Stochastic Recruitment Strategies for Controlling Artificial Muscles

by

Lael Ulam Odhner

S.B., Massachusetts Institute of Technology (2004)
S.M., Massachusetts Institute of Technology (2006)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2009

Author .....
Department of Mechanical Engineering
August 19, 2009

Certified by .....
H. Harry Asada
Ford Professor of Mechanical Engineering
Thesis Supervisor

Accepted by .....
David E. Hardt
Chairman, Department Committee on Graduate Theses

Stochastic Recruitment Strategies for Controlling Artificial Muscles

by

Lael Ulam Odhner

Submitted to the Department of Mechanical Engineering
on August 19, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Science in Mechanical Engineering

Abstract

This thesis presents a new architecture for controlling active material actuators inspired by biological motor recruitment. An active material is broken down into many small fibers and grouped together to form one large actuator. Each of these fibers is held in a binary state, either relaxed or contracted, using a small local controller which responds to a broadcast input signal from a central controller. The output force and displacement of the actuator is a function of the number of contracted fibers at any point in time. This architecture enables the creation of large-scale, controllable actuators from highly non-linear active materials.

The key innovation enabling the central controller to coordinate the behavior of very many small identical units is to randomize the behavior of each unit. This thesis explains how a collection of active material motor units responding in a random, uncorrelated fashion to broadcast commands will exhibit a predictable response that can be stabilized with feedback control and observed using a Kalman filter. Various control strategies will be presented and discussed, including open-loop plant behavior, linear feedback, optimal control, and model-based look-ahead control. Performance metrics such as accuracy and convergence time will be analyzed using dynamic programming and other control techniques. Parallels will also be discussed between this control problem and similar control problems in the field of swarm robotics.

The stochastic, recruitment-like actuator architecture is demonstrated in shape memory alloy actuators, each composed of 60 individual elements, having a displacement of over 20 mm and a peak force of over 100 N. Control of displacement, isometric force and stiffness are demonstrated using the observer-controller framework. Two actuators are used in an antagonistic fashion to control the stiffness and position of a 1-DOF arm joint.

Thesis Supervisor: H. Harry Asada
Title: Ford Professor of Mechanical Engineering

Acknowledgments

Having spent the last decade in the Department of Mechanical Engineering at MIT, I am indebted to more people than I can recount in the space of a page. All of my teachers have been generous with their time and talents, and have supported me both as a student, as a researcher, and as a person. I would particularly like to thank my undergraduate advisor, Ernesto Blanco, and my undergraduate thesis advisor, Nam Pyo Suh, for their mentorship in my early years here. Additionally, I would like to thank Kim Vandiver, Sanjay Sarma, Seth Lloyd, and Eduardo Kausel for their open door attitudes and their good advice over the years.

It has been a tremendous privilege to study control systems and robotics with the great community of researchers here at MIT. I have thoroughly enjoyed learning from Kamal Youcef-Toumi, Jean-Jacques Slotine, Warren Seering, Ian Hunter, Russ Tedrake, Derek Rowell and Dave Gossard. Most of all, I am grateful to the members of my thesis committee, Neville Hogan and John Leonard, for all their support, and my graduate advisor Harry Asada. Harry has been my friend and mentor for the last five years. He cares about robotics with a passion and he has done a great job of communicating that enthusiasm to myself and my co-workers in the d'Arbeloff Laboratory.

My fellow students at MIT have been a tremendous inspiration to me. I would like to acknowledge the students and staff in my research group, Jun Ueda, Andrew Reisner, Tao Cheng, Eric Wade, Kyu Jin Cho, Phil Shaltis, Devin McCombie, Binayak Roy, Jin Oh Hahn, Levi Wood, Manas Menon, Tom Secord, Chris Brown, Alex Marinkovic, Josiah Rosmarin, Devin Neal, Ani Mazumdar, Dan Cunningham, Waleed Farahat, Patrick Barragan, Alisha Schor, and Ian Rust. I would also like to thank Bryan Ruddy for his thoughtful insight.

Lastly, I would like to thank my best friend, Mariko Jameson, for her constant care in the months leading up to my graduation, and my family, Mom, Dad, Nathan, Johanan, Chelsea, Lamar, Ben and Katie for supporting me from a distance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Artificial Muscle Actuators

The development of new and better actuator technologies is a widely recognized challenge facing the robotics community today. People believe that there are better technologies possible for actuation because the natural example of biological muscle is so compelling. Muscle is an actuator that is well-suited to human tasks such as manipulation and locomotion, that is lightweight, powerful and can fit into form factors that cannot be achieved with electric motors. Inspired by this example, many researchers have sought to produce actuators that mimic the basic properties of muscle. These actuators, including shape memory alloys (SMAs), dielectric elastomers, piezoelectrics, conducting polymers, and polymer gel actuators, produce local stress in response to local energetic input, delivered in the form of charge, heat or chemical energy [3], [24], [38]. Because the material response is intrinsic, the force, displacement and stiffness of the actuator can be tuned by tailoring size, shape and configuration of the active material, in a fashion similar to skeletal muscle [67], [34].

1.2 Control Challenges

The goal of this thesis is to address some of the obstacles to progress in the area of control of artificial muscle actuator materials. Most conventional actuators, such as

motors, solenoids, and capacitive actuators produce force by exerting an electric or magnetic field on a moving or static charge. Because the propagation times associated with electromagnetic fields and electron conduction are so short, it is possible to describe the distributed force produced by these actuators in terms of lumped model parameters, such as the armature torque constant of a permanent magnet motor. As a result, feedback control of electromagnetic field-based actuators is very successful based on models that can safely abstract away localized behavior in the actuator.

One of the principal difficulties faced in controlling the output of many new artificial muscle-like materials is the fact that in order to elicit an intrinsic response from the material, energy must be moved locally into and out of the material, often by thermal diffusion or mass diffusion of chemical species [7]. This diffusion behavior can dominate the response of the material, becase the amount of time it takes to diffuse into a bulk material increases with the square of the diffusion distance [44]. Additionally, the thermodynamic effects central to the actuation process, such as phase transitions, or reaction-diffusion effects, can further complicate models of actuator behavior. As a result of these intrinsic difficulties, as well as gradients due to leakage and non-uniform boundary conditions, it can be difficult to uniformly control the intrinsic material state at all points within the actuator [39]. Critical dimensions of these active materials, such as film thickness or wire diameter, must be thin to enhance diffusive transport. Often, composite construction techniques must be used to increase the local energy mobility to the point where diffusion delays are tolerable [66], [4], [39]. There have been successful attempts at closed-loop control of active materials, including polymer gels [44], conducting polymers [39], and shape memory alloys [37], [62]. Nonetheless, these good results are obtained either by reducing the feedback gain so that the closed-loop bandwidth is sacrificed for stability, or by reducing the size of the active material element being controlled so that the diffusion delays associated with heat or mass transfer are negligible. They do not address the larger question of how these materials can be scaled up to produce large forces or displacements, on scales which require more active material than can be realistically controlled with a servo loop.

1.3 Alternative Control Architectures

This thesis seeks to build on the state of the art in control of small-scale active material actuators. Rather than taking on the herculean task of controlling a large monolithic active material element with a classical feedback loop, the approach taken in this work is to break a large actuator down into many small functional units. The problem of producing a large force or displacement therefore reduces to the problem of coordinating the force or displacement production of very many small elements. Just how much force or displacement can be produced is determined by the number of units in the actuator, not the size of each unit. The other architectural design decision made here is to make each small functional unit discrete, so that it is either actively contracted or passively relaxed. This further simplifies the control problem. If the actuator is commanded to produce 50% of its peak force, it activates 50% of its individual units, rather than commanding each to a 50% response. This is particularly useful for materials such as shape memory alloys, which produce an output that is easier to control as a discrete on-off switching behavior than as a continuously regulated output [56].

This architecture is inspired by the organization of muscle. Muscle is a complex chemically-activated polymer, which activates when exposed to the correct concentration of chemical species [42], [68]. Despite the fact that the activation of skeletal muscle requires reaction-diffusion kinetics that are bound by the same performance constraints as their artificial counterparts, the response time of muscles is fast because the triggering chemical species, calcium, is stored throughout the muscle right next to each fiber. The muscle is organized into very thin fibers that are activated by depolarizing the fiber membrane with a nervous impulse. The calcium, stored outside the fibers in small tubules in much higher concentrations than inside the fibers, diffuses quickly into the thin fibers, which contract rapidly. The overall level of force produced by the actuator is not modulated by fractionally activating all of the muscle fibers; rather, the muscle fibers are grouped into motor units, collections of fibers that are all innervated by a single motor neuron. The motor neurons are activated

sequentially in response to increasing nervous stimulus, so that the fraction of active fibers can be modulated smoothly [20]. This process of activating discrete sub-units of muscle in order to produce varied levels of output is called recruitment.

1.4 Stochastic Recruitment

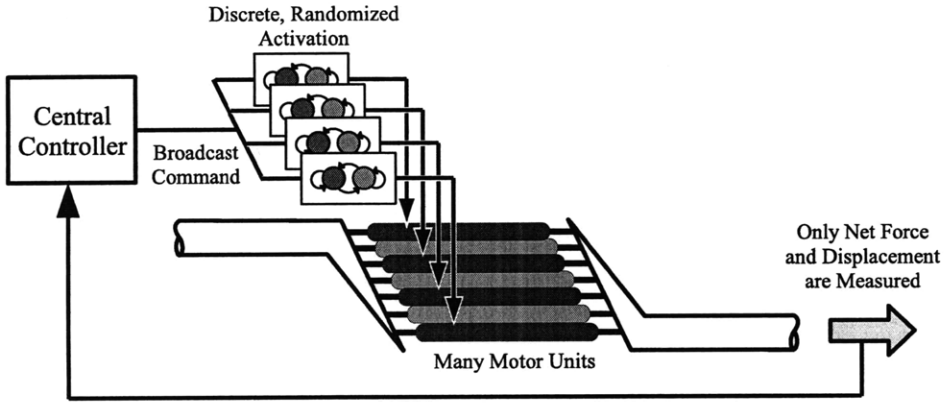

Figure 1-1: The architecture of the proposed recruitment-based actuator.

We wish to develop an architecture for coordinating the behavior of actuators made from many small discrete sub-units, without having to explicitly consider the internal state of each and every unit, as shown in Fig. 1-1. We would like a control architecture which broadcasts a single command to all of the units, and measures the aggregate output of the whole actuator. This, too, draws some inspiration from muscle, where the number of motor units in the muscle is far larger than the number of force and displacement sensors, called golgi tendons and muscle spindles [1]. Without such an architecture, each small actuator must be determined one by one central controller aware of each actuator's state [56]. In past work, great care was taken to minimize the number of units needed to produce the range of desired outputs. Cho *et al.* used data compression techniques to construct an actuator array capable of reproducing functional grasps with a minimal set of discrete actuator units [9]. However, for a general-purpose actuator whose outputs are not specifically constrained to one set of tasks, this approach does not work. The other approach that was specif-

ically avoided was the creation of networks of smart actuator units who coordinate their outputs using distributed computation [12]. The costs associated with inter-unit communication, as well as the costs of computational power per unit, justify avoiding this approach if possible.

The solution we propose is to design each unit to transition between relaxed and contracted states with a controllable probability, uncorrelated from the other units in the actuator. Rather than sending a command signal that somehow selects which agents activate, the central controller broadcasts a signal that modulates the probability with that each unit will turn on. The resulting state transition dynamics, illustrated in Fig. 1-2 are very close to deterministic as the number of units becomes large. The "noise" in the response is also very easy to characterize, since it is the summation of many uncorrelated random variables. This state transition noise, and more conventional sources of disturbance, can be compensated for by feedback control. Simple linear feedback laws are adequate to stabilize the tracking behavior of the actuator. Beyond the simple linear laws, it is possible to use the stochastic nature of the system response to design control laws that perform optimally, in terms of expectation-based performance metrics such as expected convergence time.

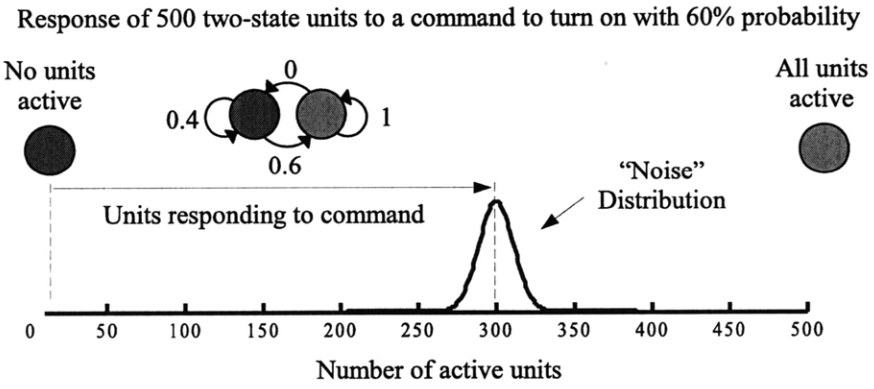

Figure 1-2: The response of many random two-state agents to a command to turn on with probability 0.6. All units are intially inactive.

It is also worth noting that this control architecture could be applied to a wide variety of control problems. Already, similar stochastic, hybrid-state agent models

are used in describing the behavior of swarms of insects and animals [2], [19], or large swarms of robots [33], [5], and gene networks in bacteria [27]. However, most of the work on controlling swarms of this kind is based on synthesizing local rules for agent behavior *a priori*, so that their random transitions lead to the desired aggregate behavior. Choosing these local state transition rules in real time based on feedback measurements is not usually attempted[1]. Because the framework for feedback control presented in this thesis can be generalized from the two-state control of actuator units to the control of any ensemble of hybrid-state units with any number of discrete states, It is applicable to many of these other problems, and has the potential to impact these other problems of distributed control.

1.5 Organization of This Thesis

This thesis develops a framework for modeling and control of the SMA actuators we have built, based on a recruitment-like architecture in which the shape memory alloy is segmented into many independent units, which transition between heated and cooled states randomly, based on a globally broadcast command signal. In chapter 2, the design of the indivdual shape memory alloy units is introduced. These units are described using lumped models to represent the force-length relationship in the heated and cooled states. System identification results for this model are shown for experimental actuator units. This single unit physical model is then used to model the force-length behavior of the entire actuator in various configurations of units, as shown in Fig. 1-3. Additionally, a design is presented for small circuits capable of making random state transitions on a centrally-controlled state distribution.

Chapter 3 presents a dynamic model of the actuator behavior, based on the discrete random response of each actuator unit to the broadcast command. In addition to the 2-state problem immediately relevant to the design of actuator units, chapter 3 also includes a general model of the random behavior of small units having m states. Several probabilistic models for predicting the system response to commands

[1]The work by Julius, Halasz and Pappas in [27] is the notable exception to this generalization.

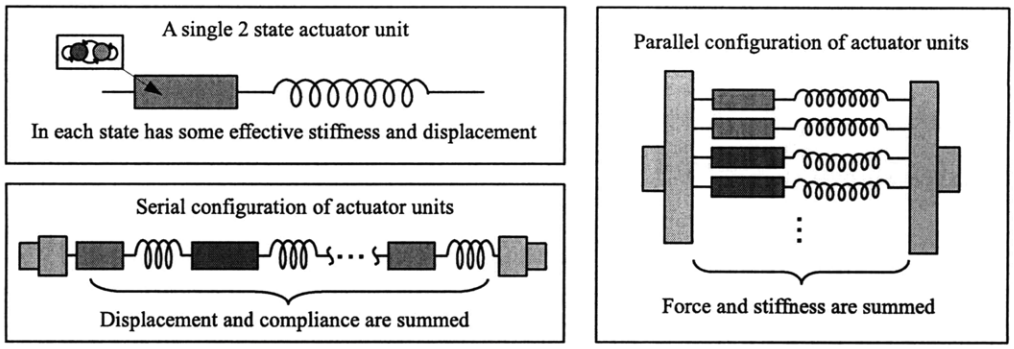

Figure 1-3: Chapter 2 presents an equivalent stiffness model of actuator behavior.

given in the form of a state transition graph are derived, including an exact model, a model based on Markov chain approximations, and a model based on approximating the expectation and covariance of the number of units making state transitions at each point in time. This chapter also includes an in-depth discussion of the open-loop behavior of the stochastic recruitment-based actuator. Constant feed-forward inputs are shown to produce a response that converges on a predictable probability distribution, as shown in Fig. 1-4.

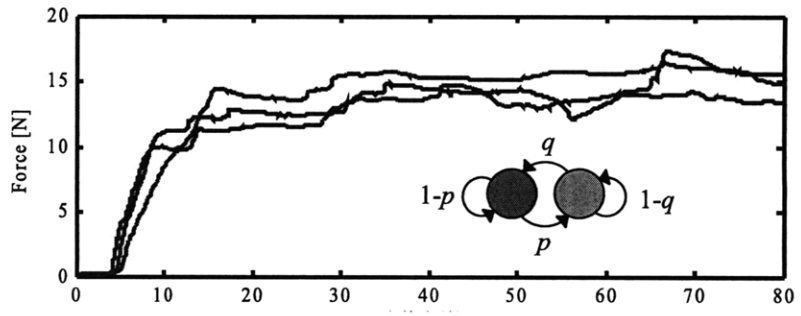

Figure 1-4: The isometric force output of an actuator in response to a constant broadcast state transition graph is shown for three different sets of graph parameters.

Chapter 4 discusses feedback control in the context of the stochastic system models developed in chapter 3. Stability criteria for possible control laws are discussed, and probability one convergence is used as a criterion for determining which control laws are admissible for performance comparison. The control laws are then compared using constant policy value iteration as a means of computing the expectation and

variance of the time the actuator takes to converge to the desired state. The expected convergence time of the actuator is shown to scale favorably compared to the settling time of the active material. An experimental example of closed-loop feedback is presented, as shown in Fig. 1-5.

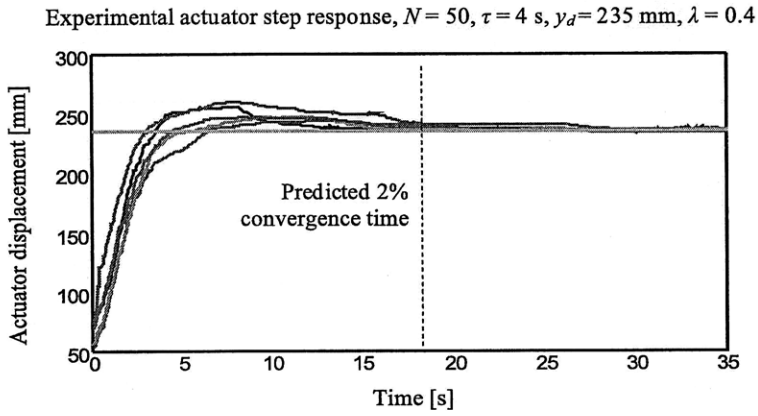

Figure 1-5: A plot of the closed-loop step response of a 50 unit SMA actuator to a position reference.

Chapter 5 discusses the development of state observers for full-state feedback control laws, in the state space observer-controller framework of Fig. 1-6. The random dynamic model of chapter 3 is adapted so that it can be used in a Kalman filter. This general framework for control is very useful for systems whose output is not rich enough to allow for algebraic estimates of the number of actuator units in each discrete state. An experimental example is given, showing how a state observer can be used to approximate the transient dynamics of the actuator repsonse in order to develop control laws which respond quickly to disturbances.

Chapter 6 describes in depth the prototype actuator developed for testing the actuator architecture, depicted in Fig. 1-7. It is a single degree of freedom arm having two antagonistic actuators, each composed of 60 shape memory alloy units. The force and stiffness of each actuator can be adjusted to control the position and angular stiffness of the arm's elbow joint.

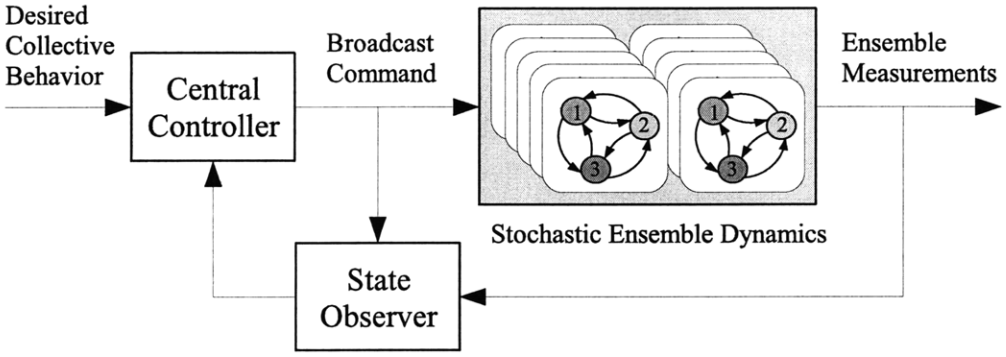

Figure 1-6: A block diagram of the recruitment problem as an observer-controller problem.

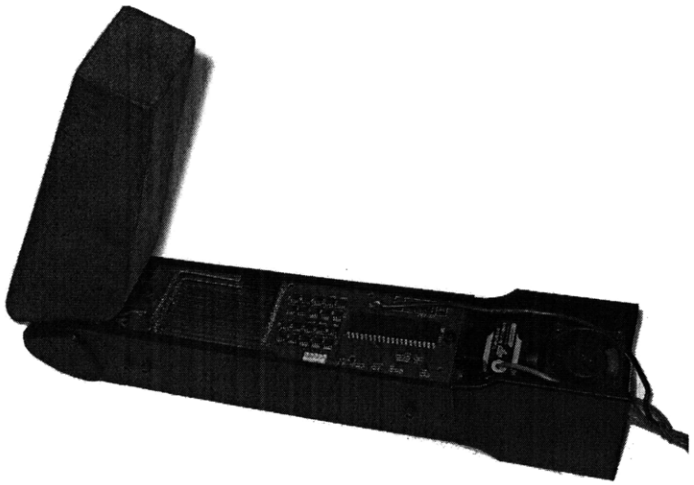

Figure 1-7: The 1-DOF arm developed for validating the stochastic recruitment control architecture.

Chapter 2

Actuator Architecture

2.1 Introduction

This chapter gives an overview of the physical behavior of the recruitment-based system architecture presented in this thesis. It demonstrates how many small binary actuator units can be combined in serial and parallel configurations to produce variable force and displacement. Each unit is modeled as a hybrid-state, quasi-static physical system, having some Thévenin equivalent representation for the contracted and relaxed state. The combination of the individual units into a larger actuator can also be represented with a Thévenin equivalent model, based on the number of units in the contracted state, $N_{on}(t)$, and the number of units in the relaxed state, $N_{off}(t)$.

