientific findings ... are meaningless outside their cultural context. A theoretical science ... where the initiated continue musing to each other in terms that are, at best, understood by a small group of close fellow travellers, will necessarily be cut off from the rest of cultural mankind; in the long run it is bound to atrophy and ossify however virulently esoteric chat may continue within its joyfully isolated groups of experts.
>
> —Erwin Schrödinger
>
> ARE THERE QUANTUM JUMPS?, 1952

# BIBLIOGRAPHY


[Aicardi, 1995] Aicardi, M. (1995). Coordination and control of a team of mobile robots. In *Proceedings of the INRIA/IEEE Symposium on Emerging Technologies and Factory Automation*, volume 2, pages 335–344.

[Aicardi et al., 1987] Aicardi, M., Davoli, F., and Minciardi, R. (1987). Decentralized optimal control of markov chains with a common past information set. *IEEE Transactions on Automatic Control*, AC-32(11):1028–1031.

[Andre et al., 1997] Andre, D., Bennett, F. H., and Koza, J. R. (1997). Evolution of intricate long-distance communication signals in cellular automata using genetic programming. In Langton, C. G. and Shimohara, K., editors, *Artificial Life V, Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*. MIT Press.

[Antsaklis and Michel, 1997] Antsaklis, P. J. and Michel, A. N. (1997). *Linear Systems*. McGraw-Hill, New York.

[Aoki, 1972] Aoki, M. (1972). On feedback stabilizability of decentralized dynamic systems. *Automatica*, 8:163–173.

[Åstrom, 1965] Åstrom, K. J. (1965). Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10:174–205.

[Bak, 1996] Bak, P. (1996). *How Nature Works, The Science of Self-Organized Criticality*. Springler-Verlag, New York.

[Balch, 1997a] Balch, T. (1997a). Learning roles: Behavioural diversity in robot teams. In *AAAI Workshop on Multiagent Learning*.

[Balch, 1997b] Balch, T. (1997b). Social entropy: A new metric for learning multi-robot teams. In *Proceedings of the 10th International FLAIRS Conference*.

[Balch et al., 2001] Balch, T., Khan, Z., and Velosa, M. (2001). Automatically tracking and analyzing the behaviour of live insect colonies. Montréal, Canada. The 5th International Conference on Autonomous Agents (AGENTS). To appear.

[Barfoot and D'Eleuterio, 1999] Barfoot, T. D. and D'Eleuterio, G. M. T. (1999). An evolutionary approach to multiagent heap formation. In *Proceedings of the Congress on Evolutionary Computation*, Washington D C, USA.

[Barfoot and D'Eleuterio, 2001] Barfoot, T. D. and D'Eleuterio, G. M. T. (2001). Multiagent coordination by stochastic cellular automata. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Seattle, USA.

[Barfoot et al., 1999] Barfoot, T. D., Earon, E. J. P., and D'Eleuterio, G. M. T. (1999). Controlling the masses: Control concepts for network robotics. In *Proceedings of the Second Canadian Space Exploration Workshop*, Calgary, Canada.

[Barfoot et al., 2000] Barfoot, T. D., Earon, E. J. P., and D'Eleuterio, G. M. T. (2000). A step in the right direction: Learning hexapod gaits through reinforcement. In *Proceedings of the 31st International Symposium on Robotics (ISR)*, Montréal, Canada.

[Barfoot et al., 2001] Barfoot, T. D., Earon, E. J. P., and D'Eleuterio, G. M. T. (2001). A new breed: Development of a network of mobile robots for space exploration. In *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Montréal, Canada.

[Barto et al., 1995] Barto, A. G., Bradtke, S. J., and Singh, S. P. (1995). Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72:81–138.

[Baxter and Bartlett, 1999] Baxter, J. and Bartlett, P. L. (1999). Direct gradient-based reinforcement learning: I. gradient estimation algorithms. Technical report, Computer Science Lab., Australian National University.