2.2 Active Material Actuator Units

Figure 2-1 shows a diagram of the individual units to be combined into larger actuators. These actuator units have a hybrid-state design, that is, they have a discrete state component $s_i(t)$ which can take two states, *on* and *off*, and a continuous state component representing the relationship between the actuator's length, $d_i(t)$, and force, $f_i(t)$. The discrete state $s_i(t)$ is determined by a state machine, which controls whether the actuator is contracted (*on*) or relaxed (*off*). This is a simplified architecture analogous to the way a motor neuron controls the activation state of its

associated muscle fibers.

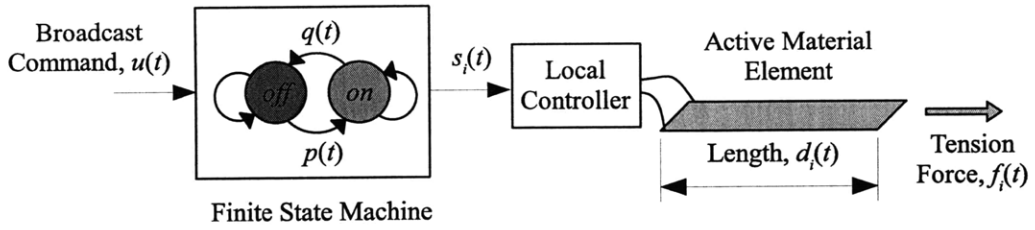

Figure 2-1: The design of an actuator sub-unit for recruitment control.

2.2.1 Units Used in This Work

We chose to use Toki BioMetal Helix wire, part no. BMX150 [10], which provided a contractile displacement of 50% of its relaxed length, at a peak force of roughly 2 N. The wire was Joule heated and convectively cooled. The Joule heating was accomplished using a N-type FET, wired as shown in Fig. 2-2. A driving voltage was chosen that reliably heated the material into the austenite state when heated, in the presence of the forced cross-flow cooling.

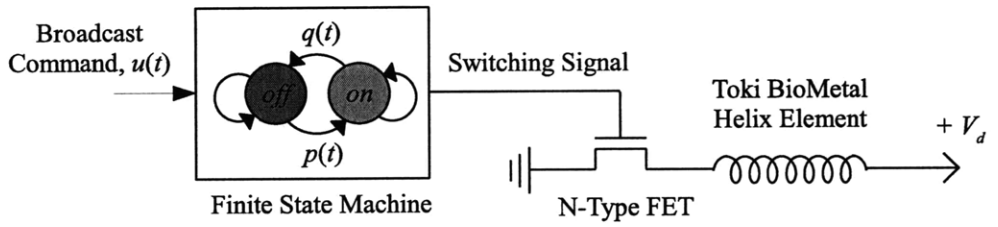

Figure 2-2: The Joule-heated shape memory alloy actuator units used for this work.

2.2.2 A Quasi-Static, Hybrid-State Unit Model

One defining characteristic of most active materials is that they are not well-represented as either force sources or displacement sources. The elastic properties of the material are a strong function of the activation state of the material [51], [53]. One way of describing this state-dependent elasticity is an equivalent circuit model [55], [46]. Fig. 2-3 depicts a Thévenin equivalent model, shown an ideal source in series with an

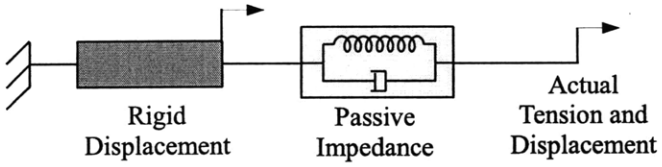

Figure 2-3: An equivalent circuit model for each active material element.

equivalent impedance. Many different active materials can be characterized with this simple model. For a SMA helix, the equivalent impedance can be assumed to purely elastic, as shown in Fig. 2-4[1]. Each element was assumed to have some effective resting length $\delta_i(t)$, and an effective series elasticity $k_i(t)$, describing the relationship between the overall element length $d_i(t)$ and the tension force $f_i(t)$:

$$f_i(t) = k_i(t)(d_i(t) - \delta_i(t)) \tag{2.1}$$

The steady-state equivalent circuit model can be characterized for each of the discrete states once the transients associated with state transitions have settled.

A static mechanical analyzer was constructed to measure the force-displacement relationship of the SMA wire in its relaxed and contracted states. The results are plotted in Fig. 2-5 for a sample having a 40 mm resting length in the martensite phase. A cyclic load was exerted on the element to measure the actuator stiffness while the actuator was heated and cooled to the austenite and martensite phases. The force-displacement data was extrapolated as shown in the figure to determine the relative difference in resting length, $\delta_{off} - \delta_{on}$. The results agree well with the assumed purely elastic model. The difference in resting lengths is found to be 20 mm, and the stiffnesses are found to be 93 mN/mm in the austenite phase, and 83 mN/mm in the martensite phase. The only somewhat surprising result is that the austenite phase is only stiffer than the martensite phase by 12%, whereas Young's modulus is significantly higher for austenite than for martensite [54]. This is probably due to the

[1]There is damping associated with the superelastic effect in shape memory alloys. However, it is complicated to model and small enough that the zero-damping system provides a decent description of the actuator behavior.

observed fact that the pitch of the helix changes as the helical element stretches.

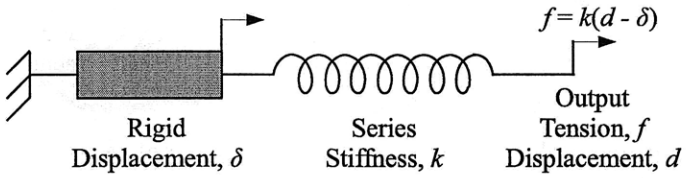

Figure 2-4: An equivalent circuit model for the shape memory alloy actuator units.

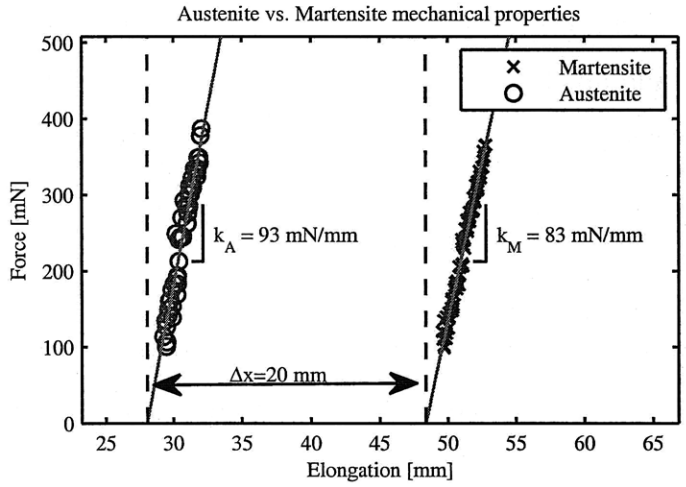

Figure 2-5: The results from a static mechanical analysis of the SMA wire, showing the stiffness and the difference in resting length between the two material states.

2.3 Arranging Units to Tune Actuator Impedance

The geometric configuration of an active material actuator determines force-length output characteristics, like a gearing ratio. This is also true of muscle [34]. Because the lumped stiffness of an active material element scales inversely with length, it stands to reason that a shorter actuator will be both stiffer and produce less displacement. Similarly, stiffness and force will scale proportional to the cross-sectional area of the actuator. This section will describe how the arrangement of many small actuator units can be converted into an equivalent circuit model of overall actuator force-length properties. The goal is to produce a quasi-passive output model similar

in form to (2.1) governing the behavior of the whole actuator,

$$F(t) = K(t)(D(t) - \Delta(t)) \tag{2.2}$$

Where the $F(t)$ is the force produced by the whole actuator, and $D(t)$ is the actuator displacement. The resting length $\Delta(t)$ and stiffness $K(t)$ of the actuator change with the number of actuator units $N_{on}(t)$ in the *on* state, depending on how the units are connected, as shown in Fig. 2-6.

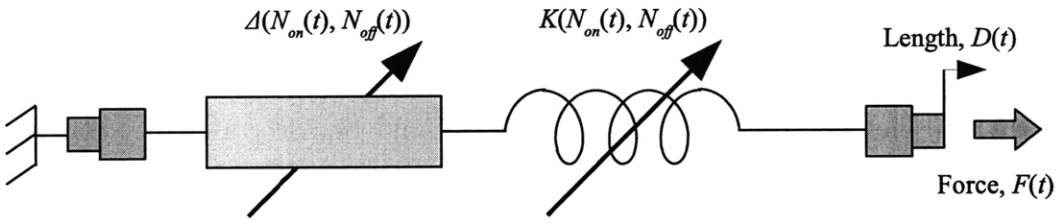

Figure 2-6: The whole actuator can be modeled as an equivalent circuit whose impedance parameters are functions of the number of discrete units in each state.

2.3.1 Serial Configurations

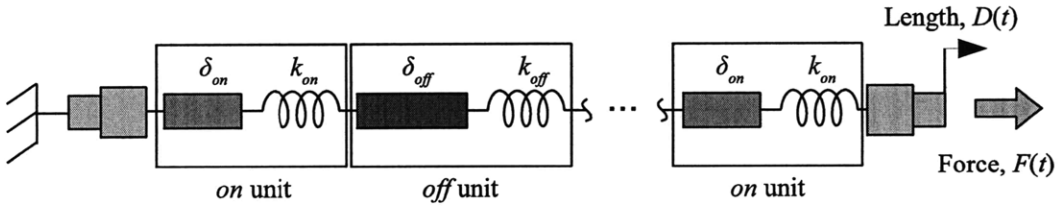

Figure 2-7: A serial chain of hybrid-state units, shown in equivalent circuit form.

Much of the early work done on segmented actuators in [56], [47], [48] was based on serial chains of small elements, as shown in Fig. 2-7. Similar work on arrays of tendon actuators composed of individual units in series was also explored [9]. When actuator units are placed in a serial configuration, the equivalent impedance model for the whole actuator is simple. Because the same force is transmitted through all actuator units, the model from (2.1) can be used for each, with a common tension

force $F(t)$ in each unit of the actuator. This then determines the actuator's net length $D(t)$,

$$D(t) = \sum d_i(t) = \sum \frac{F(t)}{k_i(t)} - \sum \delta_i(t) \tag{2.3}$$
$$F(t) = \left(\sum \frac{1}{k_i(t)}\right)^{-1} \left(D(t) - \sum \delta_i(t)\right)$$

The effective stiffness $K(t)$ can then be found from (2.3),

$$K(t) = \frac{\partial F}{\partial D} = \left(\sum \frac{1}{k_i(t)}\right)^{-1} \tag{2.4}$$

Similarly, the net effective resting length $\Delta(t)$ could be ascertained by finding the displacement for which the force is euqal to zero. This is somewhat trivial for a serial chain,

$$\Delta(t) = \sum \delta_i(t) \tag{2.5}$$

One thing that is important about this configuration is that the compliance and effective resting length can be written in terms of the number of units that are in the *on* and *off* states,

$$\Delta(t) = \delta_{on} N_{on}(t) + \delta_{off} N_{off}(t) \tag{2.6}$$
$$\frac{1}{K(t)} = \frac{N_{on}(t)}{k_{on}} + \frac{N_{off}}{k_{off}} \tag{2.7}$$

This means that tuning the force-length relationship of this actuator is a matter of adjusting several linear functions of the number of units recruited into the *on* state.

2.3.2 Parallel Configurations

When many actuator units are placed in parallel, the displacement of all units is equal to $D(t)$, which is assumed to be greater than the resting length in either the *on* or *off* state, so that all units are in tension. The total force produced is the sum of the

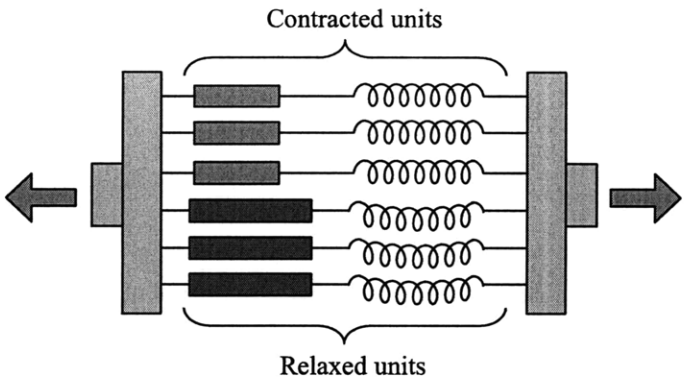

Figure 2-8: In the parallel configuration, each unit's force and stiffness sum to produce an overall output.

force produced by each unit,

$$F(t) = \sum k_i(t)(D(t) - \delta_i) \tag{2.8}$$

This means that the total stiffness is also a function of the stiffnesses of each individual unit,

$$K(t) = \sum k_i(t) = k_{on}N_{on}(t) + k_{off}N_{off}(t) \tag{2.9}$$

The effective resting length of the actuator is a little more difficult to calculate. The effective resting length can be found by setting (2.8) equal to zero,

$$\begin{aligned} \Delta(t)\sum k_i &= \sum k_i\delta_i \\ \Delta(t) &= \left(\sum k_i\delta_i\right)\left(\sum k_i\right)^{-1} \end{aligned} \tag{2.10}$$

One interesting practical case occurs when the actuator displacement is smaller than the resting length in the *off* state, $D(t) < \delta_{off}$. This occurs very frequently when using the Toki BioMetal helical elements, because of the extremely large difference in the resting lengths between the martensite and austenite phases. In this case, many units do not actually contribute to the actuator force or elasticity, but rather remain slack. The resting length $\Delta(t)$ is always equal to the resting length in the *on* state,

δ_{on}, and the stiffness varies linearly with $N_{on}(t)$,

$$\Delta(t) = \delta_{on} \tag{2.11}$$

$$K(t) = k_{on} N_{on}(t) \tag{2.12}$$

Figure 2-9 shows the force-displacement curve for such an actuator, indicating the point on the curve where the *off* units become taut and contribute to the tension force.

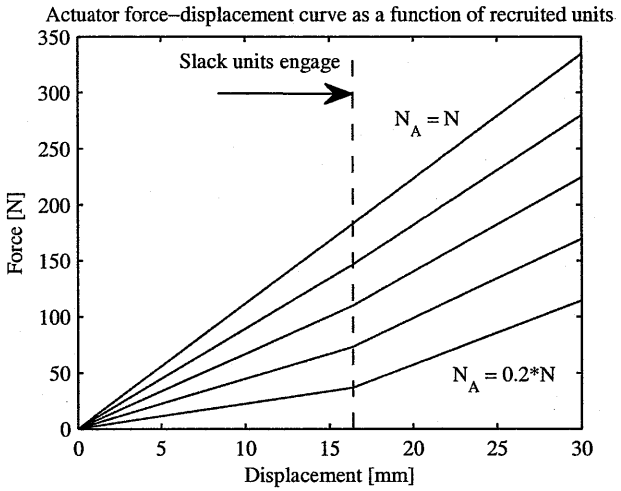

Figure 2-9: The modeled force-displacement curve as a function of the number of contracted units.

2.4 Summary

This chapter has outlined the physical architecture of an actuator made of many discrete units, and the specific design of a shape memory alloy actuator unit that will serve as the basis of all further experimentation. Both the serial and parallel actuator configurations described here produce lumped, quasi-passive system models whose parameters can be expressed as linear functions of the number of units in each state. One configuration parameter that was not discussed in this section was the possibility of a variable pennation angle, that is, the angle made between the direction of output motion and the alignment of the actuator units. Exploration of this would

be interesting future work.

Many improvements could be made to the fundamental SMA actuator unit used as a building block in this thesis. One easy improvement that can be made is to modulate the current, rather than simply choosing a constant driving voltage. A simple pulse width modulation scheme could be used to increase the current when the actuator transitions from the *off* state to the *on* state, so that the phase transition is sped up. This has been shown to work in previous, related studies [9]. It would be interesting to attempt some kind of closed-loop compensation based on the thermistor effect, as Teh and Featherstone demonstrated in their recent work [62]. This was not attempted because the helical SMA wire might pose some difficulty. The spring contacting itself may cause some change in the element's resistance. Still, it remains an interesting future avenue of inquiry.

Chapter 3

Modeling Stochastic Recruitment as a Random Process

3.1 Introduction

The previous chapter outlined a physical model of the actuator units, assuming that they were at steady state. This chapter develops a dynamic model of the actuator as it responds to state transition commands. The key to developing control laws for this system is in developing a good model for describing how many individual random state transitions impact the overall dynamics of an ensemble of units.

The first model developed in this chapter describes the state transition behavior of a single unit, which is modeled as a random finite-state agent. Conditions are outlined defining when this model adequately describes the output of each unit, based on the transient dynamics of the underlying active material. This model is then used to build a model of the dynamics of the whole ensemble of actuator units, based on a state space consisting of the number of units in each state. Both the unit model and the ensemble model are generalized to m states.

In addition to exact process models, two approximate models are introduced. The first of these is a m-order Markov chain model whose state is the probability distribution on the state of any randomly selected agent. This model is useful for describing the open-loop response of the actuator to a constant state transition graph. The

properties of the open-loop response are discussed mathematically and experimentally. The second approximate model keeps track of the expectation and covariance of the number of units in each state, rather than the whole distribution. This model can be used to express the state transition dynamics as a linear state-space system with additive random noise.

The specifics of generating uncorrelated, random state transitions in software or hardware will not be discussed in this chapter; all of the results from this thesis were generated using prepackaged software pseudo-random number generators, which were more than adequate for guaranteeing the required stochastic behavior of each unit. Unlike more advanced problems of random number generation requiring cryptographic security, the requirements placed on the behavior of each unit are mild, as only uncorrelated, random state transitions are needed. For those interested in more specifics, Appendix A explores how minimalistic random, uncorrelated state machines can be implemented using a small number of digital logic gates.

3.2 Modeling Stochastic Recruitment as a Random Process

3.2.1 A Single Unit Model

A recruitment-based active material actuator is made up of many small units, each controlled by a two-state machine. If the state transitions of this machine are intentionally randomized, then the behavior of each unit can be expressed as a random Markov process, having a state vector $s(t) \in \{off, on\}$. The random transitions between the *on* and *off* states are determined by two parameters: $p(t)$, the probability of transitioning from *off* to *on* at time t, and $q(t)$, the probability of transitioning from *on* to *off*.

In general, the dynamics of units having m states could be considered, having a state $s(t) \in \{1 \ldots m\}$, and a state transition graph $\mathbf{A}(t)$, as shown in Fig. 3-1. This time-varying state transition matrix $\mathbf{A}(t)$ can be thought of as the conditional

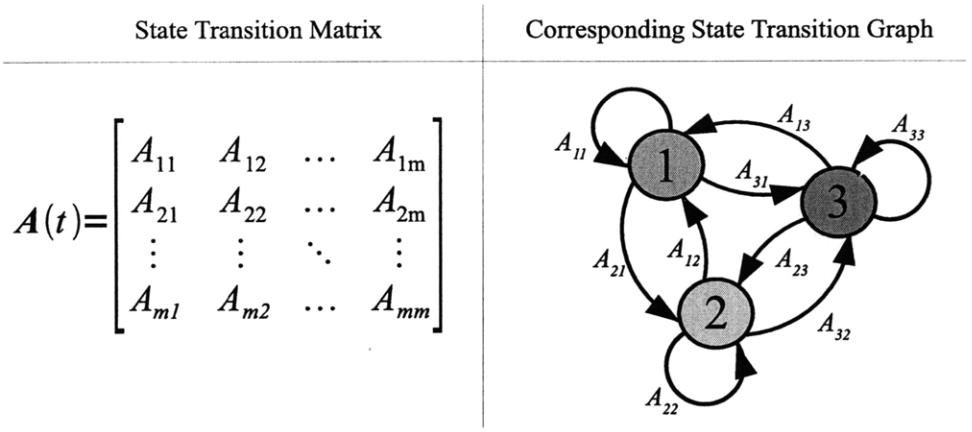

Figure 3-1: An illustration of the state transition matrix $\mathbf{A}(t)$.

probability of the agent's future state as a function of the present state,

$$A_{ij}(t) = \Pr\{s(t+1) = i |\ s(t) = j\} \tag{3.1}$$

The steady-state output of a single unit is determined by its finite state machine, but the transient behavior is based on the continuous-state reponse of the physical system, as illustrated in Fig. 3-2. These dynamics are assumed to be stable and repeatable, so that the discrete state always determines the output of the unit. In the case of the recruitment-based SMA actuator, the physical state of the actuator is a sharply nonlinear function of temperature, as discussed in §2.2.1. Consequently, constant-power resistive heating will produce the same output very reliably, as long as the equilibrium temperature of each SMA element is significantly higher or lower than the two-phase temperature region. The units' stable output behavior could also be a result of some local closed-loop regulator, as it was in the case of the work of Cho *et al.* on SMA actuator arrays driven by thermoelectric heat pumps [56], or Teh and Featherstone's minor loop force control [62].

The time scale on which the output of an actuator unit is determined by discrete state $s(t)$ is illustrated in Fig. 3-3. A two-state machine which delivers different state-dependent inputs (such as current) to the active material, so that within some time ΔT_s, the physical output of the unit $y(t)$ converges to the steady-state values, shown in the figure as h_{on} and h_{off}, as if these steady-state values were references fed

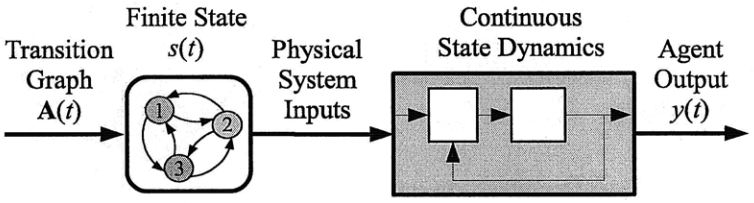

Figure 3-2: A hybridized continuous system, with some discrete state agent determining the inputs to the continuous (physical) dynamics of unit.

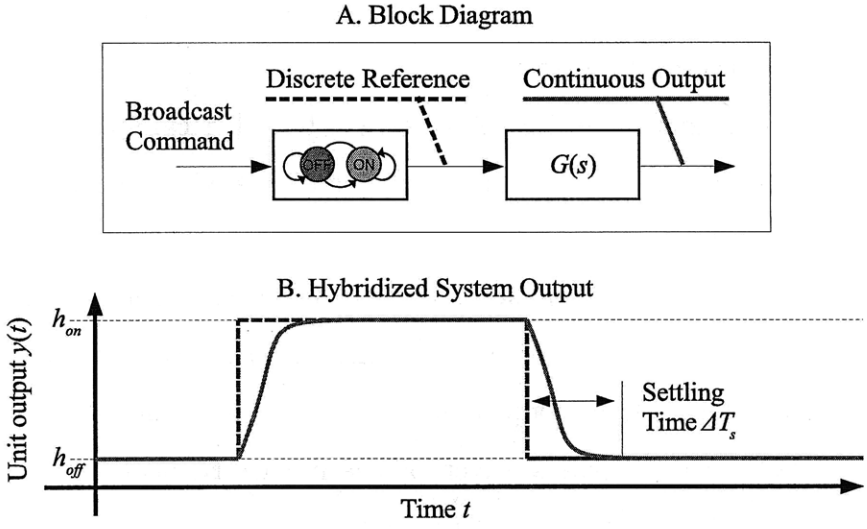

Figure 3-3: An illustration of how a simple system could be hybridized. If the time scale of interest is longer than ΔT_s, the output is determined by the discrete dynamics.

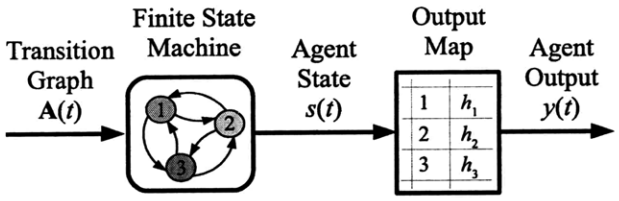

Figure 3-4: The discrete system approximation.

to a feedback loop. The output $y(t)$ can be treated as a static function of the discrete state $s(t)$, as shown in Fig. 3-4, provided that the time delay between discrete time intervals is greater than the continuous settling time ΔT_s. The output of a each unit can also be a vector $\underline{y}(t)$, as it would be in the case of many parallel actuators whose force and stiffness are both considered:

$$\underline{h}_{on} = \begin{bmatrix} f_{on} \\ k_{on} \end{bmatrix} \tag{3.2}$$

$$\underline{h}_{off} = \begin{bmatrix} f_{off} \\ k_{off} \end{bmatrix}$$

The time evolution of the finite state $s(t)$ can be described using some vector $\underline{x}(t) = [\ x_1(t) \quad x_2(t) \quad \ldots \quad x_m(t)\ ]^T$, an m-dimensional vector representing the probability that the agent is in any particular state,

$$x_i(t) = \Pr\{s(t) = i\} \tag{3.3}$$

Using (3.1) the time evolution of each individual unit could be described in terms of $\underline{x}(t)$ and $\mathbf{A}(t)$,

$$\underline{x}(t+1) = \mathbf{A}(t)\underline{x}(t) \tag{3.4}$$

Recall that in the case of the actuator units, having states *off* and *on*, the state

transition matrix is determined by two parameters, $p(t)$ and $q(t)$,

$$\mathbf{A}(t) = \begin{bmatrix} 1-q(t) & p(t) \\ q(t) & 1-p(t) \end{bmatrix} \tag{3.5}$$

The probability distribution over the state of a single agent has one free parameter, which we will write as $x_{on}(t)$,

$$\underline{x}(t) = \begin{bmatrix} x_{on}(t) \\ x_{off}(t) \end{bmatrix} = \begin{bmatrix} x_{on}(t) \\ 1-x_{on}(t) \end{bmatrix} \tag{3.6}$$

This simple, one-agent random process model forms the basis of a model for the dynamic behavior of the whole ensemble.

3.2.2 An Ensemble of Many Agents

Now, consider the case of N^{Total} identical units, all making discrete state transitions based on the same broadcast state transition matrix $\mathbf{A}(t)$. It is impractical to estimate the full state of the system in terms of some "master" $N^{Total} \times 1$ discrete state distribution vector $\underline{S}(t) = [s_1(t)\ s_2(t)\ ...\ s_{N^{Total}}(t)]^T$. This approach scales poorly, and also makes little sense if the units are effectively anonymous. Instead, it would be more reasonable to represent knowledge of the system states in terms of a $m \times 1$ vector $\underline{N}(t)$, corresponding to the number of agents in each discrete state as illustrated as a histogram in in Fig. 3-5. This is often called a population model, because it closely resembles models for populations of herd animals, insects or fish [2], [61]. Because the number of agents is conserved, the state distribution $\underline{N}(t)$ for the two-state model has one free parameter, $N_{on}(t)$, as $\underline{x}(t)$ did in (3.6),

$$\underline{N}(t) = \begin{bmatrix} N_{on}(t) \\ N_{off}(t) \end{bmatrix} = \begin{bmatrix} N_{on}(t) \\ N^{Total} - N_{on}(t) \end{bmatrix} \tag{3.7}$$

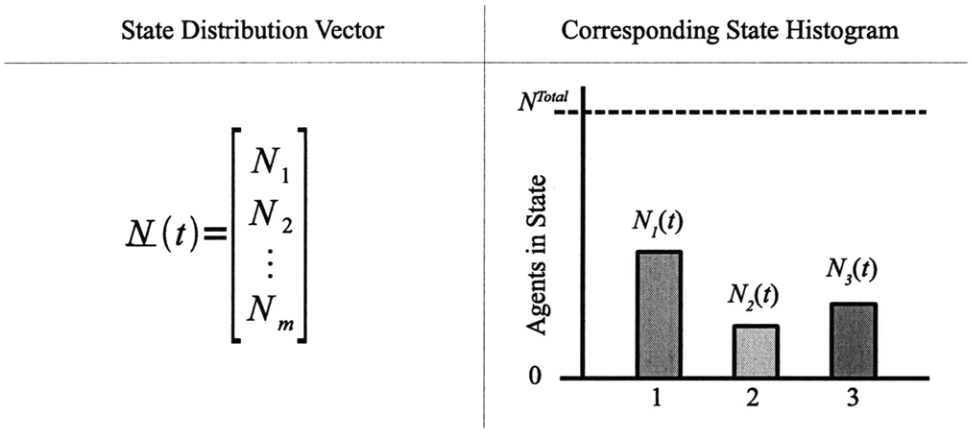

Figure 3-5: An illustration of the state distribution vector.

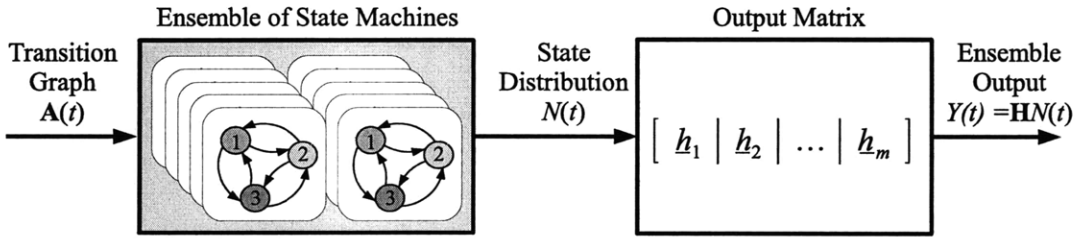

Figure 3-6: An illustration of the output model for a large number of units.

3.2.3 Ensemble Output

It was shown previously in §2.3 that the output characteristics of the actuator, such as the equilibrium position $\Delta(t)$ and the stiffness $K(t)$, can be defined as linear functions of the number of units in each discrete state for some geometric configurations. These outputs could be represented as a vector $\underline{Y}(t)$,

$$\underline{Y}(t) = \sum_{i=1}^{N^{Total}} \underline{y}_i(t) \tag{3.8}$$

Because the output of each unit is determined by the state, $\underline{Y}(t)$ can be written as a weighted sum of the elements of $\underline{N}(t)$, or as $\underline{N}(t)$ multiplied by an output matrix $\mathbf{H}$,

$$\underline{Y}(t) = \sum_{i=1}^{N^{Total}} \underline{h}_{s_i(t)} N_i(t) = \underline{h}_1 N_1(t) + \underline{h}_2 N_2(t) + ... \underline{h}_m N_m(t) = \mathbf{H}\underline{N}(t) \tag{3.9}$$

One thing worth considering here is that the space of possible state distributions operates under constraints that provide information about the system's state. This is why the two-state system was described in (3.7) by one free parameter, $N_{on}(t)$. In general, conservation implies that a constraint exists in the form of a pseudo-output for the system,

$$\begin{aligned} N^{Total} &= N_1(t) + N_2(t) + ... + N_m(t) \\ &= \begin{bmatrix} 1 & 1 & ... & 1 \end{bmatrix} \underline{N}(t) \end{aligned} \tag{3.10}$$

This pseudo-output can be appended onto the output matrix, and it is principally useful for increasing the rank of $\mathbf{H}$. For example, the force of an actuator may be f_{off} in the *off* state, and f_{on} in the *on* state. This single output is not enough to uniquely specify the state based on a force measurement. However, when the conservation constraint is added, $\mathbf{H}$ has full rank and could be used to produce an

algebraic estimate $\underline{\hat{N}}(t)$ from the output $\underline{Y}(t)$,

$$\mathbf{H} = \begin{bmatrix} f_{on} & f_{off} \\ 1 & 1 \end{bmatrix} \tag{3.11}$$

$$\underline{\hat{N}}(t) = \begin{bmatrix} f_{on} & f_{off} \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \sum f(t) \\ N^{Total} \end{bmatrix}$$

$$\hat{N}_{on}(t) = \frac{\sum f(t) - N^{Total} f_{on}}{f_{on} - f_{off}}$$

3.2.4 Ensemble Time Evolution

The time evolution of the state distribution vector $\underline{N}(t)$ as a function of $\mathbf{A}(t)$ is far more complicated than the Markov model derived for the single unit. It is best understood by introducing an intermediate random variable $T_{ij}(t)$ equal to the number of units transitioning from state j to state i at time t. The lattice diagram in Fig. 3-7 illustrates the relationship between these intermediate variables and the components of $\underline{N}(t+1)$. The probability of any number of units transitioning in this manner is dependent on the number of units $N_j(t)$ in the originating state j. Because it is the summation of many binary random decisions, the probability distribution over the number of transitioning agents is binomial,

$$\Pr\{T_{ij}(t) = k |\ N_j(t), A_{ij}(t)\} = \begin{pmatrix} N_j \\ k \end{pmatrix} A_{ij}(t)^k (1 - A_{ij}(t))^{N_j - k} \tag{3.12}$$

The future value of N_i is equal to a sum of the number of agents departing for state i from all states, as illustrated in Fig. 3-7,

$$N_i(t+1) = \sum_{j=1}^{m} T_{ij}(t) \tag{3.13}$$

This, in turn, implies that the probability distribution of $N_i(t+1)$ conditioned on the present state $\underline{N}(t)$ and the transition matrix $\mathbf{A}(t)$ is the convolution of the distribution

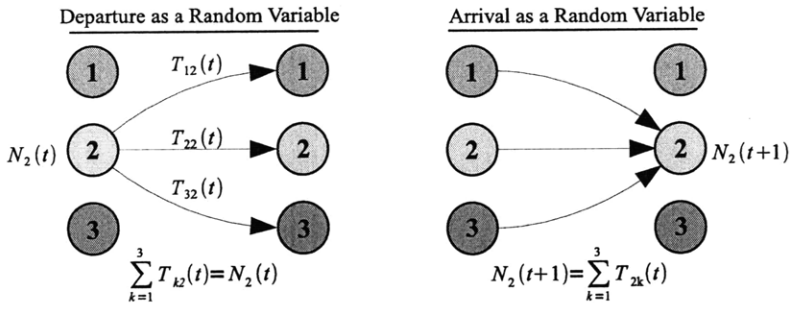

Figure 3-7: A lattice diagram illustrating the relationship between the number of agents departing a single state and the number of agents arriving in a single state.

over the independent transitions from the agents in each state,

$$\begin{aligned}\Pr\{N_i(t+1) = k|\ \underline{N}(t), \mathbf{A}(t)\} = &\Pr\{T_{i1}(t) = 1...N_1(t)|\ N_1(t), A_{i1}(t)\} \\ &\otimes \Pr\{T_{i2}(t) = 1...N_2(t)|\ N_2(t), A_{i2}(t)\} \\ &\otimes ... \\ &\otimes \Pr\{T_{im}(t) = 1...N_m(t)|\ N_m(t), A_{im}(t)\}\end{aligned} \tag{3.14}$$

For a two state system with states *on* and *off*, this convolution is easy to compute because there are only two independent transitions that can be made, $T_{off \to on}(t)$ and $T_{on \to off}(t)$. The probability of some number of *on* agents at time $t+1$ can be written as a function of the state transition parameters $p(t)$ and $q(t)$,

$$\Pr\{T_{off \to on}(t) = a|\ N_{off}(t), p(t)\} = \begin{pmatrix} N_{off}(t) \\ a \end{pmatrix} p(t)^a (1-p(t))^{N_{off}(t)-a} \tag{3.15}$$

$$\Pr\{T_{on \to off}(t) = b|\ N_{on}(t), q(t)\} = \begin{pmatrix} N_{on}(t) \\ b \end{pmatrix} q(t)^b (1-q(t))^{N_{on}(t)-b}$$

Because the number of agents transitioning into the *on* state is the difference of these two random variables, the probability distribution over a future state conditioned on the present state is just the convolution of these two distributions. In the two-state case, this is simple enough; however, it becomes impractical very quickly to

calculate the probability of arriving in some state for a system with many states. Calculating the joint probability function for the number of agents arriving in two different states i and j involves the computation of joint or conditional probabilities for the intermediate variables $T_{ik}(t)$ and $T_{jk}(t)$, and convolution over the joint event space:

$$\Pr\{T_{ik}(t) = a, T_{jk}(t) = b |\ N_k(t), A_{ik}(t), A_{jk}(t)\} = \tag{3.16}$$
$$\begin{pmatrix} N_k(t) \\ a \quad b \end{pmatrix} A_{ik}(t)^a A_{jk}(t)^b (1 - A_{ik}(t) - A_{jk}(t))^{N_k - a - b}$$

Alternatively, the joint probability of arrival in some state could be computed using the conditional probabilities of departure based on the fixed outcome of one state transition,

$$\Pr\{T_{ik}(t) = a |\ T_{jk}(t) = b, N_k(t), A_{ik}(t), A_{jk}(t)\} = \tag{3.17}$$
$$\begin{pmatrix} N_k(t) - b \\ a \end{pmatrix} \left(\frac{A_{ik}(t)}{1 - A_{jk}(t)}\right)^a \left(\frac{1 - A_{ik}(t) - A_{jk}(t)}{1 - A_{jk}(t)}\right)^{N_k - a - b}$$

This can be generalized to computing the joint or conditional probabilities for all transitions as a multinomial random variable. Unfortunately, it is apparent that calculation of a probabilistic time evolution model based on this complete state representation quickly becomes difficult as the number of agents and the number of states increases, as it requires a sequence of convolutions be performed for every possible future state.

3.2.5 In Summary

A random process model for the time evolution of both a single random agent and an ensemble of random agents has been derived. However, the explicit calculation of the probability distribution for a future state conditioned on the present state is complex and involves performing many successive convolutions. The remainder of

the chapter focuses on two simpler models of the system behavior that shed more light on the properties of its dynamic response. First, a m-order Markov chain model will be introduced in §3.3. This model is useful for predicting the open-loop behavior of the system, and for relating the control problems we are interested in to control problems in the greater community of distributed control researchers. The second model, covered in §3.4, is a two-moment approximation of the ensemble random process model. It is far more computationally efficient than the explicit calculation of probability distributions for $\underline{N}(t)$, and it is capable of modeling the propagation of an estimated state trajectory for the model, as well as the estimation uncertainty.

3.3 A m-Order Markov Chain Model

Rather than calculating the probability distribution over all possible state distributions $\underline{N}(t+1)$, a slightly coarser but much simpler model for the ensemble behavior is to consider the probability distribution over each state for a single unit, $\underline{x}(t)$, as representing the whole ensemble. Knowledge of $\underline{x}(t)$ does not guarantee that the number of agents in some state $N_i(t)$ can be predicted exactly. Instead, it can only be predicted as a Bernoulli random variable whose probability distribution is a function of $\underline{x}(t)$,

$$\Pr\{N_i(t) = k | \, \underline{x}(t)\} = \left(\begin{array}{c} N^{Total} \\ k \end{array} \right) x_i(t)^k (1 - x_j(t))^{N^{Total}-k} \tag{3.18}$$

Correspondingly, one could estimate a maximum *a posteriori* likelihood value for $\underline{x}(t)$ given $\underline{N}(t)$ using the probability distribution in (3.18). This estimate, $\hat{\underline{x}}(t)$, is equal to the number of agents in each state divided by the total number of agents,

$$\hat{\underline{x}}(t) = \underline{N}(t)/N^{Total} \tag{3.19}$$

The expected value of $\underline{N}(t)$ can be expressed as a function of $\underline{x}(t)$,

$$\mathrm{E}\{\underline{N}(t) | \, \underline{x}(t)\} = N^{Total}\underline{x}(t) \tag{3.20}$$

The covariance of $\underline{N}(t)$ can also be computed,

$$\mathrm{Cov}\{N_i(t), N_j(t)|\ \underline{x}(t)\} = N^{Total} x_i(t)(\delta_{ij} - x_j(t)) \tag{3.21}$$

Here δ_{ij} is the Kronecker delta function. There are several reasons why using $\underline{x}(t)$ as a state variable for describing system behavior makes more sense than using $\underline{N}(t)$, if little feedback information is available. First, it is easy to predict the future behavior of $\underline{x}(t)$ given an initial condition and a broadcast command, using (3.4). In the absence of other information, such as a measured ensemble output, this will provide a good guess of long-term behavior. Second, as N^{Total} becomes large, the law of large numbers will guarantee that the true value of $\underline{N}(t)$ will approach its expected value. This can be shown by computing the covariance of $\underline{N}(t)$ divided by the total number of units N^{Total}, in a manner similar to (3.21),

$$\mathrm{Cov}\left\{ \frac{N_i(t)}{N^{Total}}, \frac{N_j(t)}{N^{Total}} \middle|\ \underline{x}(t) \right\} = \frac{x_i(t)}{N^{Total}}(\delta_{ij} - x_j(t)) \tag{3.22}$$

As N^{Total} becomes large, these terms all approach zero as $1/N^{Total}$. The behavior of the system starting from some initial probability distribution $\underline{x}(0)$ can be found by applying (3.4) recursively,

$$\underline{x}(t) = \left(\prod_{k=0}^{t-1} \mathbf{A}(k) \right) \underline{x}(0) \tag{3.23}$$

It is worthwhile to note that this model, which relies only on an initial estimate of $\underline{x}(t)$, will have bounded covariance at any point in time. Of course, estimation of any bounded random process should be possible with bounded covariance. After all, there is a clear maximum and minimum number of agents that can be in any state. In this case, though, the interesting thing is that the value of the estimate does not approach any kind of entropy limit as time passes. If, for example, the estimate of $\underline{N}(t)$ approached a uniform distribution instead of a multinomial distribution, then the boundedness of the covariance would not be remarkable. Instead, the *a priori* prediction of $\underline{N}(t)$ based on some initial value $\underline{x}(0)$ will remain more or less equally

valid throughout time.

3.3.1 Behavior given a constant state transition graph

One special case of particular interest is the case when the broadcast state transition matrix is held at some fixed value $\mathbf{A}$. If this is true, then (3.23) reduces to a simpler expression,

$$\underline{x}(t) = \mathbf{A}^t \underline{x}(0) \tag{3.24}$$

Naturally, this implies that the steady state behavior of $\underline{x}(t)$ will be determined by the eigenvalues of $\mathbf{A}$. According to the Perron-Frobenius theorem, if $\mathbf{A}$ represents a conservative Markov chain, the eigenvalues are very constrained [15]. The largest eigenvalue of $\mathbf{A}$, λ_1, will be equal to 1, and it will correspond to a unique eigenvector $\underline{u}_1$. This stationary distribution describes the steady-state behavior of the ensemble output. From (3.20), we know that the expected state distribution given $\underline{x}(t)$ could be calculated for this stationary distribution,

$$\mathrm{E}\{\underline{N}(ss) |\, \underline{x}(ss) = \underline{u}_1\} = N^{Total}\underline{u}_1 \tag{3.25}$$

Based on the matrix output model from (3.9), the expected steady state output can be found from this,

$$\mathrm{E}\{\underline{Y}(ss) |\, \underline{x}(ss) = \underline{u}_1\} = \mathbf{H}N^{Total}\underline{u}_1 \tag{3.26}$$

Recall that the two-state transition graph to be broadcast to the actuator units has two free parameters, p and q, corresponding to the constant probability per time step of transitioning from *off* to *on* and from *on* to *off*, respectively. The largest

eigenvalue of this matrix can be computed:

$$\mathbf{A}(t) = \begin{bmatrix} 1-q & p \\ q & 1-p \end{bmatrix} \tag{3.27}$$

$$\underline{u}_1 = \begin{bmatrix} \frac{p}{p+q} \\ \frac{q}{p+q} \end{bmatrix} \tag{3.28}$$

3.3.2 Rate of convergence to steady state

Eigenvalue decomposition can also be used to describe the transient behavior of $\underline{x}(t)$. The time evolution model from (3.24) could be rewritten in terms of the eigenvalues $\lambda_1 ... \lambda_m$ and the corresponding eigenvectors, $\underline{u}_1 ... \underline{u}_m$:

$$\underline{x}(t) = \sum_{i=1}^{m} \lambda_i^t \underline{u}_i \left(\frac{\underline{u}_i^T \underline{x}(0)}{|\underline{u}_i|} \right) \tag{3.29}$$

The steady-state behavior of the system will be dominated by the eigenvector $\underline{u}_1$ corresponding to the largest eigenvalue $\lambda_1 = 1$. The time it takes to approach this steady-state behavior is dominated by the second-largest eigenvalue of the constant state transition matrix $\mathbf{A}$, λ_2. For the two-state case, the second-largest eigenvalue λ_2 is:

$$\lambda_2 = 1 - p - q \tag{3.30}$$

In the general m-state case, it is difficult to say anything about the value of λ_2 without directly computing it. The state transition matrix $\mathbf{A}$ is not symmetric, nor is it positive, properties which would greatly aid in the construction of bounding cases for eigenvalues.