[Baxter et al., 1999] Baxter, J., Weaver, L., and Bartlett, P. L. (1999). Direct gradient-based reinforcement learning: Ii. gradient ascent algorithms and experiments. Technical report, Computer Science Lab., Australian National University.

[Beckers et al., 1994] Beckers, R., Holland, O., and Deneubourg, J. (1994). From local actions to global tasks: Stigmergy and collective robotics. In Brooks, R. A. and Maes, P., editors, *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems.* MIT Press.

[Bellman, 1957] Bellman, R. (1957). *Dynamic Programming.* Princeton University Press.

[Berstein et al., 2000] Berstein, D. S., Zilberstein, S., and Immerman, N. (2000). The complexity of decentralized control of markov decision processes. In *Proceedings of the 16th International Conference on Uncertainty in AI.*

[Bonabeau et al., 1994] Bonabeau, E., Theraulaz, G., Arpin, E., and Sardet, E. (1994). The building behaviour of lattice swarms. In Brooks, R. A. and Maes, P., editors, *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems.* MIT Press.

[Bonabeau et al., 1998] Bonabeau, E., Theraulaz, G., and Cogne, F. (1998). The design of complex architectures by simple agents. Working Paper 98-01-005, Santa Fe Institute.

[Boutilier, 1996] Boutilier, C. (1996). Planning, learning and coordination in multiagent decision processes. In *Proceedings of TARK-96.*

[Boutilier, 1999] Boutilier, C. (1999). Multiagent systems: Challenges and opportunities for decision-theoretic planning. *AI Magazine*, 20(4):35–43.

[Brooks, 1986] Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Autonomous Systems*, RA-2:14–23.

[Brooks, 1990] Brooks, R. A. (1990). Elephants don't play chess. *Robotics and Autonomous Systems*, 6:3–15.

[Brooks, 1991] Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence Journal*, 47:139–160.

[Brooks, 1994] Brooks, R. A. (1994). Coherent behaviour from many adaptive processes. In Cliff, D., Husbands, P., Meyer, J. A., and Wilson, S. W., editors, *Simulation of Adaptive Behaviour: from Animals to Animats 3*. MIT Press.

[Brooks, 1997] Brooks, R. A. (1997). From earwigs to humans. *Robotics and Autonomous Systems*, 20:291–304.

[Bushev, 1994] Bushev, M. (1994). *Synergetics: Chaos, Order and Self-Organization*. World Scientific, Singapore.

[Cao et al., 1997] Cao, Y. U., Fukunaga, A. S., and Kahng, A. B. (1997). Cooperative robotics: Antecedents and directions. *Autonomous Robots*, 4(1):1–23.

[Cassandra et al., 1997] Cassandra, A., Littman, M., and Zhang, N. (1997). Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. In *Proceedings of the 13th International Conference on Uncertainty in Artificial Intelligence*.

[Cassandra, 1998] Cassandra, A. R. (1998). *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, Dept. of Computer Science, Brown University.

[Cassandra et al., 1994] Cassandra, A. R., Kaebling, L. P., and Littman, M. L. (1994). Acting optimally in partially observable stochastic domains. In *Proceedings of the 12th International Conference on Artificial Intelligence*.

[Cassandra et al., 1996] Cassandra, A. R., Kaelbling, L. P., and Kurien, J. A. (1996). Acting under uncertainty: Discrete bayesian models for mobile robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[Chrisman, 1992] Chrisman, L. (1992). Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In *Proceedings of the Tenth National Conference on AI (AAAI)*.

[Chu, 1972] Chu, K.-C. (1972). Team decision theory and information structures in optimal control problems - part ii. *IEEE Transactions on Automatic Control*, AC-17(1):22–28.

[Claus and Boutilier, 1997] Claus, C. and Boutilier, C. (1997). The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the AAAI Workshop on Multiagent Learning*.