3.3.3 Covariance in transient and steady state cases

Using (3.21), the covariance of $\underline{N}(t)$ can be calculated at any point in time. At steady state, this covariance depends only on the first eigenvector of $\mathbf{A}$. The value of the other eigenvectors are still important; they determine how the magnitude of the

correlation between the present and past values of $\underline{N}(t)$ conditioned on $\underline{x}(t)$.

$$\mathrm{Cov}\{N_i(t), N_j(t+1)|\, \underline{x}(t), \mathbf{A}\} = N^{Total} x_i(t) \left(A_{ii}\delta_{ij} - \sum_{k=1}^{m} A_{jk} x_k(t) \right) \tag{3.31}$$

The summation over k can be recognized as the time evolution of $\underline{x}(t)$,

$$\mathrm{Cov}\{N_i(t), N_j(t+1)|\, \underline{x}(t), \mathbf{A}\} = N^{Total} x_i(t) \left(A_{ii}\delta_{ij} - x_j(t+1) \right) \tag{3.32}$$

In the steady state case, $\underline{x}(t+1) = \underline{x}(t) = \underline{u}_1$,

$$\mathrm{Cov}\{N_i(t), N_j(t+1)|\, \underline{x}(t) = \underline{u}_1, \mathbf{A}\} = N^{Total} u_{1i}(t) \left(A_{ii}\delta_{ij} - u_{1j} \right) \tag{3.33}$$

The diagonal terms of this autocovariance matrix are dependent on the magnitude of the diagonal transition probabilities. For the two-state system, the covariance at the present time and the one-step-ahead autocovariance are as follows:

$$\mathrm{Cov}\{\underline{N}(t)|\, \underline{x}(t)\} = N^{Total} x_{on}(t)(1 - x_{on}(t)) \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{3.34}$$

$$\mathrm{Cov}\{\underline{N}(t), \underline{N}(t+1)|\, \underline{x}(t), \mathbf{A}(t)\} = \tag{3.35}$$

$$N^{Total} x_{on}(t)(1 - x_{on}(t)) \begin{bmatrix} 1-p-q & p-1-\frac{q}{x_{off}(t)} \\ q-1-\frac{p}{x_{on}(t)} & 1-p-q \end{bmatrix} \tag{3.36}$$

Notice that the diagonal elements of the one-step-ahead autocovariance matrix contain factors of $\lambda_2 = 1 - p - q$. The role that the second-largest eigenvalue has in determining the autocovariance of the two-state system is apparent here. A state transition matrix with a fast convergence rate $\lambda_2 \approx 0$ will cause the state distribution to exhibit relatively little autocovariance. A slow convergence rate ($|\lambda_2| > 0$) will yield higher autocovariance.

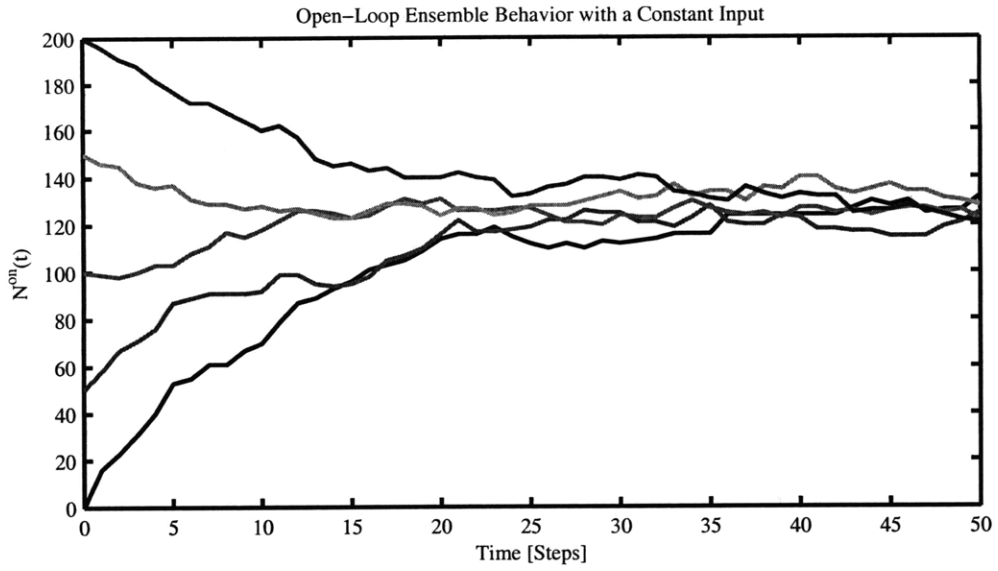

Figure 3-8: A simulation of 5 trajectories having different initial conditions and the same value of x_{ss} all converge to the same distribution.

3.3.4 Rate of state transitions at steady state

Another possibly important performance metric which the Markov chain model can be used to predict is the number of state transitions which occur when the system is at steady state. The number of units which do not transition at any point in time can be found by looking at the transition probabilities along the diagonal of $\mathbf{A}$,

$$\mathrm{E}\{N^{Trans}(ss)|\ \underline{x}(ss) = \underline{u}_1\} = N^{Total}\sum_{i=1}^{m}(1 - A_{ii})\underline{u}_i \tag{3.37}$$

Notice that, like the steady-state autocovariance in (3.33), the expected number of transitions per unit time at steady state is a function of the steady state probability vector $\underline{u}_1$ and the diagonal elements of $\mathbf{A}$. In the two-state case, the rate of transitions can be found as a function of p and q,

$$\mathrm{E}\{N^{Trans}(ss)|\ p, q\} = \frac{2N^{Total}pq}{p+q} \tag{3.38}$$

3.3.5 Application: open-loop input planning for control

Finite-state Markov chain models have been used commonly to describe the behavior of multi-agent systems, including swarms of robots [11], [33], [32], [40], [43], [60], animals, [2], insects [5], robots interacting with insects, [19], and bacteria [26], [27]. They are also in use among these communities for the open-loop synthesis of systems that exibit a desired collective behavior [18], [8], [59]. A similar analysis and control system synthesis can be performed in the context of recruitment. It is illuminating because it highlights some of the advantages and drawbacks of this general approach to distributed control, as well as to the system at hand. Suppose that a controller having no feedback information is left with the task of recruiting an ensemble of 2-state *on*/*off* agents into a desired distribution $\underline{N}^{ref}$. Equation (3.28) demonstrated that the state transition dynamics have a steady-state component in the response, so p and q could be chosen so that some desired number or fraction is expected according to (3.18),

$$\begin{bmatrix} \frac{p}{p+q} \\ \frac{q}{p+q} \end{bmatrix} = \begin{bmatrix} \frac{N_{on}^{ref}}{N^{Total}} \\ \frac{N_{off}^{ref}}{N^{Total}} \end{bmatrix} \tag{3.39}$$

Unlike most feed-forward control problems, where initial conditions must be known for controlling the system, this broadcast control will guarantee that the expected number of agents will be *on* or *off*, irrespective of the initial condition $\underline{x}_0$. Figure 3-8 illustrates this, showing an ensemble of 200 units, with a desired state distribution of 125 *on* agents, 75 *off* agents. However, (3.39) does not completely specify p and q. For example, setting $p = 0.125$ and $q = 0.075$ will drive expected fraction of recruited agents to the desired distribution. The control policy $p = 0.250$ and $q = 0.15$ will also accomplish this. Performance factors other than expected value, such as convergence time, error variance, and steady-state energy costs may be of interest, and the multiple policies satisfying (3.39) can be further analyzed to find the policy with the most applicable performance trade-offs. The following section presents an analysis of these different performance factors are affected by variations in policy parameters, based on the analysis from §3.3.1-§3.3.4.

3.3.6 Performance Trade-offs in open-loop response

Convergence rate and steady state distribution are independent. In order to show this, the state transition graph parameters can be rewritten under a change of variables. It then becomes clear that there is a factor which impacts convergence rate but not steady state distribution. The control policy parameters p and q can be rewritten in terms of three variables, β, p_0 and q_0,

$$\begin{aligned} p &= \beta p_0 \\ q &= \beta q_0 \\ p_0 + q_0 &= 1 \end{aligned} \tag{3.40}$$

When the transition probabilities are scaled in this way, the steady-state distribution $\underline{x}_{ss}$ is independent of β,

$$\underline{x}_{ss} = \underline{u}_1 = \begin{bmatrix} \frac{\beta p_0}{\beta(p_0+q_0)} \\ \frac{\beta q_0}{\beta(p_0+q_0)} \end{bmatrix} = \begin{bmatrix} p_0 \\ q_0 \end{bmatrix} \tag{3.41}$$

However, (3.30) shows that the rate of convergence λ_2 still depends on β,

$$1 - p - q = 1 - \beta(p_0 + q_0) = 1 - \beta \tag{3.42}$$

This means that β is a free parameter with which the convergence time can be arbitrarily varied while still satisfying the steady-state condition imposed in (3.39). In the most extreme case, β is chosen to be 1, so that $\lambda_2 = 0$. In this case, $\underline{x}(t)$ converges to $\underline{u}_1$ after only one time interval, according to (3.24). Figure 3-9 shows $\underline{x}(t)$ converging to steady-state. Notice that for several values of λ_2, the same steady-state behavior is produced by each policy.

Convergence rate and variance are independent. Open-loop stochastic recruitment guarantees only that the expected number of *on* agents in steady state $N_{on}(ss)$ will be equal to N_{on}^{ref}. There will be some error at steady-state, which can be

characterized by the variance of N_{ss}^{on}. One striking and perhaps counter-intuitive feature of these feed-forward control policies is that this variance will not depend on the rate of convergence. This can be proven by examining the variance of $N_{on}(ss)/N^{Total}$. As previously discussed, the predicted value of $\underline{N}(t)$ as a function of $\underline{x}(t)$ is a Bernoulli random variable, so the variance at steady state normalized by the total number of agents is given by

$$Var\left\{\frac{N_{on}(ss)}{N^{Total}}\middle|\ x_1 = u_1\right\} = u_1(1-u_1)\frac{1}{N^{Total}} = \frac{pq}{(p+q)^2 N^{Total}} \tag{3.43}$$

The β scaling argument from (3.41) and (3.42) can be applied to the variance calculation. The numerator and denominator of (3.43) both vary by a factor of β^2, so the variance is independent of the convergence rate, $\lambda_2 = 1-\beta$ at which the aggregate output converges to its steady state probability distribution,

$$\frac{\beta^2 p_0 q_0}{\beta^2 (p_0+q_0)^2 N^{Total}} = \frac{p_0 q_0}{N^{Total}} \tag{3.44}$$

This is an important observation; it means that nothing is to be gained by taking "baby steps", that is, selecting very small values of p and q in hopes of improving the accuracy of recruitment in exchange for a slower rate of response. For example, Figure 3-9 shows three policies having values of $\lambda_2 = 0.0, 0.2, 0.5$. The probability distribution of $N_{on}(t)$ given $\underline{x}(t)$, superimposed vertically on the plot, converges to the same distribution at the steady-state. This result also implies that the only way to improve the accuracy of this control system for any value of $\underline{N}^{ref}$ is to increase the number of agents, N^{Total}. Despite this, it is worthwhile to note that there is an important relationship between the autocovariance of $\underline{N}(t)$ and λ_2, as (3.35) demonstrated. The practical implication of this in the context of control is in the power spectrum of the error. A feed-forward broadcast command which causes $\underline{x}(t)$ to converge slowly toward $\underline{u}_1$ will have a very broad autocorrelation function, and consequently a very narrow frequency distribution. In contrast, feed-forward broadcast command which causes $\underline{x}(t)$ to settle rapidly will have a very narrow auto-correlation function, and much more high frequency content in the error power spectrum.

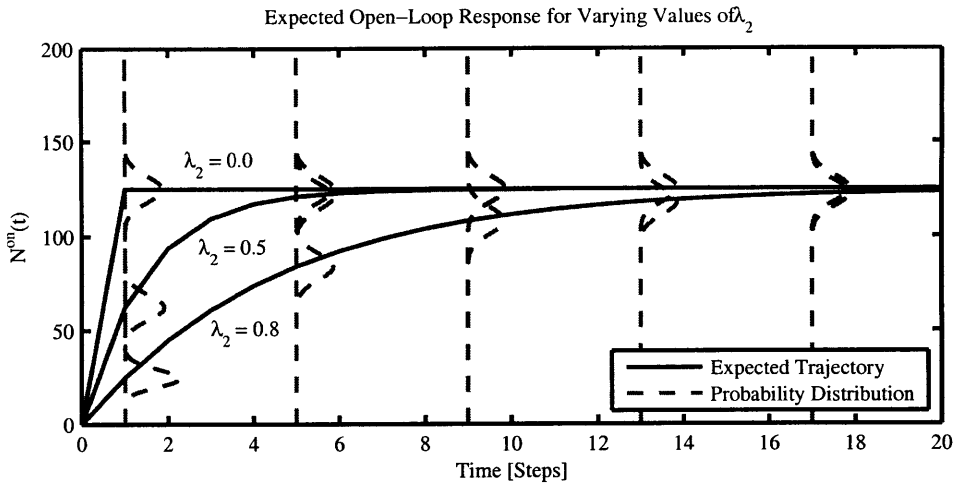

Figure 3-9: A graph showing the minimal impact of convergence time on the steady-state variance

The steady state transition rate depends on convergence rate. In a physical system, there is often a significant energy cost associated with switching agents from one state to another. For example, a mobile robot switching between patrolling two different areas will expend energy in driving from place to place. A shape memory alloy actuator has significant latent heat associated with the phase transition used for actuation, so spurious phase transitions are costly. As a consequence, it may be useful to consider the expected number of state transitions per unit time when formulating a control policy. We can apply the scaling argument again to (3.37), $N^{Trans}(ss)$ can be rewritten in terms of βp_0 and βq_0.

$$\mathrm{E}\{N^{Trans}(ss)|\ \beta p_0, \beta q_0\} = \frac{\beta^2 2N^{Total} p_0 q_0}{\beta(p_0 + q_0)} = \beta(2N^{Total} p_0 q_0) \tag{3.45}$$

It is clear in this expression that the most energy consuming steady-state is the middle point, i.e. $p_0 = q_0 = 0.5$, and that an increase in β implies more expected transitions per unit time in the steady state.

In summary, the results of the scaling analysis, shown in Table 3.1, show that there is a clear trade-off between between the rate at which the system converges and the expected number of transitions at steady-state, but not in the overall output covariance.

Table 3.1: Scaling of Performance Measures Versus β

Performance Measure	Dependency on β	Goal
$E(N_{ss}^{on})$	none	N^{ref}
$Var(N_{ss}^{on})$	none	0
Convergence Rate λ	$1-\beta$	0
$E(N_{ss}^{trans})$	$\beta(2Np_0q_0)$	0

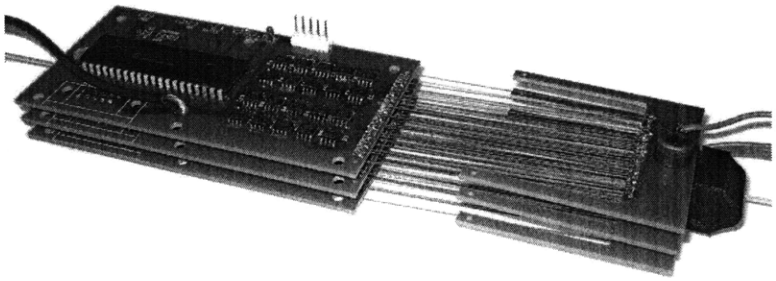

Figure 3-10: An actuator made from 60 parallel SMA springs.

3.3.7 Experimental example

In order to demonstrate the actual response of a recruitment system to an open-loop input, an actuator made of 60 SMA units placed in parallel was used to produce force at a constant displacement. Each unit was controlled with a small 2-state agent, which received a constant state transition graph from a central controller. The SMA spring in each unit was heated with Joule heating when the unit was in the *on* state. This actuator, shown in Fig. 3-10, produced a force ranging from 0-30 Newtons, depending on the number of activated elements. Several plots of step response are shown in Fig. 3-11, for inputs having values of λ_2 ranging between 0 and 0.8, for a steady state distribution of 60% *on* and 80% *on*, respectively. It is difficult from these plots to distinguish the impact of λ_2, probably because the lag due to the output behavior of the material is dominating the time constant of response, adding an additional dominant low-pass filtering characteristic. As λ_2 becomes large (for example, $\lambda_2 = 0.8$ is shown), the effect of this parameter can be noticed.

Figure 3-12 shows the steady-state response of the actuator to commands with

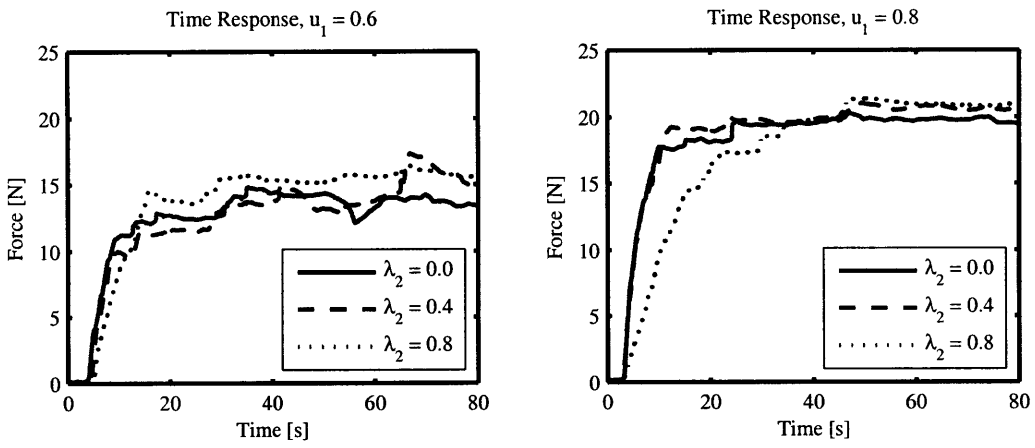

Figure 3-11: Open-loop response of the SMA actuator to constant feed-forward input commands.

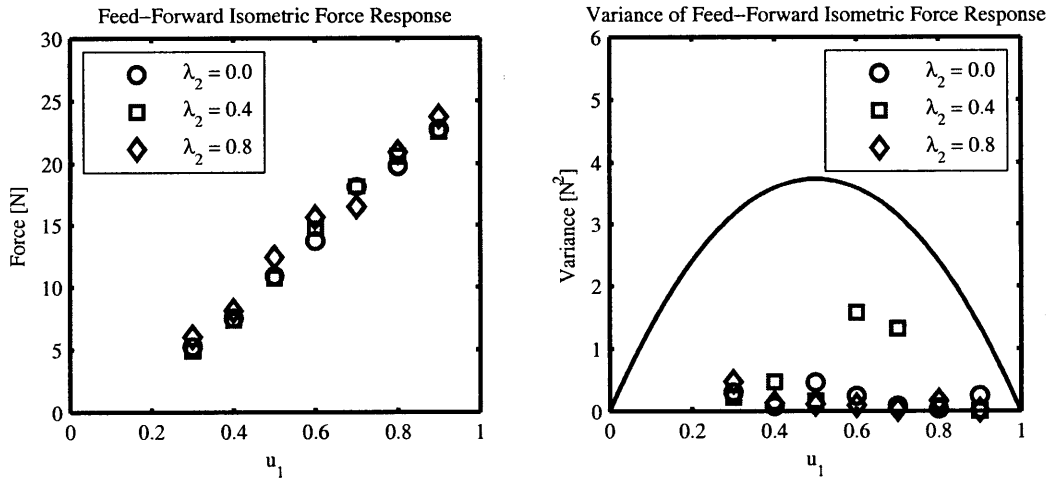

Figure 3-12: Average and variance of the steady-state response to open-loop input.

varying steady-state distributions and rates of convergence. The relationship between the output force and the steady state eigenvector is very linear. There was no discernable relationship between the rate of convergence and the steady-state behavior, as predicted. Based on this data, each actuator unit exerted approximately 0.5 N of force in the *on* state. The variance of the steady-state output is also plotted in Fig. 3-12, alongside the variance predicted by the model. The measured variance was found to be much smaller than the variance predicted by the model. There are two likely causes for this reduction in variance. First, the physical response time of the actuator was longer than than the transition time interval, set to 2 seconds. It took approximately 5 seconds for each unit to transition entirely from relaxed to contracted state, and the transitions were made every second. Consequently, the SMA activation dynamics acted as a sort of low-pass filter, smoothing the output and reducing the variance. The other likely cause was that the time window over which each output was averaged was roughly 80 seconds, which may not have been long enough to pick up on longer-term random trends. In order to see the full effect, one could increase the length of the time intervals, and take a much larger sample of data to avoid missing variance due to extremely low-frequency trends in the output.

3.3.8 In Summary

This section has shown that a Markov chain model based on the probability distribution of a single unit's state is useful for predicting the open-loop behavior of ensembles, and for the synthesis of open-loop inputs that drive the system toward a desired distribution, irrespective of the initial conditions of the ensemble. However, these open-loop inputs will not cause the system to settle with zero error, nor are they robust to disturbances or modeling errors. To account for these factors, closed-loop control is needed.

3.4 Moment-Based Models

In the previous section, the use of a single-unit probability distribution $\underline{x}(t)$ as a sufficient statistic for describing the ensemble was examined thoroughly. Another reasonable approximation of the ensemble behavior is to examine the expection and covariance of the state transition dynamics as a function of the present state distribution $\underline{N}(t)$. It is simpler to compute the expectation and covariance of the state transition behavior than it is to deal with the whole probability distribution, and it also has the advantage of utilizing the available information about the system state than the m-order model. This approximate model has the dual usefulness of being representable as a linear, time-varying model, shown in Fig 3-13, treating all of the state evolution uncertainty as an additive, independent random signal, under the right assumptions and bounding estimates. In future chapters, this moment-based process model will be used for synthesizing control laws, and for online estimation of the state distribution in full-state feedback control laws.

3.4.1 Expectation and Covariance of State Transitions

Going back to (3.12), the state transitions were modeled in terms of the random variables $T_{ij}(t)$, the number of agents transitioning from state j to state i at time t. The expectation of $T_{ij}(t)$ conditioned on $N_j(t)$ and $A_{ij}(t)$ was found to be the following:

$$\mathrm{E}\{T_{ij}(t)|\ N_j(t), A_{ij}(t)\} = A_{ij}(t)N_j(t) \tag{3.46}$$

Because the value of N_i at time $t+1$ is equal to the sum of all agents arriving in that state, the expected number of agents in state i can be found:

$$\mathrm{E}\{N_i(t+1)|\ \underline{N}(t), \mathbf{A}(t)\} = \sum_{k=1}^{m} A_{ik}N_k(t) \tag{3.47}$$

Written in vector form, this becomes a matrix multiplication,

$$\mathrm{E}\{\underline{N}(t+1)|\ \underline{N}(t), \mathbf{A}(t)\} = \mathbf{A}(t)\underline{N}(t) \tag{3.48}$$

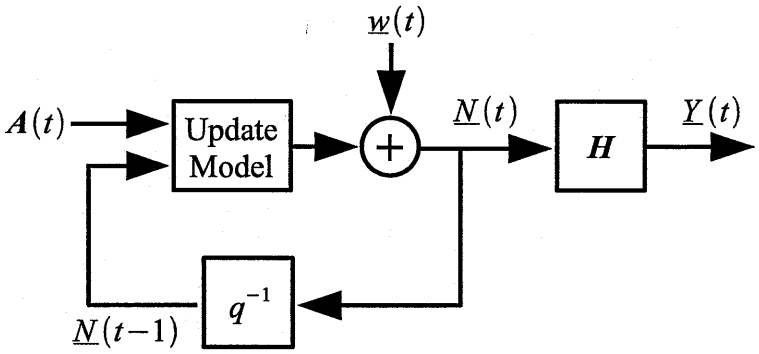

Figure 3-13: The moment-based model, illustrated as a block diagram

The expected future output $\underline{Y}(t+1)$ can be predicted in a similar fashion,

$$\mathrm{E}\{\underline{Y}(t+1)|\ \underline{N}(t), \mathbf{A}(t)\} = \mathbf{H}\mathbf{A}(t)\underline{N}(t) \tag{3.49}$$

3.4.2 State Transition Covariance

The covariance of the state transition variables $T_{ik}(t)$ and $T_{jk}(t)$ can be calculated using the multinomial distribution,

$$\mathrm{Cov}\{T_{ik}(t), T_{jk}(t)|\ \underline{N}(t), \mathbf{A}(t)\} = N_k A_{ik}(t)(\delta_{ij} - A_{jk}(t)) \tag{3.50}$$

Here δ_{ij} is the Kronecker delta function. In a manner analogous to the expectation, each element in the covariance matrix of $\underline{N}(t+1)$, $\mathbf{Q}(t)$, is a sum of the terms from (3.50),

$$\mathrm{Cov}\{N_i(t+1), N_j(t+1)|\ \underline{N}(t), \mathbf{A}(t)\} = Q_{ij}(t) = \sum_{k=1}^{m} N_k A_{ik}(t)(\delta_{ij} - A_{jk}(t)) \tag{3.51}$$

With a little rearranging, this can be written as a matrix equation,

$$\mathbf{Q}(t) = \mathrm{diag}\{\mathbf{A}(t)\underline{N}(t)\} - \mathbf{A}(t)\,\mathrm{diag}\{\underline{N}(t)\}\mathbf{A}(t)^T \tag{3.52}$$

These two moments are simple to compute and could be used in a variety of ways. In §4.5.3, this model will be used to construct control laws based on expected future

error. In §4.7, possible uses of this model in the context of optimal control laws will be mentioned. This model will also be used in §5.2 to construct approximate models suitable for use in a Kalman filter.

3.5 Chapter Summary

This chapter has shown how a system made up of many hybrid-state units, such as the actuator under discussion, can be described as a random process, whose state at any point in time is the number of units recruited into each discrete state. The random process model, which can be thought of as a state-space model with some state vector, probabilistic time evolution dynamics, and a linear output function, will be used to synthesize closed-loop control laws in chapter 4. It will also be used to construct full-state observer-controller systems in chapter 5.

The lengthy discussion of open-loop dynamics was included partially in response to the very large body of literature devoted to modeling all kinds of real systems as ensembles of random finite state agents. The open loop behavior of the actuator, while not of supreme interest to the design of an actuator, remains an interesting are of study, and an interesting point of connection between this work and the work of others.

Chapter 4

Feedback Control

4.1 Introduction

The goal of this thesis is to find a scalable way to control the summed output of many small binary units within a large distributed system. Specifically, the task at hand is to coordinate the behavior of actuators made from small compliant active material "motor units," placed in series or parallel, as shown in Fig. 4-1. One method of eliciting a varied response from many identical units is to randomize the behavior of each unit, which can be done using very simple hardware, as Appendix A demonstrates. A central controller determines the probabilities for transitioning between the relaxed and contracted states, and broadcasts these to all of the units. Chapter 3 demonstrated that the system dynamics imposed by this architecture are predictable using a variety of simple models. This chapter will explain how these models can be used to obtain convergent control of the system.

The process of developing control laws for stochastic recruitment systems occurs in several steps. First, in Section 4.3, sufficient conditions for probablity 1 convergence will be presented. This criterion is a straightforward and useful probability guarantee that can be made about a random system such as this one. Then, Section 4.4 will present quantitative metrics for performance that can be used for comparison between control laws that converge with probability 1, paying specific attention to how the system performs relative to the transient dynamics of the underlying units,

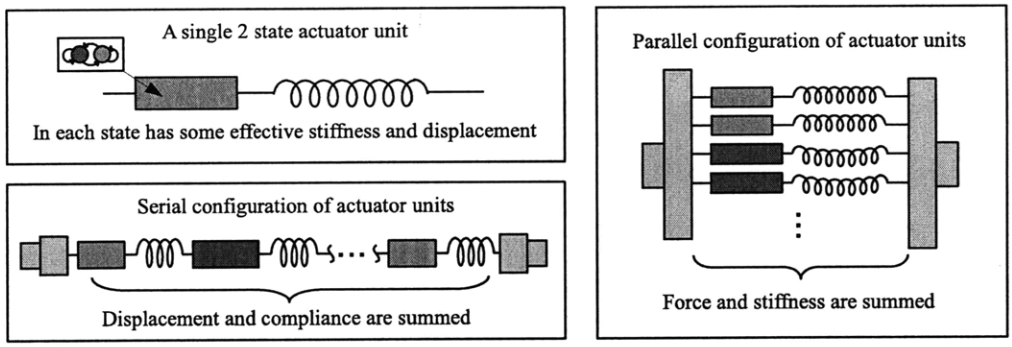

Figure 4-1: A diagram of the basic configuration of the 2-state actuator units.

and scalability of the system as units are added. Section 4.5 will go through several different heuristic control laws, which are compared by evaluating the expected convergence time to a neighborhood centered on a reference state $\underline{N}^{ref}$, as a function of the initial state distribution, $\underline{N}(0)$. These results are also compared to results from numerically computed optimal control laws. This chapter ends with an experimental illustration of how well the performance indicators predict the true system behavior.

4.2 Control problem statement

The practical control problem under consideration is to regulate the force, displacement, stiffness, or compliance of a serial or parallel network of small actuator units, as shown in Fig. 4-1. Each unit has two states, *on* and *off*, in which the active material is contracted or relaxed, respectively. The measured force or displacement $\underline{Y}(t)$ produced by the actuator is measured and used to produce an estimate of the number of units in each state, $\hat{\underline{N}}(t)$[1]. The simplest way that this estimate could be found is if the rank of the output matrix $\mathbf{H}$ for predicting $\underline{Y}$ based on $\underline{N}$ is equal to the number of states, m. Then the pseudo-inverse of $\mathbf{H}$ could be used to produce an estimate $\hat{\underline{N}}(t)$,

$$\hat{\underline{N}}(t) = \mathbf{H}^{\#}\underline{Y}(t) \tag{4.1}$$

[1]In this chapter, the estimate of the state distribution, $\hat{\underline{N}}(t)$, assumed to be accurate and feasible to compute. A more thorough treatment of state distribution estimation can be found in chapter 5.

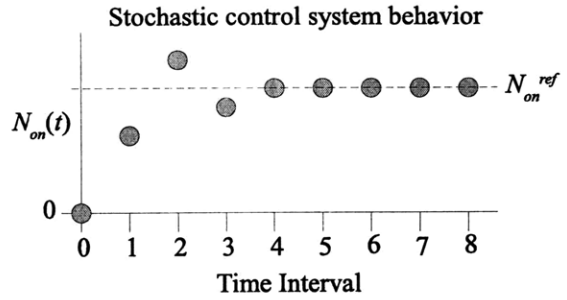

Figure 4-2: The desired control system response, while not exactly prescribed in the transient region, should settle to the desired state after a reasonably short number of time steps.

In the case of a two-state actuator with one output, for example, force, $\mathbf{H}$ can be made full rank if the number of units is constant, enabling the algebraic solution of the output equation for an estimated state distribution, as in (3.11),

$$F(t) = f_{on} N_{on}(t) + F_{off} N_{off} \tag{4.2}$$

$$N^{Total} = N_{on} + N_{off}$$

$$\mathbf{H} = \begin{bmatrix} f_{on} & f_{off} \\ 1 & 1 \end{bmatrix}$$

The state estimates are then fed into a control law, a function which outputs two values, $p(t)$ and $q(t)$. These values are the parameters defining a state transition matrix $\mathbf{A}(t)$, which is broadcast as a command to all of the units.

4.2.1 Defining a Target Set

It is useful sometimes to think about the reference or target as a set of states, rather than a single state. For example, one may wish to produce a force within 2% of a desired value. Several state distribution vectors may satisfy this output. The nomenclature $\underline{N}^{ref}$ will be used to refer to the nominal desired value, and the set R to refer to all states within some margin of error ϵ of the desired state:

$$R = \{\underline{N} : \ |N_{on} - N_{on}^{ref}| \leq \epsilon\} \tag{4.3}$$

This notation is very useful for discussing the convergence behavior of control laws.

4.3 Convergence Guarantees

A typical first step in the synthesis of control laws for any system is finding the general class of laws that satisfy a convergence or stability criterion. Once this basic property has been established, discussion about the relative performance of control laws within this class is possible. There are many notions of stability often used in this manner. A few are listed here:

1. Hurwitz stability (for linear systems with negative eigenvalues) [45]
2. Lyapunov stability (convergence to a point in state space) [57]
3. Barbalat's lemma (convergence to an invariant set in state space) [57]
4. Contracting coordinates (convergence to any particular solution of the system dynamics) [35]

For many stochastic systems, including the ensemble of recruited agents specific to this problem, these notions of convergence are not applicable, because all of these criteria rely on deterministic state evolution models. Instead, designers of stochastic control laws must rely on techniques:

1. The method of nonnegative supermartingale functions, that is, proof that the expected absolute output error looking forward in time is always less than the presently known output error, as presented by Doob [13].
2. Stochastic Lyapunov functions, a variant on the nonnegative supermartingale approach based on the state of a random process, rather than the output, thus making guarantees about the state space rather than the output space. This was proposed by Kushner [31].
3. Probability 1 convergence, or proof that the probability of being in the stable state approaches 1 as time increases [6].

4. Exponential probability 1 convergence, a variant of Pr{1} convergence that guarantees an exponential bound on the probability of being in the stable state as time increases [6].

In this thesis, exponential probability 1 convergence was chosen for selecting control law candidates because clear sufficient conditions for exponential Pr{1} convergence exist, and because exponential Pr{1} convergence provides useful guarantees about the computation of control law performance metrics.

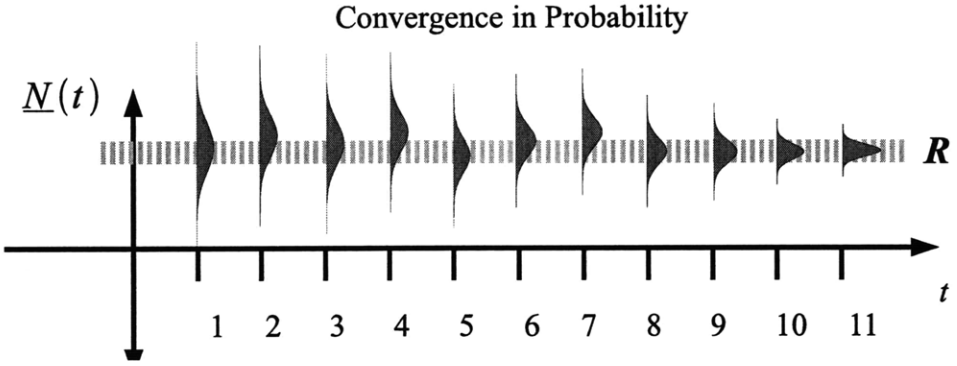

Figure 4-3: A random process which converges exponentially with probability 1 is one for which the probability of being in a target set R approaches one exponentially.

4.3.1 Sufficient conditions for Pr{1} convergence

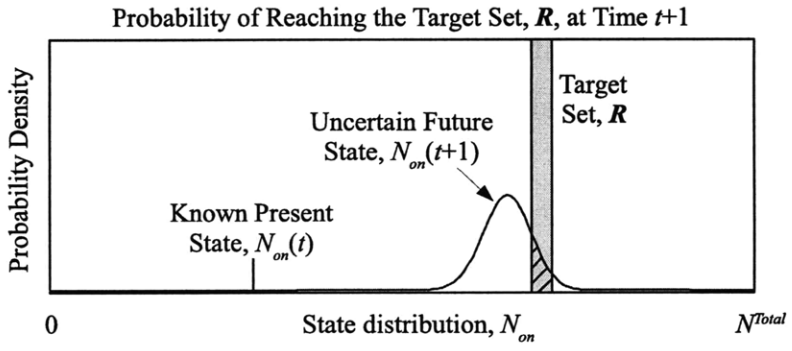

Figure 4-4: In order to converge with probability 1, the control law must guarantee that system reaches the target state with nonzero probability in bounded time.

Two sufficient conditions exist for proving probability one convergence for a random process. First, the set of target states R to which the system converges must

be *reachable* with some nonzero probability from any starting state, after a finite length of time. For the two-state example, one way to guarantee this would be to ensure that if the number of units currently in the *on* state, $N_{on}(t)$, is less than the desired number N_{on}^{ref}, then the probability of transitioning from *off* to *on* must be non-infinitesimally greater than zero and less than one:

$$\begin{aligned} N_{on}(t) < N_{on}^{ref} : \ \delta \leq p(t) \leq 1-\delta, \ \ 0 \leq q(t) \leq 1-\delta \\ N_{off}(t) < N^{Total} - N_{on}^{ref} : \ 0 \leq p(t) \leq 1-\delta \ \ \delta \leq q(t) \leq 1-\delta \\ \delta > 0 \end{aligned} \tag{4.4}$$

If these restrictions are observed, then the probability of reaching R will be non-zero at any point in time, as shown in Fig. 4-4. The probability of reaching the desired state can be bounded from below by some ρ_{min}, which can be found by considering the least likely state transition behavior possible given some command. This would be the joint probability of all units transitioning from *off* to *on* given a command to turn *on* with minimal probability $p(t) = \delta$,

$$\rho_{min} = \delta^{N^{Total}} \tag{4.5}$$

The second sufficient condition needed to ensure convergence with probability 1 is that the system must halt all state transitions when the state distribution $\underline{N}(t)$ is within the target set R, so that the target set of states is absorbing[2],

$$\underline{N}(t) \in R : p(t) = 0, \ q(t) = 0 \tag{4.6}$$

Because of this, the probability of being in the target state at time $t+1$, conditioned on being in the target state at time t, is 1. It is possible to bound the probability of being outside of the target set at any point in time recursively, by conditioning on

[2]A set of states is absorbing if, once entered, the system never leaves. It is essentially equivalent to the noation of an invariant set.

whether the system is in the target state one time step prior:

$$\begin{aligned}\Pr\{\underline{N}(t) \notin R\} = \Pr\{\underline{N}(t) \notin R |\ \underline{N}(t-1) \in R\} \Pr\{\underline{N}(t-1) \in R\} + \\ \Pr\{\underline{N}(t) \notin R |\ \underline{N}(t-1) \notin R\} \Pr\{\underline{N}(t-1) \notin R\} \\ \Pr\{\underline{N}(t) \notin R\} \leq (1 - \delta^{N^{Total}}) \Pr\{\underline{N}(t-1) \notin R\}\end{aligned} \tag{4.7}$$

Thus, the probability that the system is not in the target set diminishes exponentially.

4.3.2 Properties of $\Pr\{1\}$ Convergent Control Laws

The guarantees provided by probability one convergence are subtle. The fact that a control law makes a system converge with probability one does not necessarily mean that it is a good control law, in any sense. For illustration, say that Alice is trying to locate a long lost friend Bob by picking up the phone and dialing random ten digit numbers until Bob answers. This could be represented as a control problem, if the system state is defined to be whether or not Alice has reached Bob. The control policy would then be:

- Bob has not been reached: Alice dials a random 10 digit number.
- Bob has been reached: Alice stops dialing numbers.

Assuming that Bob has a US phone number, then this control policy will converge on Alice finding Bob with probability one, since there is a non-zero probability that she dials the right number every time she randomly selects the right one. By any practical performance metric, say, the amount of time it takes, or the amount due on Alice's next phone bill, it is an insane proposition. If Bob has one number among the 10 billion numbers Alice could dial randomly, then the expected number of tries it takes Alice to reach Bob friend is 10 billion. The real benefit obtained from proving probability one convergence is that many practical performance metrics can be calculated – specifically, performance metrics that taking the general form of the expectation of

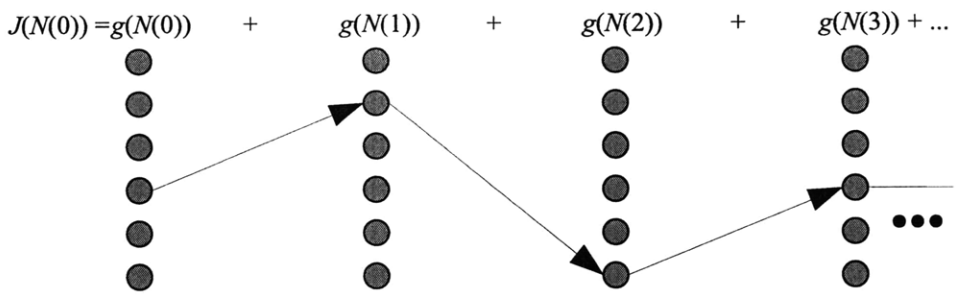

Figure 4-5: The cost functional J is evaluated for a specific state trajectory by summing a cost-per-stage in every state through time.

a summed, non-negative, bounded function of the state at any point in time,

$$J(\underline{N}(0)) = \mathrm{E}\left\{ \sum_{t=0}^{\infty} g(\underline{N}(t)) \middle| \underline{N}(0) \right\} \tag{4.8}$$

$$0 \leq g(\underline{N}(t)) \leq G \tag{4.9}$$

A cost function of this kind, essentially an integration of cost along a state trajectory, as shown in Fig. 4-5, is in the commonly used form of a discrete time functional, often used for Hamiltonian optimal control. All such performance metrics are guaranteed to be bounded. The proof of this bound follows:

Proof. The cost of any state at any point in time $g(\underline{N}(t))$ is bounded from below and above, $0 \leq g(\underline{N}(t)) \leq G$, and is identically zero when the system state $\underline{N}(t)$ is in the target set R. The expected cost can be bounded by the following expression:

$$0 \leq \mathrm{E}\left\{ \sum_{t=0}^{\infty} g(\underline{N}(t)) \middle| \underline{N}(0) \notin R \right\} \leq G \sum_{t=0}^{\infty} \Pr\{\underline{N}(t) \notin R | \underline{N}(0) \notin R\} \tag{4.10}$$

If the probability of not being in the target state is 1 at time $t = 0$, then (4.7) can be bounded as an explicit sequence,

$$\Pr\{\underline{N}(t) \notin R | \underline{N}(0) \notin R\} \leq (1 - \delta^{N^{Total}})^t \tag{4.11}$$

Applying this bound to (4.10), we find that the total cost can be bounded by a convergent series,

$$0 \leq \mathrm{E}\left\{ \sum_{t=0}^{\infty} g(\underline{N}(0)) \middle| \ \underline{N}(0) \notin R \right\} \leq G \sum_{t=0}^{\infty} (1 - \delta^{N^{Total}})^t = \frac{G}{\delta^{N^{Total}}} \tag{4.12}$$

This ends the proof. The boundedness of the cost function is very important because it is necessary for the convergence of many iterative methods of estimating and optimizing J.

4.4 Computing Expected Convergence Time

One very simple cost function to use for a system such as this is the expected time it takes the system to converge. This can be expressed in the summed, non-negative form just discussed. The cost of any state is defined to be 1 if the system has not reached the target set, and 0 if it has,

$$g(\underline{N}) = \begin{cases} 0, & \underline{N} \in R \\ 1, & \underline{N} \notin R \end{cases} \tag{4.13}$$

Optimal control problems of this kind are often called stochastic shortest path problems. The computation of numerically optimal control laws will be discussed, but it is also useful to compute the cost of heuristic control laws. These heuristic laws are much cheaper to compute in real time than the numerically estimated optimal control laws, and their performance is comparable. Chapter 4 of Bertsekas' *Dynamic Programming and Optimal Control* is a fairly good reference on this topic; It covers everything in this section and the previous section regarding proofs of convergence, boundedness of cost functions, and iterative computation of cost functions [6].

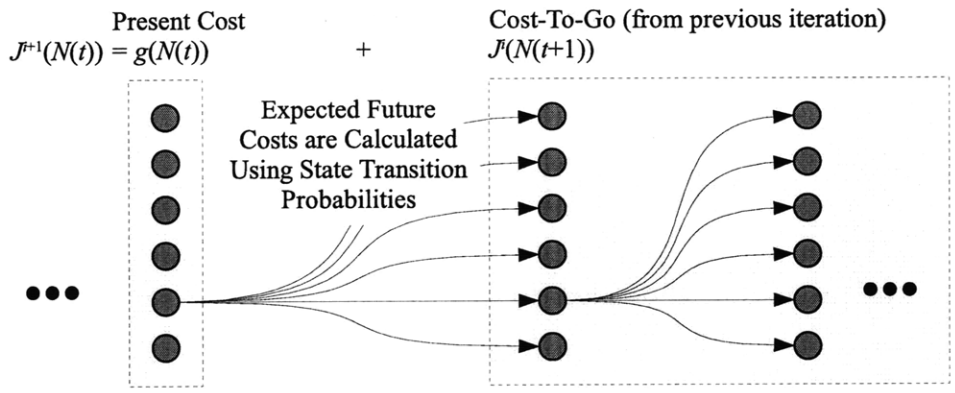

Figure 4-6: If the process is Markov and the cost-per-stage g is solely a function of the state, then the expected future cost can be treated recursively, and estimated iteratively.