[Cox and Miller, 1965] Cox, D. R. and Miller, H. D. (1965). *The Theory of Stochastic Processes*. John Wiley and Sons Inc, New York.

[Dagaeff et al., 1997] Dagaeff, T., Chantemargue, F., and Hirsbrunner, B. (1997). Emergence-based cooperation in a multi-agent system. Technical report, University of Fribourg, Computer Science Department, PAI group.

[Darwin, 1839] Darwin, C. (1839). *Journal of Researches into the Geology and Natural History of the Various Counties Visited by H.M.S. Beagle*.

[Das et al., 1995] Das, R., Crutchfield, J. P., Mitchell, M., and Hanson, J. E. (1995). Evolving globally synchronized cellular automata. In Eshelman, L., editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 336–343, San Fransisco, CA.

[Davison, 1976] Davison, E. J. (1976). The robust decentralized control of a general servomechanism problem. *IEEE Transactions on Automatic Control*, AC-21(1):14–24.

[Deneubourg et al., 1990] Deneubourg, J. L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chrétien, L. (1990). The dynamics of collective sorting: Robot-like ants and ant-like robots. In Meyer, J. A. and Wilson, S., editors, *Simulation of Adaptive Behaviour: from Animals to Animats.* MIT Press.

[Dennett, 1991] Dennett, D. C. (1991). *Consciousness Explained.* Little, Brown and Company, Boston.

[Dennett, 1994] Dennett, D. C. (1994). Artificial life as philosophy. In Langton, C. G., editor, *Artificial Life: An Overview.* MIT Press.

[desJardins et al., 1999] desJardins, M. E., Durfee, E. H., Ortiz, C. L., and Wolverton, M. J. (Winter 1999). A survey of research in distributed, continual planning. *AI Magazine*, pages 13–22.

[Drake, 1962] Drake, A. (1962). *Observation of a Markov Process Through a Noisy Channel.* PhD thesis, Dept. of Electrical Engineering, MIT.

[Dyson, 1995] Dyson, F. (1995). The scientist as rebel. In Cornwell, J., editor, *Nature's Imagination.* Oxford.

[Earon et al., 2000] Earon, E. J. P., Barfoot, T. D., and D'Eleuterio, G. M. T. (2000). From the sea to the sidewalk: The evolution of hexapod walking gaits by a genetic algorithm. In *Proceedings of the Third International Conference on Evolvable Systems (ICES)*, Edinburgh, Scotland.

[Earon et al., 2001] Earon, E. J. P., Barfoot, T. D., and D'Eleuterio, G. M. T. (2001). Development of a multiagent robotic system with application to space exploration. In *Proceedings of Advanced Intelligent Mechatronics (AIM01)*, Como, Italy.

[Epstein and Axtell, 1996] Epstein, J. M. and Axtell, R. (1996). *Growing Artificial Societies: Social Science from the Bottom Up.* MIT Press, Cambridge, Massassachusetts.

[Fraleigh and Beauregard, 1989] Fraleigh, J. B. and Beauregard, R. A. (1989). *Linear Algebra.* Addison-Wesley, Massachusetts.

[Goldberg and Matarić, 1999a] Goldberg, D. and Matarić, M. J. (1999a). Coordinating mobile robot group behaviour using a model of interaction dynamics. In *Proceedings of Autonomous Agents.*

[Goldberg and Matarić, 1999b] Goldberg, D. and Matarić, M. J. (1999b). Mobile robot group coordination using a model of interaction dynamics. In McKee, G. and Schenker, P., editors, *Proceedings of the SPIE: Sensor Fusion and Decentralized Control in Robotics Systems II*, pages 63–73.

[Goldsmith and Mundhenk, 1998] Goldsmith, J. and Mundhenk, M. (1998). Complexity issues in markov decision processes. In *Proceedings of the IEEE Conference on Computational Complexity.*

[Graham and Wunderlin, 1987] Graham, R. and Wunderlin, A. (1987). *Lasers and Synergetics.* Springer-Verlag.

[Greub, 1974] Greub, W. (1974). *Linear Algebra*, volume Graduate Texts in Mathematics, 23. Springer-Verlag, New York.

[Haken, 1983a] Haken, H. (1983a). *Advanced Synergetics.* Springer-Verlag, Berlin.

[Haken, 1983b] Haken, H. (1983b). *Synergetics, An Introduction, 3rd Edition.* Springer-Verlag, Berlin.

[Haken, 1984] Haken, H. (1984). *Synergetics: The Science of Structure.* Van Nostrand Reinhold Company, New York.

[Haken, 1987] Haken, H. (1987). *Computational Systems - Natural and Artificial.* Springer-Verlag, Berlin.

[Haken and Mikhailov, 1994] Haken, H. and Mikhailov, A. (1994). *Interdisciplinary Approaches to Nonlinear Complex Systems.* Springer-Verlag, Berlin.

[Haken and Wagner, 1973] Haken, H. and Wagner, M. (1973). *Cooperative Phenomena.* Springer-Verlag, Berlin.

[Hansen, 1998a] Hansen, E. A. (1998a). *Finite-Memory Control of Partially Observable Systems.* PhD thesis, Dept. of Computer Science, University of Massachusetts Amherst.

[Hansen, 1998b] Hansen, E. A. (1998b). Solving pomdps by searching in policy space. In *Proceedings of the 14th International Conference on Uncertainty in Artificial Intelligence.*

[Hansen et al., 1997] Hansen, E. A., Barto, A. G., and Zilberstein, S. (1997). Reinforcement learning for mixed open-loop and closed-loop control. In *Proceedings of the 9th Neural Information Processing Systems Conference.*

[Hanson and Crutchfield, 1995] Hanson, J. E. and Crutchfield, J. P. (1995). Computational mechanics of cellular automata: An example. Working Paper 95-10-095, Santa Fe Institute. Submitted to Physica D, Proceedings of the International Workshop on Lattice Dynamics.

[Hao, 1990] Hao, B.-L. (1990). *Chaos II.* World Scientific, Singapore.

[Ho and Chu, 1972] Ho, Y.-C. and Chu, K.-C. (1972). Team decision theory and information structures in optimal control problems - part i. *IEEE Transactions on Automatic Control*, AC-17(1):15–21.

[Hochstein et al., 2000] Hochstein, L., Lerner, S., Clark, J. J., and Cooperstock, J. (2000). Soccer-swarm: A visualization framework for the development of robot soccer players. In *Proceedings of the 31st International Symposium on Robotics*, pages 108–113. IEEE World Congress on Computational Intelligence.

[Hodgins and Brogan, 1994] Hodgins, J. K. and Brogan, D. C. (1994). Robot herds: Group behaviors for systems with significant dynamics. In Brooks, R. A. and Maes, P., editors, *Artificial Life IV, Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems.* MIT Press.

[Howard, 1960] Howard, R. A. (1960). *Dynamic Programming and Markov Processes.* MIT Technical Press and Wiley, New York.

[Hoyt, 1996] Hoyt, E. (1996). *The Earth Dwellers: Adventures in the Land of Ants.* Simon and Schuster, New York.

[International Space University, 1993] International Space University (1993). International lunar farside observatory and science station. WWW. http://exploration.jsc.nasa.gov/explore/Data/Lib/DOCS/EIC039.HTML.

[iRobot, 2000] iRobot (2000). Swarm: Distributed programming of autonomous robots. WWW. http://www.irobot.com/rd/research_swarm.htm.

[Kaebling et al., 1998] Kaebling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101.

[Kalman, 1960] Kalman, R. E. (1960). On the general theory of control systems. In *Proceedings of the First International Congress of the IFAC*, pages 481–492, Moscow. Butterworths, London.