4.4.1 Iterative Cost Function Computation

The fact that the state trajectory is determined by a Markov process makes it easier to compute, if the cost is solely a function of state and not of time. Consider a cost function in the form:

$$J(\underline{N}(0)) = \mathrm{E}\left\{ \sum_{t=0}^{\infty} g(\underline{N}(t)) \middle| \ \underline{N}(0) \right\} \tag{4.14}$$

If the cost at the present state is known, then the right hand side can be split into two terms:

$$J(\underline{N}(0)) = g(\underline{N}(0)) + \mathrm{E}\left\{ \sum_{t=1}^{\infty} g(\underline{N}(t)) \middle| \ \underline{N}(0) \right\} \tag{4.15}$$

The sum remaining inside the expectation brackets is the expectation of the cost function starting one step ahead,

$$\begin{aligned} J(\underline{N}(0)) &= g(\underline{N}(0)) + \mathrm{E}\left\{ J(\underline{N}(1)) | \ \underline{N}(0) \right\} \\ &= g(\underline{N}(0)) + \sum_{k} J(\underline{N}_k) \Pr\{\underline{N}(1) = \underline{N}_k | \ \underline{N}(0)\} \end{aligned} \tag{4.16}$$

Here $\underline{N}_k$ is each element of the set of possible state distributions, and $J(\underline{N}_k)$ is the cost function assuming that $\underline{N}_k$ is the initial state distribution of the ensemble. This recursive expression is called Bellman's equation. Because J depends on state, and

not on time, Bellman's equation can be thought of as an identity that the cost function must satisfy. It can also be used to iteratively estimate J by successively iterating the computation of J in each state, starting with an initial estimated cost function J^0 which is used to compute J^1, and so on, until the estimate converges on the true value:

$$J^{i+1}(\underline{N}) = g(\underline{N}) + \sum_k J^i(\underline{N}_k) \Pr\{\underline{N}(t+1) = \underline{N}_k | \ \underline{N}\} \tag{4.17}$$

This estimation method, called constant policy value iteration, is convergent because the control laws we are choosing converge with probability 1. Figure 4-6 illustrates the recursive estimation based on a truncated cost series.

4.4.2 Second Moment Approximation

Once the cost function is estimated, it is also possible to compute the second moment of the same functional [58]. This second moment cost function looks like this:

$$J_2(\underline{N}(0)) = \mathrm{E}\left\{ \left(\sum_{t=0}^{\infty} g(\underline{N}(t)) \right)^2 \middle| \ \underline{N}(0) \right\} \tag{4.18}$$

The first cost term can be taken out by binomial expansion,

$$\begin{aligned} J_2(\underline{N}(0)) &= g(\underline{N}(0))^2 + 2g(\underline{N}(0)) \, \mathrm{E}\left\{ \sum_{t=1}^{\infty} g(\underline{N}(t)) \middle| \ \underline{N}(0) \right\} \\ &\quad + \mathrm{E}\left\{ \left(\sum_{t=1}^{\infty} g(\underline{N}(t)) \right)^2 \middle| \ \underline{N}(0) \right\} \\ &= g(\underline{N}(0))^2 + 2g(\underline{N}(0)) \, \mathrm{E}\{J(\underline{N}(1)) | \ \underline{N}(0)\} + \mathrm{E}\{J_2(\underline{N}(1)) | \ \underline{N}(0)\} \end{aligned} \tag{4.19}$$

Substituting in (4.16), this can be reduced to a Bellman-like equation,

$$\begin{aligned} J_2(\underline{N}(0)) &= g(\underline{N}(0))^2 + 2g(\underline{N}(0))(J(\underline{N}(0)) - g(\underline{N}(0))) \\ &\quad + \sum_k J_2(\underline{N}_k) \Pr\{\underline{N}_k | \ \underline{N}(0)\} \\ &= 2g(\underline{N}(0))J(\underline{N}(0)) - g(\underline{N}(0))^2 + \sum_k J_2(\underline{N}_k) \Pr\{\underline{N}_k | \ \underline{N}(0)\} \end{aligned} \tag{4.20}$$

This second moment equation then can be iteratively approximated via the same method of guesses (constant policy value iteration) to find the variance of any control law.

4.5 Comparing Several Control Laws

Iterative cost function approximation is intended as a tool for the synthesis of good control laws. The logical way to design a control law for some application is as follows:

1. Define a control problem in terms of the desired outcome and a system model.
2. Define a convergence criterion which all control laws must satisfy.
3. Define a performance metric for the comparison of laws.
4. Propose multiple control law candidates satisfying the convergence criteria specified, and compare the performance of the candidate laws.

Thus far, this chapter has followed this template. The control laws that follow are intended as exemplar laws demonstrating the upper and lower performance bounds that can be expected from reasonably well-posed control laws. The first control law attempts the most naïve control strategy, which, incidentally, resembles the open-loop input plans most closely, and requires the least amount of information to implement. The other control law candidates look more like traditional control policies: linear feedback and feedback based on the moment model from §3.4.2. The last control law considered is the numerically computed law minimizing the expected convergence time to the desired set R.

4.5.1 Minimal Feedback Laws

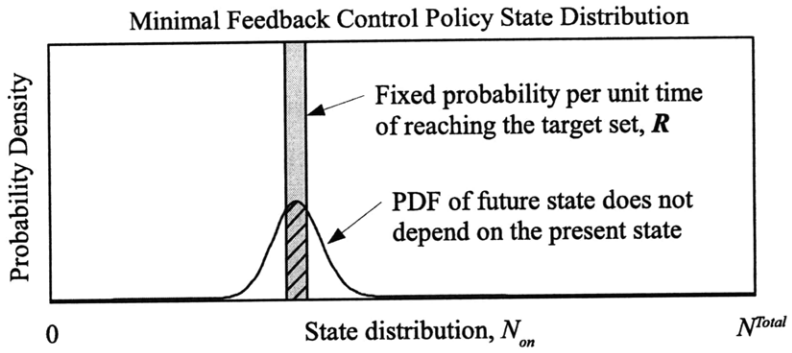

Figure 4-7: The probability per unit time of convergence for the minimal feedback law is constant, as shown by this plot of the probability distribution of $N_{on}(t)$. at any point in time.

One simple control law which serves as a good bounding case is the law which seeks to drive the system toward the desired state $\underline{N}^{ref}$ as fast as possible using an open-loop input like those discussed in §3.3.5, with the exception that when the system reaches the desired set of states, the command is broadcast to cease all transitions. It is analogous to a bang-bang positioning stage that has a dead reckoning sensor, so that the controller can halt when the desired output has been reached. The control policy can be expressed in terms of the transition graph parameters $p(t)$ and $q(t)$,

$$p(t) = \begin{cases} \frac{N_{on}^{ref}}{N^{Total}}, & \underline{N}(t) \notin R \\ 0, & \underline{N}(t) \in R \end{cases} \tag{4.21}$$

$$q(t) = \begin{cases} \frac{N^{Total} - N_{on}^{ref}}{N^{Total}}, & \underline{N}(t) \notin R \\ 0, & \underline{N}(t) \in R \end{cases}$$

The second-largest eigenvalue of the open-loop response to this control law, λ_2, as computed in (3.30), is zero. As a corollary, the covariance between successive predictions of $\underline{N}(t)$, as calculated in (3.35), is also zero. Thus, this law has an independent chance of reaching the target set R at every point in time. This chance can be calculated by integrating the probability distribution over the target set, as depicted in Fig. 4-7. Because the odds of reaching the target state are independent at any point

in time, and because the system will stay in the target state once it is there, the distribution of convergence times should be exponential.

The minimal feedback law is described here because with any amount of useful information from the system, such as the magnitude of the output error, the performance of the minimal feedback law should be improved upon. Appendix B contains a more in-depth analysis of minimal feedback laws, and lays out the rationale for why the particular law presented here (choosing p and q such that $\lambda_2 = 0$) is a good bounding case candidate for comparison.

4.5.2 Linear Laws

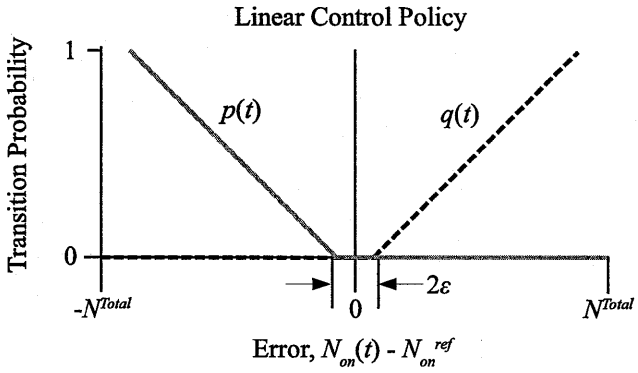

Figure 4-8: An illustration of the linear feedback control policy.

The first proposed control laws for the feedback control of stochastic recruitment actuators was in the form of proportional feedback [64]. The units are commanded to transition unilaterally from the state with too many units to the state with too few, with a probability proportional to the error, as shown in Fig. 4-8,

$$p(t) = \begin{cases} k\frac{N_{on}^{ref} - N_{on}(t)}{N^{Total}}, & N_{on}(t) < \underline{N}_{ref}^{on} - \epsilon \\ 0, & N_{on}(t) \geq \underline{N}_{ref}^{on} - \epsilon \end{cases} \tag{4.22}$$

$$q(t) = \begin{cases} k\frac{N_{off}^{ref} - N_{off}(t)}{N^{Total}}, & N_{off}(t) < N_{off}^{ref} - \epsilon \\ 0, & N_{off}(t) \geq N_{off}^{ref} - \epsilon \end{cases} \tag{4.23}$$

The choice of gain k in a linear control law is limited by the convergence criteria

outlined above. These criteria could also be posed in terms of a nonnegative supermartingale bound, as we showed in [63]. Practically, the gain k always produces convergent results for $\delta < k < 1$, though the best performance for target states near the middle of the state space for the two-state problem is about 1.4. It would be possible to investigate some kind of gain scheduling for this system, but simpler solutions were found in the form of moment-based control laws.

4.5.3 Moment-Based Laws

The most direct approach to control law derivation is motivated by the moment-based model from §3.4.2. These control laws seek to reduce the expected future error as a function of the present state and the state transition graph parameters,

$$\mathrm{E}\{\underline{N}^{ref} - \underline{N}(t+1) |\, \underline{N}(t), \mathbf{A}(t)\} = \underline{N}^{ref} - \mathbf{A}(t)\underline{N}(t) \tag{4.24}$$

Suppose it is desirable to guarantee that the expectation of this error decreases over time exponentially. This can be expressed as a constraint on the value of $\mathbf{A}(t)$:

$$\begin{aligned} \mathrm{E}\{\underline{N}^{ref} - \underline{N}(t+1) |\, \underline{N}(t), \mathbf{A}(t)\} &= \alpha(\underline{N}^{ref} - \underline{N}(t)) \\ \underline{N}^{ref} - \mathbf{A}(t)\underline{N}(t) &= \alpha(\underline{N}^{ref} - \underline{N}(t)) \\ \mathbf{A}(t)\underline{N}(t) &= \alpha\underline{N}(t) + (1-\alpha)\underline{N}^{ref} \end{aligned} \tag{4.25}$$

Here α is a design parameter that affects the rate at which the system converges. This also means that the error, represented as a random variable, is a nonnegative supermartingale, or non-positive sub-martingale, depending on the initial sign of the error, because it is defined in such a way that the value of the expected one-step-ahead error is always a fraction of the present error. One practical factor determining the choice of α is the avoidance of overshoot. Figure 4-9 shows the probability distribution on the future state of a two-state system, with one law predicting no expected error ($\alpha = 0$) and another predicting some expected error, but no overshoot. For the two-

state system under full-state control, the moment-based control law can be computed,

$$p(t) = \begin{cases} (1-\alpha)\frac{N_{on}^{ref} - N_{on}(t)}{N_{off}}(t), & N_{on}(t) < \underline{N}_{ref}^{on} - \epsilon \\ 0, & N_{on}(t) \geq \underline{N}_{ref}^{on} - \epsilon \end{cases} \tag{4.26}$$

$$q(t) = \begin{cases} (1-\alpha)\frac{N_{off}^{ref} - N_{off}(t)}{N_{on}(t)}, & N_{off}(t) < N_{off}^{ref} - \epsilon \\ 0, & N_{off}(t) \geq N_{off}^{ref} - \epsilon \end{cases} \tag{4.27}$$

This is very similar in form to the linear control laws proposed above. The term introducing the non-linearity in the denominator could be thought of as a kind of state-dependent gain, which demonstrates why further inquiries in the direction of gain scheduling were abandoned as probably redundant.

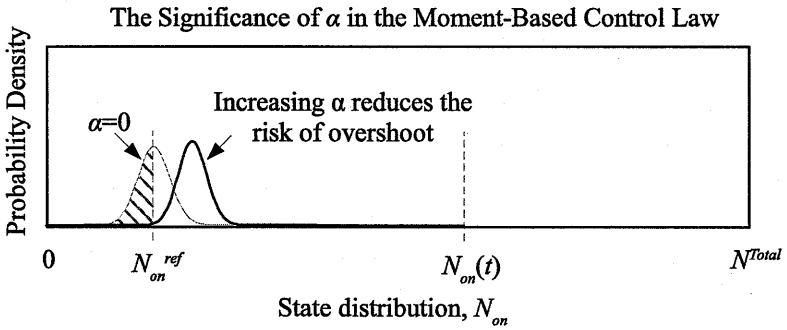

Figure 4-9: The probability of overshooting the desired state, shown hatched, can be reduced by increasing the value of α.

4.5.4 Optimal Control Laws

The definitive bookend case for comparison of control law performance is the optimal control law. This law can be computed numerically for any initial state and any target set using value iteration, as previously discussed. Rather than using the same control policy for every iteration of cost approximation, as was done in (4.17), the state transition parameters p and q are optimized for each state at iteration to minimize the expected future cost. The cost estimate J^i and the values of p and q for each

state converge simultaneously on the optimal cost and the optimal control policy,

$$J^{i+1}(\underline{N}) = g(\underline{N}) + \min_{p,q} \left\{ \sum_k J^i(\underline{N}_k) \Pr\{\underline{N}(t+1) = \underline{N}_k | \ \underline{N}(t) = \underline{N}, p, q\} \right\} \quad (4.28)$$

The output of this algorithm is an optimal expected cost function $J^*(\underline{N})$, and a vector of values for p and q defining the optimal state transition graph from each state.

4.5.5 Comparing Expected Settling Time

The four control laws just introduced were compared for the case of an actuator having 200 two-state agents, being driven to a desired state of $N^{ref} = 125$ ON units, with a margin of error of $\epsilon = 2$ units. The control law parameters were chosen as follows:

1. A minimal feedback control law.
2. A linear feedback control law, $k = 1.4$. This gain was tune by observing performance over a range of target states and a range of initial conditions.
3. A moment-based feedback control law designed to set the expected future error to 0 ($\alpha = 0$).
4. The numerically computed feedback control law minimizing the expected convergence time.

Figure 4-10 shows the results of expected settling time for these three laws. All three laws are expected to converge, from any initial condition, to the target within 5 time intervals. Recall from §3.2.1 that the length of the time interval here is determined by the settling time of the continuous-time dynamics, ΔT_S. The moment-based control law performs strictly better than either of its competitors, with an expected convergence time never exceeding twice the continuous settling time of the underlying physical units. The minimal feedback law appears to perform surprisingly well compared to the linear feedback law. However, the close performance is only skin-deep. The variance of the convergence time, plotted in Fig. 4-11, shows that the

minimal control law exhibits much more variance than either of the two laws that utilize feedback information more intelligently. The reason for this huge performance advantage is that the linear and moment-based control laws reduce the commanded transition probabilities as the system nears the desired state. The variance in the number of units transitioning becomes correspondingly smaller. Being close counts for something, in other words. Another way of looking at these performance comparisons is a histogram of simulation results. Figure 4-12 shows a histogram of simulation results for the three control laws under consideration. The difference between the minimal, linear and moment-based control laws is evident from the shape of the distribution. The convergence times of the minimal control law are, as discussed in §4.5.1, exponentially distributed with a long "tail," reflecting the predicted increase in variance.

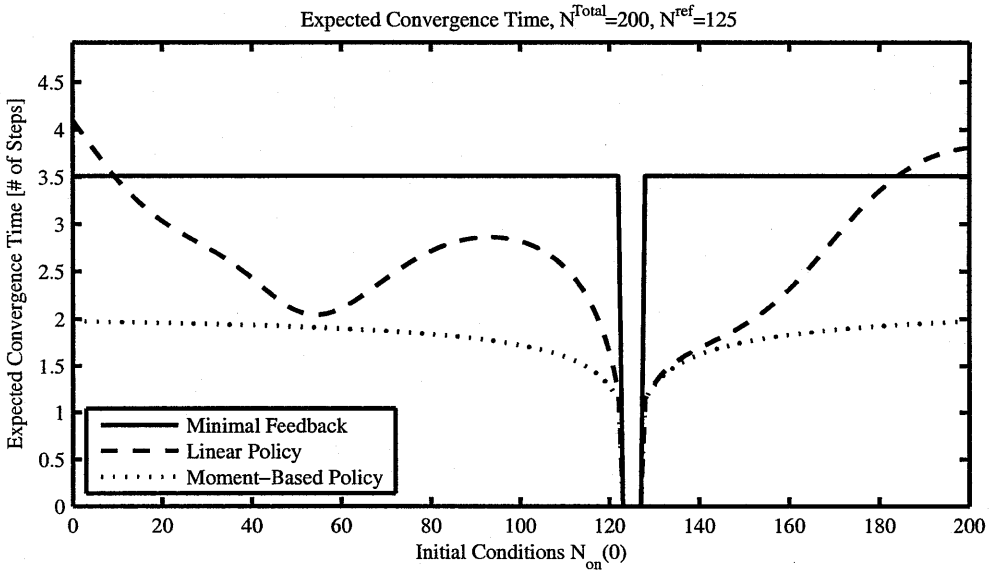

Figure 4-10: A comparison of expected convergence time for the minimal, linear and moment-based control laws, $N^{Total} = 200$, $N_{on}^{ref} = 125$

The optimal control law performed predictably better than the other control laws, with the exception of the moment-based control law. Here, the difference between the expected convergence time of the two control laws was so small that the numerical computation of the cost function was not accurate enough to distinguish the two for most initial values of N_{on}. Figure 4-13 shows the same performance comparison made

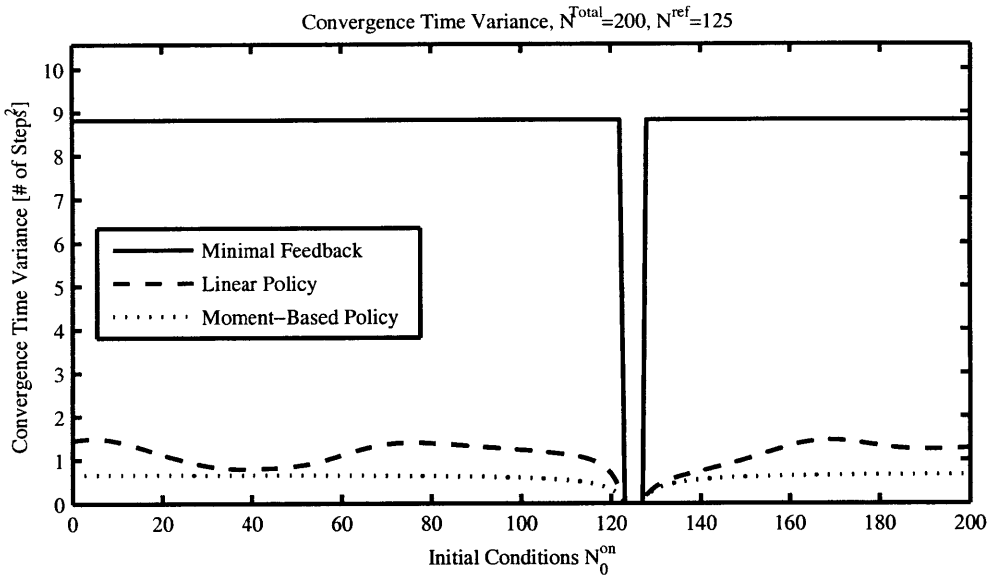

Figure 4-11: A comparison of the convergence time variance for the minimal, linear, and moment-based laws.

in Fig. 4-10 between just the optimal and moment-based laws. This is good news; the numerically computed optimal control law requires a look-up table, or a parametric approximation, to implement. It is far simpler to use the moment-based law as an acceptable replacement.

The final quantitative comparison that is useful to understand is the behavior of the expected convergence time as the number of units increases. Figure 4-14 shows the expected convergence time as a function of the initial state for the moment-based, zero expected error control law having 200, 400, and 2000 units. For an even comparison, the window of acceptable error was held constant at +/-1% of the total number of units. The resulting plot shows that the expected convergence time decreases as the number of units increases. Remember that the shortest time in which the system can converge is 1 interval, so the vertical asymptote as N^{Total} increases is actually in the middle of the plot, not at the bottom where $J = 0$. The fact that the scaling is within a single time interval of the minimum value is impressive and encouraging. It implies that the overhead due to coordinating all of the finite state agents is a small (less than a multiplicative factor of 2), and that this overhead factor decreases as N^{Total} is increased.