[Kalman, 1962] Kalman, R. E. (1962). Canonical structure of linear dynamical systems. *Proceedings of the National Academy of Science, USA*, 48:596–600.

[Kauffman and Johnsen, 1991] Kauffman, S. A. and Johnsen, S. (1991). Co-evolution to the edge of chaos: Coupled fitness landscapes, poised states, and co-evolutionary avalanches. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II: SFI Studies in the Sciences of Complexity, vol. X.* Addison-Wesley.

[Kemeny and Snell, 1976] Kemeny, J. G. and Snell, J. L. (1976). *Finite Markov Chains.* Springer, New York.

[Khalil, 1996] Khalil, H. K. (1996). *Nonlinear Systems.* Prentice Hall, New Jersey.

[Kitano et al., 1997] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., and Osawa, E. (1997). Robocup: The robot world cup initiative. In *Proceedings of the First International Conference on Autonomous Agents*, Marina del Ray. ACM Press.

[Kube and Zhang, 1992] Kube, R. and Zhang, H. (1992). Collective robotic intelligence. In Meyer, J. A., Roitblat, H., and Wilson, S. W., editors, *Simulation of Adaptive Behaviour: from Animals to Animats 2.* MIT Press.

[Kube and Zhang, 1993] Kube, R. and Zhang, H. (1993). Collective robotics: From social insects to robots. *Adaptive Behaviour*, 2(2):189–219.

[Kube and Zhang, 1996] Kube, R. C. and Zhang, H. (1996). The use of perceptual cues in multi-robot box-pushing. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 2085–2090.

[Langton, 1989] Langton, C. G. (1989). Artificial life. In Langton, C. G., editor, *Artificial Life: proceedings of an interdisciplinary workshop on the synthesis and simulation of living systems.* Addison-Wesley.

[Langton, 1990] Langton, C. G. (1990). Computations at the edge of chaos: Phase transitions and emergent computation. *Physica D*, 42:12–37.

[Langton, 1991] Langton, C. G. (1991). Life at the edge of chaos. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II: SFI Studies in the Sciences of Complexity, vol. X.* Addison-Wesley.

[Littman, 1994] Littman, M. L. (1994). Memoryless policies: Theoretical limitations and practical results. In Cliff, D., Husbands, P., Meyer, J. A., and Wilson, S. W., editors, *Simulation of Adaptive Behaviour: from Animals to Animats 3.* MIT Press.

[Littman et al., 1995a] Littman, M. L., Cassandra, A. R., and Kaebling, L. P. (1995a). Learning policies for partially observable environments: Scaling up. In *Proceedings of the 12th International Conference on Machine Learning.*

[Littman et al., 1995b] Littman, M. L., Dean, T. L., and Kaelbling, L. P. (1995b). On the complexity of solving markov decision problems. In *Proceedings of the 11th International Conference on Uncertainty in AI.*

[Madani et al., 1999] Madani, O., Hanks, S., and Condon, A. (1999). On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems. In *AAAI.*

[Maris and te Boekhorst, 1996] Maris, M. and te Boekhorst, R. (1996). Exploiting physical constraints: Heap formation through behavioural error in a group of robots. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Senri Life Science Center, Osaka, Japan.

[Marschak, 1955] Marschak, J. (1955). Elements for a theory of teams. *Management Science*, 1:127–137.

[Matarić, 1992] Matarić, M. J. (1992). Designing emergent behaviours: From local interactions to collective intelligence. In Meyer, J. A., Roitblat, H., and Wilson, S. W., editors, *Simulation of Adaptive Behaviour: from Animals to Animats 2.* MIT Press.

[Matarić, 1994a] Matarić, M. J. (1994a). *Interaction and Intelligent Behaviour (Robotics).* PhD thesis, Artificial Intelligence Laboratory, MIT.

[Matarić, 1994b] Matarić, M. J. (1994b). Learning to behave socially. In Cliff, D., Husbands, P., Meyer, J. A., and Wilson, S. W., editors, *Simulation of Adaptive Behaviour: from Animals to Animats 3.* MIT Press.

[Matarić, 1995] Matarić, M. J. (1995). Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, 16:321–331.

[Matarić, 1997a] Matarić, M. J. (1997a). Behaviour-based control: Examples from navigation, learning, and group behaviour. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2):232–336. Special Issue on Software Architectures for Physical Agents, Editors: H. Hexmoor, I. Horswill, D. Kortenkamp.

[Matarić, 1997b] Matarić, M. J. (1997b). Learning social behaviour. *Robotics and Autonomous Systems*, 20:191–204.

[Matarić, 1997c] Matarić, M. J. (1997c). Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83.

[Matarić, 1998] Matarić, M. J. (1998). Using communication to reduce locality in distributed multi-agent learning. *Journal of Experimental and Theoretical AI, special issue on Learning in DAI Systems*, 10(3):357–369.

[Matarić and Cliff, 1996] Matarić, M. J. and Cliff, D. (1996). Challenges in evolving controllers for physical robots. *Robotics and Autonomous Systems*, 19:67–83.

[McCallum, 1992] McCallum, R. A. (1992). First results with utile distinction memory for reinforcement learning. Technical Report 446, Dept. Computer Science, University of Rochester. Proceedings of Machine Learning Conference 1993.

[Melhuish et al., 1998] Melhuish, C., Holland, O., and Hoddell, S. (1998). Collective sorting and segregation in robots with minimal sensing. In Pfeifer, R., editor, *Simulation of Adaptive Behaviour: from Animals to Animats 5.* MIT Press.

[Minar et al., 1996] Minar, N., Burkhart, R., Langton, C., and Askenazi, M. (1996). The swarm simulation system: A toolkit for building multi-agent simulations. Technical report, Santa Fe Institute, http://www.santafe.edu/projects/swarm/.

[Mitchell et al., 1996] Mitchell, M., Crutchfield, J. P., and Das, R. (1996). Evolving cellular automata with genetic algorithms, a review of recent work. In *Proceedings of the First International Conference on Evolutionary Computation and Its Applications*, Moscow, Russia. Russian Academy of Sciences.

[Mitchell et al., 1993] Mitchell, M., Hraber, P. T., and Crutchfield, J. P. (1993). Revisiting the edge of chaos: Evolving cellular automata to perform computations. *Complex Systems*, 7:89–130. SFI Working Paper 93-03-014.

[Nazin and Poznyak, 1986] Nazin, A. V. and Poznyak, A. S. (1986). *Adaptive Choice of Variants.* Nauka, Moscow. in Russian.

[Nicolis and Baras, 1984] Nicolis, G. and Baras, F. (1984). *Chemical Instabilities.* D. Reidel Publishing Company, Dordrecht, Holland.

[Nicolis and Prigogine, 1977] Nicolis, G. and Prigogine, I. (1977). *Self-Organization in Non-Equilibrium Systems.* John-Wiley and Sons, Inc., New York.

[Ono and Fukumoto, 1997] Ono, N. and Fukumoto, K. (1997). A modular approach to multiagent reinforcement learning. In Weiss, G., editor, *Distributed Aritificial Intelligence Meets Machine Learning.* Springer.

[Ooi et al., 1997] Ooi, J. M., Verbout, S. M., Ludwig, J. T., and Wornell, G. W. (1997). A separation theorem for periodic sharing information patterns in decentralized control. *IEEE Transactions on Automatic Control*, AC-42(11):1546–1550.

[Papadimitriou and Tsitsiklis, 1987] Papadimitriou, C. H. and Tsitsiklis, J. N. (1987). The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3):441–450.

[Papoulis, 1965] Papoulis, A. (1965). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill Book Companay, New York.

[Parr and Russell, 1995] Parr, R. and Russell, S. (1995). Approximating optimal policies for partially observable stochastic domains. In *Proceedings of the 14th International Joint Conference on AI*.