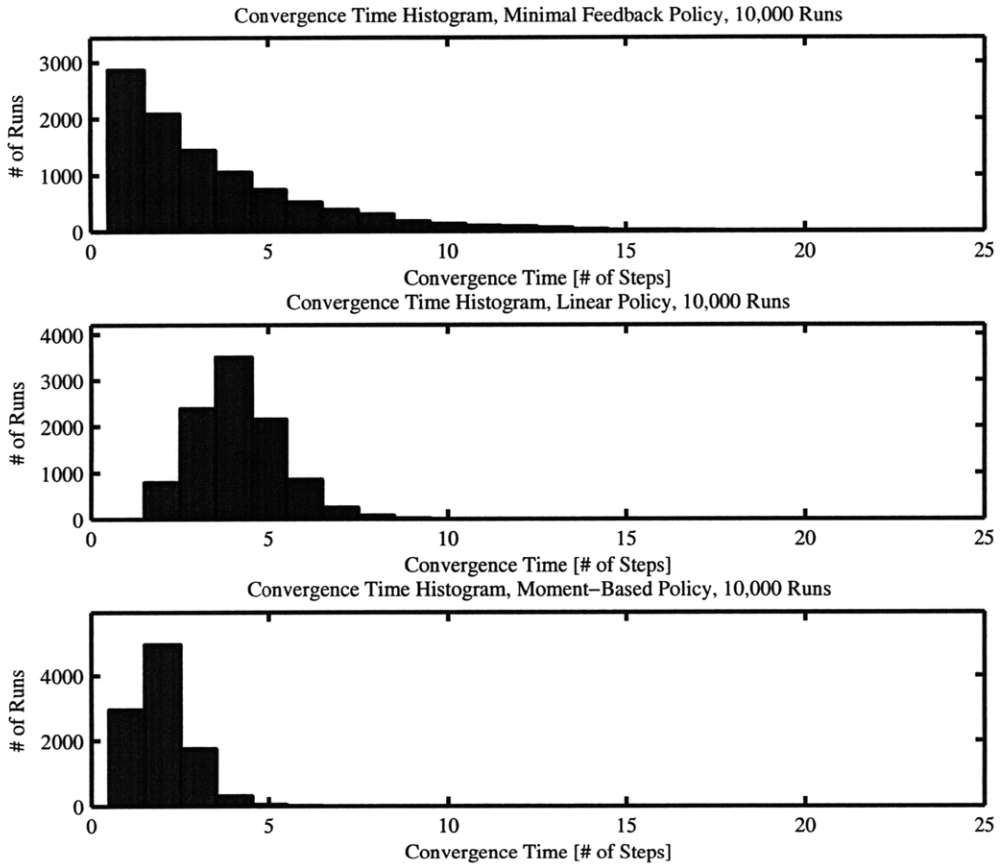

Figure 4-12: Convergence time histograms for 10,000 simulations of the minimal, linear, and moment-based laws, illustrating the reason for the much higher variance of the minimal control law.

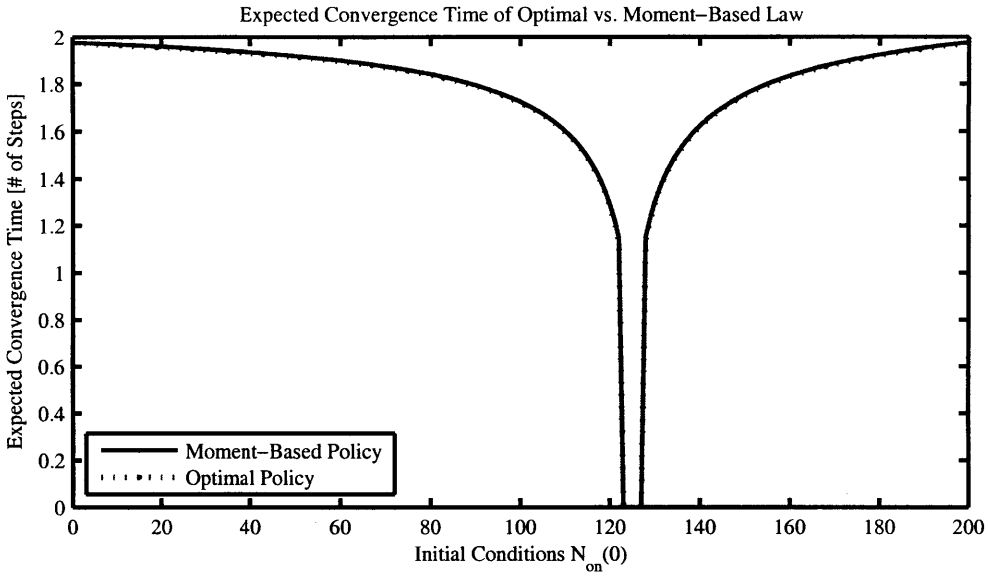

Figure 4-13: The performance of the numerically optimal control law for the system is barely distiguishable from the moment-based control law.

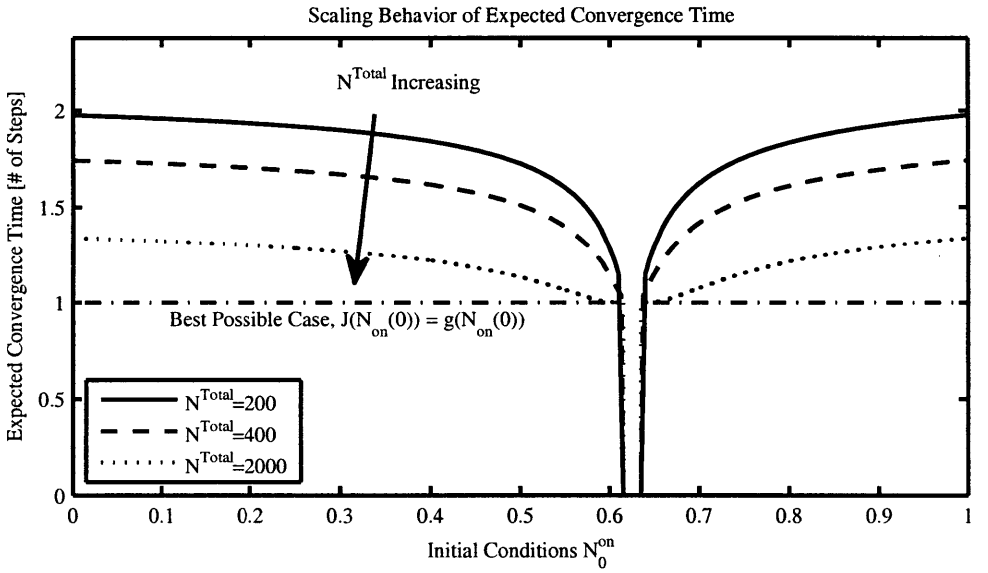

Figure 4-14: Expected convergence time for the moment-based control law, showing the performance improvements as N^{Total} increases.

4.6 An Experimental Example

To confirm that the predictions of expected convergence time are accurate, an experimental actuator composed of 50 segments of SMA wire placed in a serial chain was constructed. This chain, shown in Fig. 4-15, had a total active displacement of approximately 350 mm. Each individual element was a helical spring looped back and forth through a rigid plate, supporting approximately 4 N of force. The wire used was Toki BMX 150 SMA helix, purchased from Toki BioMetal [10]. The units were wired on a bus to a National Instruments data acquisition card, and controlled using a central control program written in C. The control program measured only the output displacement of the actuator at a constant load using a potentiometer attached to a capstan, then computed the state transition probabilities according to the moment-based law, with parameter α set to 0.4. The state transitions of each unit were simulated in the central computer, and then the state of each unit was sent through the bus. The units were individually controlled with field effect transistors, so that when each unit was in the *on* state, 300 mA of current passed through the SMA spring. All of the units were cooled using forced air convection. To determine the time interval for sampling and control, the step response of all units in the actuator activated simultaneously was measured, and is shown in Fig. 4-16. A settling time of 4 s was chosen. This is shorter than the true settling time ΔT_S, but it was within 10 percent of the steady-state, so the control law was able to cope with the modeling error due to shortened sampling intervals. Based on this 4 s time interval, the expected convergence time of the actuator was calculated, and is shown in Fig. 4-17. The initial condition from which the system was started in trials is marked on the plot, along with the predicted settling time of approximately 18 seconds. The trajectories of 5 control step responses are plotted in Fig. 4-18. All of the trajectories had settled or were within a few percent of settling by the predicted settling time. Straggler trajectories are possible, of course, but one thing that was clear in both simulation and experiment was that most of the experimental runs that failed to converge within the anticipated time window were very close to the desired point,

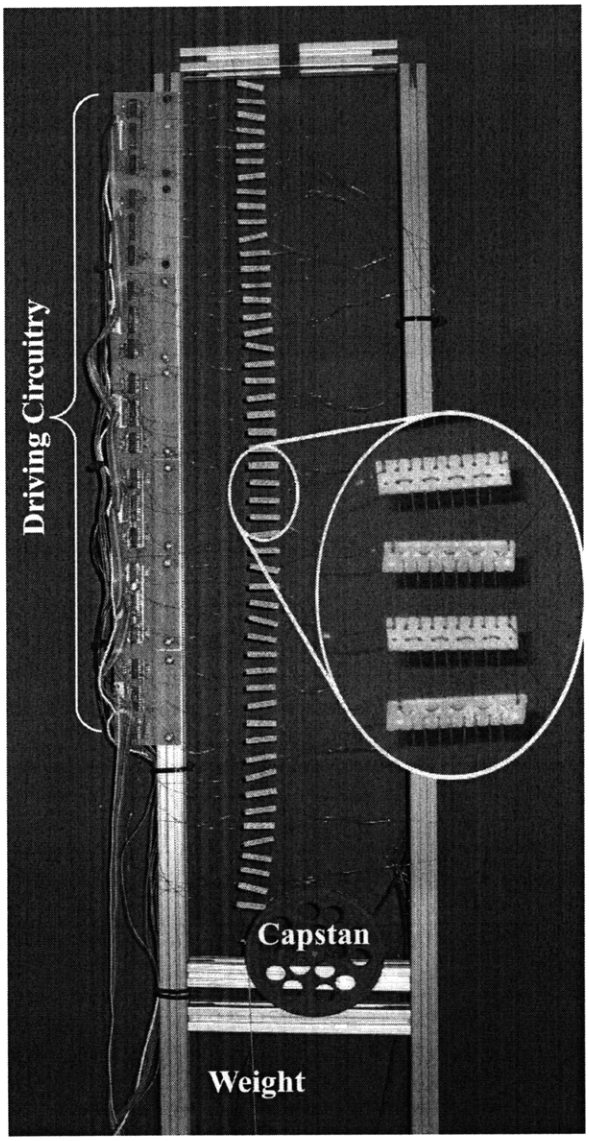

Figure 4-15: The experimental apparatus, consisting of 50 shape memory alloy elements placed in series.

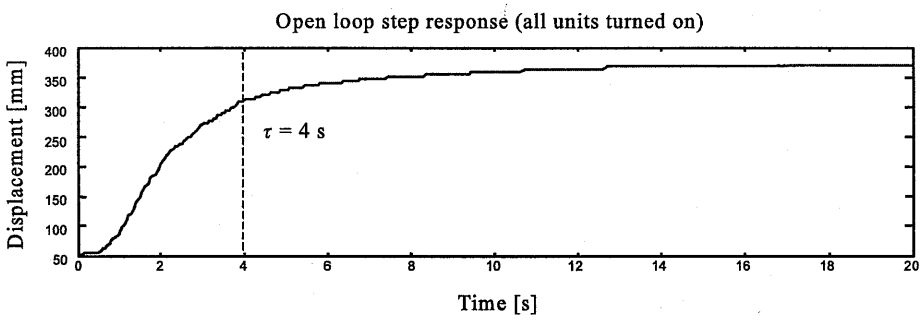

Figure 4-16: The response of the actuator when all units are simultaneously contracted, showing the system settling time.

but not completely settled. This agrees with the observation that the variance in the rate of transitions decreases dramatically for small errors, so that the response will be very small but not unpredictable. The experiment was a success from the

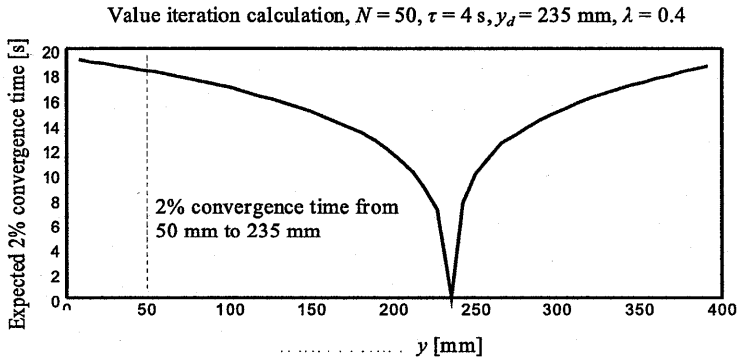

Figure 4-17: A plot of the expected settling time for the actuator.

standpoint of verifying that the dominant factor constraining the performance of the actuator was the physics of the active material itself. The estimated cost functions did a good job predicting the performance of the actuator, and inspired confidence that they could be used as design tools for speculating on the performance of actuator design candidates.

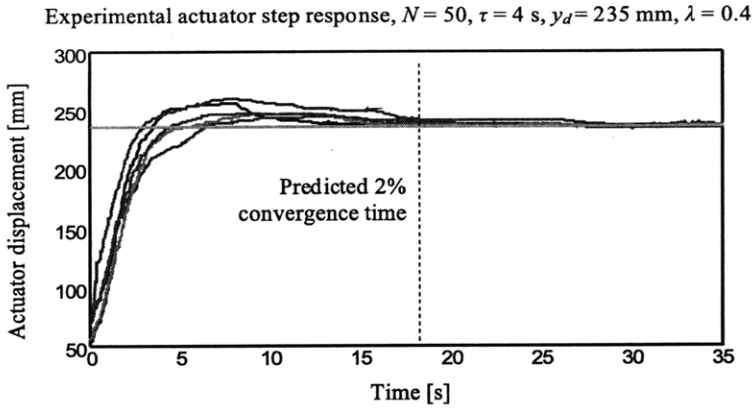

Figure 4-18: Five experimental tracking trajectories, showing the reference position and the expected 2% convergence time, for reference.

4.7 Generalization to Many States

The framework presented in Chapter 3 was all given in general terms, for state machines with m states, not two. In higher dimensions, the control problem becomes more complicated, but it remains tractable. The two-state problem generalizes to the problem of an ensemble of N^{Total} independent m-state agents, transitioning based on $\mathbf{A}(t)$. The controller must choose more than two parameters defining $\mathbf{A}(t)$, based on an estimated state distribution which becomes more difficult to estimate than the simple inversion proposed in (4.1). Chapter 5 discusses the general problem of producing state estimates from linear system outputs. Nonetheless, if problems of state estimation are addressed, then many of the same design approaches can be used to synthesize and evaluate control laws.

4.7.1 The Target Set

Previously, the target set, R, was defined in terms of a desired state $\underline{N}^{ref}$. While this could still be the case, it may also make more sense to look at convergence of the output for a system whose states may number more than the number of outputs. The error considered is no longer the error between the expected future state and the desired state, but the error $|\mathbf{H}\underline{N} - \underline{Y}^{ref}|$ between the expected future output and the

desired output,

$$R = \{\underline{N} : \ |\mathbf{H}\underline{N} - \underline{Y}^{ref}| \leq \epsilon\} \tag{4.29}$$
$$= \{\underline{N} : \ |\mathbf{H}(\underline{N} - \underline{N}^{ref})| \leq \epsilon\}$$

Other than this, not much has to change in the examination of convergent control laws. The same criteria still apply: ability to reach R in bounded time with non-zero probability, and the ability to stop the system once in R.

4.7.2 Control Law Synthesis

The moment-based control law framework is particularly useful for synthesizing control laws in m states, because it provides a hard constraint on the parameters of the state transition matrix $\mathbf{A}(t)$. For example, the future expected error could be constrained to zero,

$$\mathrm{E}\{\underline{Y}^{ref} - \underline{Y}(t+1)|\ \underline{N}(t), \mathbf{A}(t)\} = 0 \tag{4.30}$$
$$\underline{Y}^{ref} - \mathbf{H}\mathbf{A}(t)\underline{N}(t) = 0$$

Unless $\mathbf{H}$ has full rank, this will not completely constrain $\mathbf{A}(t)$. The remaining free parameters could be eliminated in several ways, either heuristically or via optimization. One heuristic approach would be to trim all transitions that are blatantly counterproductive. For example, the probability of leaving a state with too few agents should be zero, or the probability of entering a state with too many agents should be zero. This will reduce by half the parameter space. Free parameters could also be accounted for by minimizing the *a priori* variance of the future state, perhaps using the trace of $\mathbf{Q}(t)$, as defined in (3.51), while applying (4.30) as a constraint using the constraint vector Λ,

$$\mathbf{A}(t) = \arg\min_{\Lambda,\mathbf{A}} \left\{ \Lambda^T \left(\mathbf{H}\mathbf{A}\underline{N}(t) - \underline{Y}^{ref} \right) + \sum_{j=1}^{m} \sum_{k=1}^{m} N_k(t) A_{jk}(t)(1 - A_{jk}(t)) \right\} \tag{4.31}$$

When this is simplified, it reduces to the problem of minimizing the a weighed sum of the negative squared terms of $\mathbf{A}(t)$,

$$\mathbf{A}(t) = \arg\min_{\Lambda,\mathbf{A}} \left\{ \Lambda^T \left(\mathbf{H}\mathbf{A}\underline{N}(t) - \underline{Y}^{ref} \right) - \sum_{k=1}^{m} N_k(t) \left(\sum_{j=1}^{m} A_{jk}(t)^2 \right) \right\} \tag{4.32}$$

Of course, this minimization is tricky because of the many equality and inequality constraints placed on elements of $\mathbf{A}(t)$,

$$\sum_{i=1}^{m} A_{ij} = 1 \tag{4.33}$$

$$0 \leq A_{ij}(t) \leq 1 \tag{4.34}$$

Halasz and Berman's 2007 paper deals with minimization under these constraints to some degree for the problem of open-loop transition graph planning [18], but to date the problem of general, closed-loop control using this framework remains open. Alternative approaches exist that do not pose control choice as optimization; Chattopadhyay and Ray have developed algorithms for choosing an open-loop graph $\mathbf{A}(t)$ which works in the presence of arbitrary upper bounds on the elements of $\mathbf{A}(t)$ [8]. While it is beyond the scope of this thesis to pursue many-state control laws, it is quite possible that some of the techniques being applied in that area for open-loop command planning could be adapted for use in closed-loop control.

4.7.3 Control Law Performance Evaluation

The calculation of cost functions for large state spaces is very costly, because it involves marginalizing the future cost over all possible future states iteratively. There are, however, other techniques which will suffice. For example, Monte Carlo sampling methods can be used to approximate the expected future cost by generating 30-100 one-step-ahead simulations. This process, when repeated iteratively, appears to converge on the correct expected cost function. Figure 4-19 shows a picture of the cost function for an ensemble of three state agents, trying to reach a reference state.

The number of agents in the third state, $N_3(t)$, is constrained by the total number of agents, so it is not shown. The upper diagonal is outside the range of possible states ($N_1(t) + N_2(t) > N^{Total}$). The state transition graph parameters were "pruned" by eliminating unwanted transitions, then constrained by (4.30).

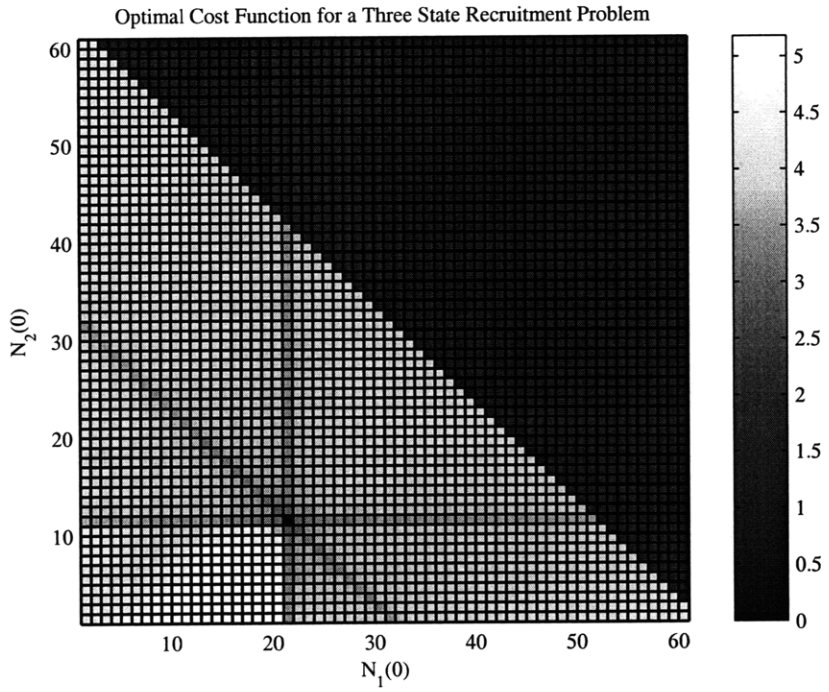

Figure 4-19: The expected convergence time of a three-state recruitment law minimizing expected convergence time for 60 units to a reference of $[10\ 20\ 30]^T$.

4.8 Conclusion

This chapter has shown that the problem of synthesizing control laws for actuators in the stochastic recruitment framework is straightforward, once good models of the random dynamics are applied and basic convergence criteria are accounted for. Because stochastic recruitment can be posed rigorously in the framework of control of discrete stochastic processes, the development of feedback control laws for this architecture can be related to the much larger literature on optimal control of stochastic processes.

The potential impact of this work on other problems in coordinating the ensemble behavior of swarms has yet to be delved into. To date, I have found no other researchers pursuing this line of reasoning within the framework of closed-loop stochastic control, although Augung Julius, Adam Halasz, and George Pappas have recently come up with some very promising work applying simple switching control laws to two-state control problems in the regulation of bacterial gene expression [27]. One future direction that will definitely be pursued in this area is the formulation of control laws for constrained control problems such as theirs. This may also necessitate abandoning probability 1 convergence as a criteria for admissible laws, in favor of the more forgiving ergodicity criteria needed for posing this control problem as one of expected cost-per-stage.

Chapter 5

State observation

5.1 Introduction

The core theoretical result presented in this thesis is the control of an ensemble of random, hybrid-state units in a state-space framework, as shown in Fig. 5-1. In chapter 4, control laws were formulated under the assumption that the distribution of states among agents, $\underline{N}(t)$, is measurable. This prompts a natural follow-up chapter on the feasibility of measuring or estimating $\underline{N}(t)$ given only the outputs $\underline{Y}(t)$ that are to be controlled. The modeling of populations of discrete-state units using a Kalman filter is common among researchers studying population distributions of real animals [61]. The application of these techniques to ensembles of artificial systems appears to be quite novel, however.

This chapter demonstrates that it is possible to construct a simple linear estimating filter that can serve as a state observer for stochastic recruitment systems. This is done by implementing a Kalman filter based on a simplified, moment-based model of the ensemble of agents. The relative ease with which an observer can be constructed for stochastic recruitment controllers is a significant argument in favor of this approach to control of ensemble outputs from distributed systems. The usefulness of state observers will be demonstrated by showing how higher-order finite state machine models can be used to approximate the transient switching dynamics of the hybrid-state units. These higher order models, which require the use of state

observers, ensure that system performance does not suffer from large time delays caused by waiting for the transient state transition outputs to settle.