[Peshkin et al., 2000] Peshkin, L., Kim, K.-E., Meuleau, N., and Kaelbling, L. P. (2000). Learning to cooperate via policy search. In *Proceedings of the 16th International Conference on Uncertainty in AI*.

[Peshkin et al., 1999] Peshkin, L., Meuleau, N., and Kaelbling, L. P. (1999). Learning policies with external memory. In Bratko, I. and Dzeroski, S., editors, *Proceedings of the 16th International Conference on Machine Learning*, pages 307–314.

[Pfeifer and Maris, 1994] Pfeifer, R. and Maris, M. (1994). Cheap designs: Exploiting the dynamics of the system-environment interaction: Three case studies on navigation. In *Conference on Prerational Intelligence - Phenomenology of Complexity Emerging in Systems of Agents Interacting Using Simple Rules*, Center for Interdisciplanary Research, University of Bielefeld.

[Poznyak and Najim, 1999] Poznyak, A. S. and Najim, K. (1999). Adaptive control of contrained finite markov chains. *Automatica*, 35:777–789.

[Prigogine and Stengers, 1984] Prigogine, I. and Stengers, I. (1984). *Order Out of Chaos: Man's New Dialogue with Nature*. Bantam Books, Toronto.

[Radner, 1962] Radner, R. (1962). Team decision problems. *Annals of Mathematical Statistics*, 33(3):857–881.

[Ramadge and Wonham, 1987] Ramadge, P. J. and Wonham, W. M. (1987). Supervisory control of a class of discrete event processes. *SIAM Journal of Control and Optimization*, 25(1):206–230.

[Schmidhuber, 1991] Schmidhuber, J. (1991). Reinforcement learning in markovian and non-markovian environments. In Lippman, D. S., Moody, J. E., and Touretzky, D. S., editors, *Advances in Neural Information Processing Systems 3*, pages 500–506. Morgan Kaufmann.

[Schmidhuber, 1996a] Schmidhuber, J. (1996a). A general method for multiagent reinforcement learning in unrestricted environments. In *Proceedings of the AAAI Symposium on Adaptation, Co-evolution and Learning in Multiagent Systems*, pages 84–87.

[Schmidhuber, 1996b] Schmidhuber, J. (1996b). Realistic multi-agent reinforcement learning. In Weiss, G., editor, *Working Notes of the ECAI Workshop, Learning in Distributed Artificial Intelligence Systems*.

[Schmidhuber, 1997] Schmidhuber, J. (1997). Multiagent learning with the success-story algorithm. In Weiss, G., editor, *Distributed Aritificial Intelligence Meets Machine Learning*. Springer.

[Schneider-Fontán and Matarić, 1998] Schneider-Fontán, M. and Matarić, M. J. (1998). Territorial multi-robot task division. *IEEE Transactions on Robotics and Automation*, 14(5):815–822.

[Schrödinger, 1952] Schrödinger, E. (1952). Are there quantum jumps? *The British Journal for the Philosophy of Science*, III:109–110.

[Schrödinger, 1956] Schrödinger, E. (1956). *What is Life?* New York.

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

[Simon, 1969] Simon, H. A. (1969). *The Sciences of the Artificial.* MIT Press, Cambridge, Massachusetts.

[Simsarian and Matarić, 1995] Simsarian, K. T. and Matarić, M. J. (1995). Learning to cooperate using two six-legged mobile robots. In *Proceedings of the Third European Workshop of Learning Robots*, Herkalion, Crete, Greece.

[Smallwood and Sondik, 1973] Smallwood, R. D. and Sondik, E. J. (1973). The optimal control of partially observable markov processes over a finite horizon. *Operations Research*, 21:1071–1088.

[Sondik, 1973] Sondik, E. J. (1973). The optimal control of partially observable markov processes over the infinite horizon: Discounted costs. *Operations Research*, 26(2):282–304.

[Stone, 2000] Stone, P. (2000). *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer.* MIT Press, Cambridge, Massachusetts.

[Sun and Peterson, 1999] Sun, R. and Peterson, T. (1999). Multiagent reinforcement learning: Weighting and partitioning. Technical report, Dept. of Computer Science, University of Alabama.

[Tanaka-Yamawaki et al., 1996] Tanaka-Yamawaki, M., Kitamikado, S., and Fukuda, T. (1996). Consensus formation and the cellular automata. *Robotics and Autonomous Systems*, 19:15–22.

[Thomas, 1974] Thomas, L. (1974). *The Lives of a Cell.*

[Thrun, 1999] Thrun, S. (1999). Monte carlo pomdps. In *Proceedings of Conference on Neural Information Processing Systems (NIPS).*

[Thrun and Langford, 1999] Thrun, S. and Langford, J. C. (1999). Monte carlo hidden markov models. Technical report, Carnegie Mellon University.

[Thrun et al., 1999] Thrun, S., Langford, J. C., and Fox, D. (1999). Monte carlo hidden markov models: Learning non-parametric models of partially observable stochastic processes. In *Proceedings of the International Conference on Machine Learning (ICML).*

[Tsitsiklis and Athans, 1985] Tsitsiklis, J. N. and Athans, M. (1985). On the complexity of decentralized decision making and detection problems. *IEEE Transactions on Automatic Control*, AC-30(5):440–446.

[Versino and Gambardella, 1997] Versino, C. and Gambardella, L. M. (1997). Learning real team solutions. In Weib, G., editor, *Distributed Artificial Intelligence Meets Machine Learning*, pages 40–61. Springer.

[von Neumann, 1966] von Neumann, J. (1966). *Theory of Self-Reproducing Automata.* University of Illinois Press, Urbana and London.

[Wang and Davison, 1973] Wang, S.-H. and Davison, E. J. (1973). On the stabilization of decentralized control systems. *IEEE Transactions on Automatic Control*, AC-18(5):473–478.

[Watkins, 1989] Watkins, C. J. C. H. (1989). *Learning From Delayed Rewards.* PhD thesis, Cambridge University, Cambridge, England.

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

[Watson et al., 1999] Watson, R. A., Ficici, S. G., and Pollack, J. B. (1999). Embodied evolution: Embodying an evolutionary algorithm in a population of robots. In *Congress on Evolutionary Computation*, Washington D C.

[Weiss, 1999] Weiss, G., editor (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* MIT Press, Cambridge, Massachusetts.

[Werner and Dyer, 1992] Werner, G. M. and Dyer, M. G. (1992). Evolution of herding behaviour in artificial animats. In Meyer, J. A., Roitblat, H., and Wilson, S. W., editors, *Simulation of Adaptive Behaviour: from Animals to Animats 2.* MIT Press.

[Wessels, 1997] Wessels, L. F. A. (1997). *Multiagent Reinforcement Learning.* PhD thesis, Faculty of Engineering, University of Pretoria.

[Wheeler and Narenda, 1985] Wheeler, R. M. and Narenda, K. S. (1985). Learning models for decentralized decision making. *Automatica*, 21(4):479–484.

[Wheeler and Narenda, 1986] Wheeler, R. M. and Narenda, K. S. (1986). Decentralized learning in finite markov chains. *IEEE Transactions on Automatic Control*, AC-31(6):519–526.

[Wolfram, 1984] Wolfram, S. (1984). Universality and complexity in cellular automata. *Physica D.* 10:1–35.

[Zhang, 1994] Zhang, N. L. (1994). *A Computational Theory of Decision Networks.* PhD thesis, Dept. of Computer Science, Hong Kong University of Science and Technology.

[Zhang and Liu, 1997] Zhang, N. L. and Liu, W. (1997). A model approximation scheme for planning in partially observable stochastic domains. *Journal of Artificial Intelligence Research*, 7:199–230.