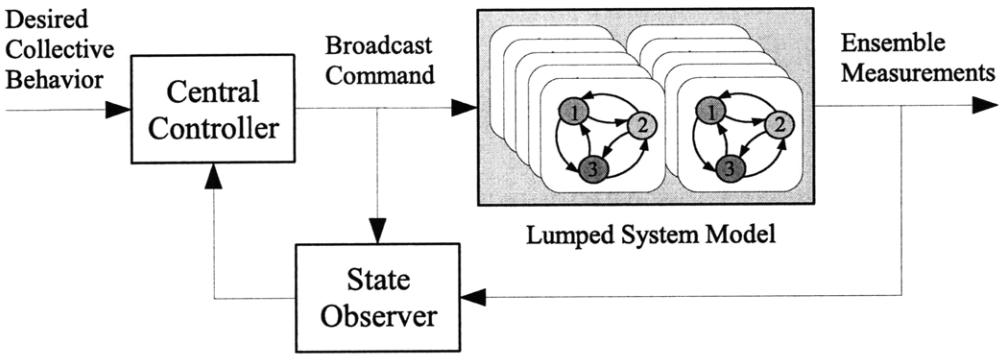

Figure 5-1: One key appeal of the stochastic recruitment control framework is the ability to represent a complex distributed ensemble of finite state agents (shaded, center) in terms of a simple lumped model using the aggregate number of agents in each state instead of complete knowledge of each agent's state. This state distribution can be estimated using a Kalman filter, by observing the true output of the recruitment system.

5.1.1 Review: Linear State Observers

In order to be clear on concepts and terminology, we will review briefly the practical mechanics of state observation for discrete-time systems. A state observer is a virtual system model that is augmented using feedback from sensor measurements to approximate the true behavior of the system. For example, consider a linear, time-varying (LTV) system with a state vector $\underline{N}(t)$ and a measurement vector $\underline{Y}(t)$, which can be described by a difference equation, as was shown in (3.48) and (3.9),

$$\begin{aligned} \underline{N}(t) &= \mathbf{A}(t)\underline{N}(t-1) \\ \underline{Y}(t) &= \mathbf{H}\underline{N}(t) \end{aligned} \tag{5.1}$$

The time evolution of an estimate $\underline{\hat{N}}$ can be assumed to behave in a similar fashion:

$$\begin{aligned} \underline{\hat{N}}(t) &= \mathbf{A}(t)\underline{\hat{N}}(t-1) \\ \underline{\hat{Y}}(t) &= \mathbf{H}\underline{\hat{N}}(t) \end{aligned} \tag{5.2}$$

However, one would expect that this model would inevitably diverge from the true system behavior if the estimated state is perturbed even slightly from the true state, or if there is a small discrepancy between the system model and the true system. An observer utilizes feedback from the error between the output prediction $\underline{\hat{Y}}(t)$ and the measured output $\underline{Y}(t)$ to correct this:

$$\begin{aligned} \underline{\hat{N}}(t|t-1) &= \mathbf{A}(t)\underline{\hat{N}}(t-1) \\ \underline{\hat{Y}}(t) &= \mathbf{H}\underline{\hat{N}}(t|t-1) \\ \underline{\hat{N}}(t) &= \underline{\hat{N}}(t|t-1) - \mathbf{K}(t)(\underline{\hat{Y}}(t) - \underline{Y}(t)) \end{aligned} \tag{5.3}$$

If the estimated output $\underline{\hat{Y}}(t)$ is identical to the true output $\underline{Y}(t)$, then the feedback term is identically zero, and equations (5.2) and (5.3) are identical. If they are different, then feedback matrix $\mathbf{K}(t)$ should be chosen to drive the estimated state toward the actual state. Of course, ensuring that this happens is the hard part of observer design. The choice (or existence) of $\mathbf{K}(t)$ that guarantees a good state estimate has been discussed at great length in a variety of contexts, such as the observability matrix, as the stability of the error system $\underline{N}(t) - \underline{\hat{N}}(t)$, as a problem of contracting coordinate systems, and as a problem in optimal control. For the purposes of this review, we will not go into much detail. The important thing to remember is that at the end of the day, some matrix $\mathbf{K}(t)$ is computed and incorporated into an observer-controller.

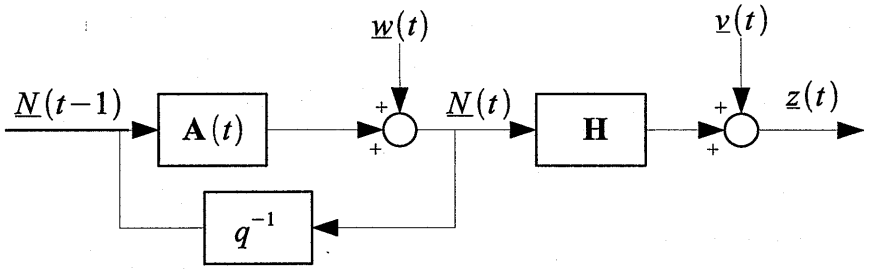

Figure 5-2: The system to be observed by the Kalman filter, represented as a finite difference equation. The time evolution of the estimates is represented as multiplication by a matrix and addition with a random variable.

5.1.2 Observation as an Optimal Control Problem

Rudolf Kalman and his contemporaries[1] viewed the problem of state observation within the framework of optimal control [28], [29]. They assumed that a system of the form of (5.1) had some "drift" associated with state update that could be written as an additive noise signal $\underline{w}(t)$, a random variable with zero mean and a covariance matrix $\mathbf{Q}(t)$. They assumed that any uncertainty in the output could similarly be expressed as an additive noise signal $\underline{v}(t)$, a random variable with zero mean and a covariance matrix $\mathbf{R}(t)$:

$$\begin{aligned} \underline{N}(t) &= \mathbf{A}(t)\underline{N}(t-1) + \underline{w}(t) \\ \underline{Y}(t) &= \mathbf{H}\underline{N}(t) + \underline{v}(t) \\ \mathrm{E}\{\underline{w}(t)\} &= 0; \mathrm{Cov}\{\underline{w}(t)\} = \mathbf{Q}(t) \\ \mathrm{E}\{\underline{v}(t)\} &= 0; \mathrm{Cov}\{\underline{v}(t)\} = \mathbf{R}(t) \end{aligned} \tag{5.4}$$

Figure 5-2 shows a block diagram of this estimated model. Kalman famously showed that there is a recursive formula for computing the linear observer gain $\mathbf{K}(t)$ which minimizes the expected squared state estimation error $\mathrm{E}\{|\underline{N}(t) - \hat{\underline{N}}(t)|^2\}$. This gain

[1]Many people, despite their substantial contributions to the development of optimal filters, were not lucky enough to have their names attached so ubiquitously to the end product.

is computed as a function of a time-evolving estimation error covariance matrix, $\mathbf{P}(t)$:

$$\mathbf{P}(t|t-1) = \mathbf{A}(t)\mathbf{P}(t-1)\mathbf{A}^T(t) + \mathbf{Q}(t) \tag{5.5}$$

$$\mathbf{S}(t) = H\mathbf{P}(t|t-1)H^T + \mathbf{R}(t) \tag{5.6}$$

$$\mathbf{K}(t) = \mathbf{P}(t|t-1)\mathbf{H}^T\mathbf{S}(t)^{-1}$$

$$\mathbf{P}(t) = (I - \mathbf{K}(t)\mathbf{H})\mathbf{P}(t|t-1)$$

The intermediate variable $\mathbf{S}(t)$ represents the covariance of the estimated output $\underline{\hat{Y}}(t)$. Kalman's choice of $\mathbf{K}(t)$ is not validated by an explicit error stability criterion as before, but is rather a question of whether the minimum estimation covariance attained by the cost function is small enough that the estimate is useful. At any point in time cost function, $\mathrm{E}\{|\underline{N}(t) - \underline{\hat{N}}(t)|^2\}$, is equal to the trace of $\mathbf{P}(t)$. Therefore, if the trace of $\mathbf{P}(t)$ grows too large, then the best possible estimate of the system's state given the outputs defined by $\mathbf{H}$ is not good enough to provide useful information about the system's state.

5.1.3 Bayesian Filtering and the Kalman Filter

Another, more rigorous approach to observing the hidden state of a random process is the Bayesian estimator. A Bayesian estimator treats a state estimate as a probability distribution over all possible states of a system. Rather than tracing the time evolution of the system with a deterministic model having additive noise, as shown in Fig. 5-2, a Bayesian estimator models the time evolution of a system in terms of conditional probability distributions, from the previous state estimate to the current state estimate, and from the current state estimate to the current output estimate, as depicted in Fig. 5-3. Information is incorporated into the state estimate by considering the measured output as a random event, whose *a priori* probability can be determined from the output model. Bayes' rule, a simple statement of reciprocity between conditional probability distributions, is used to compute a recursive *a posteriori* state estimate conditioned on both the prior state and the current measurement.

Interestingly, when the Bayesian estimator framework is applied to a Gauss-Markov

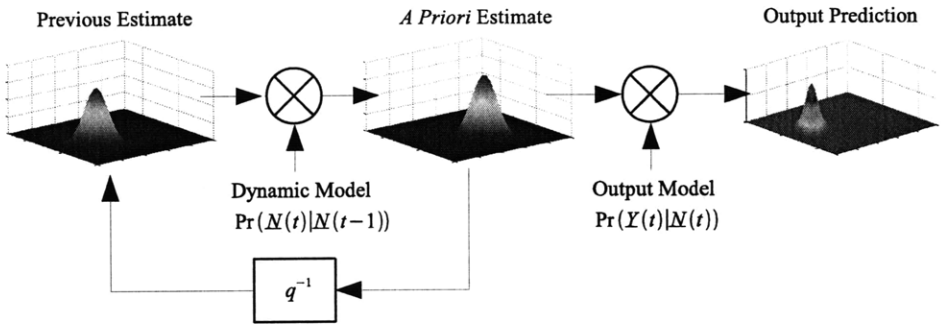

Figure 5-3: A recursive Bayesian *a priori* model for a random process, written in terms of conditional probabilities.

random process (a LTV random process with Gaussian noise), the resulting estimates are identical to the estimates produced by the Kalman filter. In other words, the distribution obtained over all possible states is a Gaussian distribution with a mean of $\underline{\hat{N}}(t)$ and a covariance of $\mathbf{P}(t)$. This duality is important because it implies that the Kalman filter is statistically correct in a much higher sense for Gauss-Markov processes. In Kalman and Bucy's 1961 paper, the authors point out that the equivalent statement in optimal filtering parlance is that the Kalman filter is an optimal *linear* observer for any process having noise on any distribution [29]. However, the Kalman filter is only optimal over *all linear and non-linear* feedback observers only if the process noise is Gaussian.

The practical implication of the duality between classical filtering and Bayesian filtering is that the quality of an estimate produced by a Kalman filter is contingent on how "close" the additive process noise is to Gaussian. If the noise distribution has one central "peak" and light "tails", then the Kalman filter is a reasonable method of calculating observer gains for a LTV system. At this point, the relevance of this discussion to the recruitment problem is apparent. The state behaviors of many finite state machines making independent, identically-distributed state transitions will, by the central limit theorem, exhibit ensemble behavior that becomes very close to Gaussian as their numbers increase. The expected behavior of the system will be in the form of a linear update law. Thus, Kalman's equation for calculating filter

gain is an eminently reasonable approach to the design of observers for stochastic recruitment control systems.

5.1.4 Chapter Outline

The remainder of this chapter will be devoted to showing how a moment-based model of the actuator units' state distribution can be cast in the framework of a Kalman filter estimator, and examining the properties of the filter that results. The model that was introduced in §3.4 will be revisited and approximated so that a bilinear model of estimation covariance is possible. The Kalman filter implementation will be presented, along with a discussion of filter behavior. The *a priori* estimation covariance will be shown to converge if a constant command is given to the system. General convergence is difficult to prove; however, this has not been a problem in simulation or experiment.

The latter half of this chapter is a look at a practical application for state observers for stochastic recruitment control. The two-state actuator model that is the main focus of this thesis is augmented with additional discrete states to model some of the continuous-state dynamics of the actuator units. The augmented models require a Kalman filter to produce state estimates for control. The resulting controller is able to function at a much faster update rate, and consequently is better able to actively resist disturbances applied to the actuator. An experimental example is given demonstrating the ability of the observer-controller to reduce the sampling time of a force-controlled actuator.

5.2 Adapting the Moment-Based Process Model for Observation

Section §3.4 showed that it is possible to predict the expectation and variance of $\underline{N}(t+1)$ from the current state, $\underline{N}(t)$ and the current command, $\mathbf{A}(t)$ by computing the moments of the transitions to and from each state. These moments are then

summed to produce the expectation and covariance of $\underline{N}(t+1)$,

$$\mathrm{E}\{\underline{N}(t+1)|\ \underline{N}(t), \mathbf{A}(t)\} = \mathbf{A}(t)\underline{N}(t) \tag{5.7}$$
$$\mathrm{Cov}\{\underline{N}(t+1)|\ \underline{N}(t), \mathbf{A}(t)\} = \mathbf{Q}(t) = \mathrm{diag}\{\mathbf{A}(t)\underline{N}(t)\} - \mathbf{A}(t)\,\mathrm{diag}\{\underline{N}(t)\}\mathbf{A}(t)^T$$

The difference between the model from §3.4 and the model needed for a Kalman filter is that some additional independence constraints must be enforced on the model if the conventional Ricatti equation form for time evolution of the *a priori* estimation covariance can be used. An illustrative way to see this is to represent the knowledge about the present state $\underline{N}(t)$ as an unbiased estimate $\hat{\underline{N}}(t)$ to which a zero-mean estimation error variable $\underline{e}(t)$ with covariance $\mathbf{P}(t)$ has been added,

$$\underline{N}(t) = \hat{\underline{N}}(t) + \underline{e}(t) \tag{5.8}$$
$$\mathrm{E}\{\underline{e}(t)\} = 0 \tag{5.9}$$
$$\mathrm{Cov}\{\underline{e}(t)\} = \mathbf{P}(t) \tag{5.10}$$

When computing the time evolution, the expectation and covariance from (5.7) can be viewed as multiplication by a matrix and addition of a zero-mean noise signal $\underline{w}$, representing the uncertainty in the state update,

$$\underline{N}(t+1) = \mathbf{A}(t)\underline{N}(t) + \underline{w}(\underline{N}(t), t) \tag{5.11}$$
$$\mathrm{E}\{\underline{w}(\underline{N}(t), t)\} = 0 \tag{5.12}$$
$$\mathrm{Cov}\{\underline{w}(\underline{N}(t), t)\} = \mathbf{Q}(\underline{N}(t), t) \tag{5.13}$$

The additive noise $\underline{w}$ depends on the present state distribution $\underline{N}(t)$, as evidenced by (5.7). When the time evolution of an estimate is computed, this state-dependence becomes problematic because $\underline{w}$ and $\underline{e}$ are coupled,

$$\underline{N}(t+1) = \mathbf{A}(t)\hat{\underline{N}}(t) + \mathbf{A}(t)\underline{e}(t) + \underline{w}(\hat{\underline{N}}(t) + \underline{e}(t), t) \tag{5.14}$$

The covariance of $\mathbf{A}(t)\underline{e}(t)$ is easy to compute:

$$\mathrm{Cov}\{\mathbf{A}(t)\underline{e}(t)\} = \mathbf{A}(t)\,\mathrm{Cov}\{\underline{e}(t)\}\mathbf{A}(t)^T = \mathbf{A}(t)\mathbf{P}(t)\mathbf{A}(t)^T \tag{5.15}$$

The covariance of $\underline{w}(\underline{\hat{N}}+\underline{e}(t))$ is much more difficult to express. What is really needed is some upper bound $\underline{\tilde{w}}(t)$ whose covariance $\tilde{\mathbf{Q}}(t)$ is independent of $\underline{N}(t)$, and which provides a conservative estimate of the state update covariance. Fortunately, conservation dictates that the number of agents in each state, $N_i(t)$, is always nonnegative and smaller than N^{Total},

$$0 \leq N_i(t) \leq N^{Total} \tag{5.16}$$

This bound can be used to construct an upper bound on the covariance of the number of transitions to any two destination states i and j from a single originating state k, as calculated in (3.50),

$$\begin{aligned} \mathrm{Cov}\{T_{ik}(t), T_{jk}(t)|\ \underline{N}(t), \mathbf{A}(t)\} &= N_k A_{ik}(t)(\delta_{ij} - A_{jk}(t)) \\ 0 \leq \mathrm{Cov}\{T_{ik}(t), T_{jk}(t)|\ \underline{N}(t), \mathbf{A}(t)\} &\leq N^{Total} A_{ik}(t)(\delta_{ij} - A_{jk}(t)) \end{aligned} \tag{5.17}$$

In turn, the elements of $\mathbf{Q}(t)$ found in (3.51) can be bounded,

$$\begin{aligned} \mathrm{Cov}\{N_i(t+1), N_j(t+1)|\ \underline{N}(t), \mathbf{A}(t)\} = Q_{ij}(t) &= \sum_{k=1}^{m} N_k A_{ik}(t)(\delta_{ij} - A_{jk}(t)) \\ 0 \leq Q_{ij}(t) &\leq N^{Total} \sum_{k=1}^{m} A_{ik}(t)(\delta_{ij} - A_{jk}(t)) \end{aligned} \tag{5.18}$$

In other words, the covariance of the bounding additive signal $\underline{\tilde{w}}(t)$ is calculated by assuming that all of the agents are in every state. This assumption overestimates the total uncertainty by a factor of the number of states, m, but it achieves indepdence of the state distribution $\underline{N}(t)$. Using the bounding model, the *a priori* calculation of the estimation expectation and covariance is the standard LTV process model used

for constructing Kalman filters,

$$\underline{N}(t+1) = \mathbf{A}(t)\underline{\hat{N}}(t) + \mathbf{A}(t)\underline{e}(t) + \underline{\tilde{w}}(t) \tag{5.19}$$

$$\underline{\hat{N}}(t+1) = \mathbf{A}(t)\underline{\hat{N}}(t) \tag{5.20}$$

$$\underline{e}(t+1) = \mathbf{A}(t)\underline{e}(t) + \underline{\tilde{w}}(t) \tag{5.21}$$

$$\mathrm{Cov}\{\underline{e}(t+1)\} = \mathbf{P}(t+1|t) = \mathbf{A}(t)\mathbf{P}(t)\mathbf{A}(t)^T + \tilde{\mathbf{Q}}(t) \tag{5.22}$$

5.3 Modeling output behavior

The other area where the model from §3.4 fails to provide enough detail for the Kalman filter is in the output covariance. We need a way to model the covariance of an output prediction, $\mathbf{S}(t)$, in a manner similar to the additive assumptions made above.

$$\underline{Y}(t) = \mathbf{H}(\underline{\hat{N}} + \underline{e}(t)) + \underline{v}(t) \tag{5.23}$$

$$\mathbf{R}(t) = \mathrm{Cov}\{\underline{v}(t)\} \tag{5.24}$$

$$\underline{\hat{Y}}(t) = \mathrm{E}\{\underline{Y}(t)\} = \mathbf{H}\underline{\hat{N}} \tag{5.25}$$

$$\mathbf{S}(t) = \mathrm{Cov}\{\underline{Y}(t)\} = \mathbf{H}\mathbf{P}(t)\mathbf{H}^T + \mathbf{R}(t) \tag{5.26}$$

One of the terms on the right hand side of (5.26) is due to the state distribution uncertainty, $\mathbf{P}(t)$; the other, $\mathbf{R}(t)$, is the result of actual variability in the output of the agents. In general, $\mathbf{R}(t)$ should be nonsingular in order for $\mathbf{S}(t)$ to have full rank [17]. The value of these terms must be calculated based on variation in the output of individual units. Hitherto the system has been assumed to consist of identical agents having identical outputs in each state. However, manufacturing imprecision could result in some output variability among the physical properties of each one. This will result in some measurement uncertainty that is a function of the number of agents in each state. Consider for a moment the case in which each sub-unit in the system has an output that can be characterized in each state as having some independent variance, The expected value of the output is equal to $\mathbf{H}\underline{N}(t)$, as previously defined.

state	k^{th} expected output	k^{th} output variance
1	H_{k1}	γ_{k1}
2	H_{k2}	γ_{k2}
$\vdots$	$\vdots$	$\vdots$
m	H_{km}	γ_{km}

The output variance of each output is a function of the state,

$$\text{Var}\{Y_k\} = R_{kk} = \sum_{j=1}^{m} N_j \gamma_{kj} \tag{5.27}$$

For example, consider an actuator made of many small actuator units connected in series, whose lengths have been trimmed by hand. Each unit produces approximately 20 mm of displacement, but manual error has introduced a standard deviation of 4 mm into the output when contracted, and 2 mm when relaxed. The covariance of this single output would then be a scalar,

$$\text{Var}\left\{\sum \delta(t)\right\} = 1.6 \times 10^{-5} \cdot N_{on}(t) + 4 \times 10^{-6} \cdot N_{off}(t) \tag{5.28}$$

In order to neatly fit within the Kalman filter framework, it may be useful to come up with a good bounding case for this output variance. This can be accomplished by noting that the number of agents in each state is bounded. Consequently, the output variance is bounded by the following relationship:

$$R_{kk} = \sum_{j=1}^{m} N_j \gamma_{kj} \leq N^{Total} \max_j \gamma_{kj} = \tilde{R}_{kk} \tag{5.29}$$

This bounding case is very similar in form to the bound assumed for the per-stage update covariance, $\tilde{\mathbf{Q}}(t)$, in (5.18).

5.3.1 Conservation Constraints on the Output

In section 3.2.3, we mentioned that the space of valid state distribution vectors $\underline{N}(t)$ is smaller than the whole vector space, due to conservation. This can be translated

into a linear constraint on the estimates of $\underline{N}(t)$, and further into a pseudo-output whose value is always known,

$$\sum_{k=1}^{m} N_k(t) = N^{Total} \tag{5.30}$$

$$[\ 1\ \ 1\ \ \ldots\ \ 1\ ]\underline{N}(t) = N^{Total}$$

This linear constraint is useful for constraining the estimate of $\underline{N}(t)$ to make sure that it is physically consistent. The pesudo-output matrix to be appended to the physical outputs will be defined as $\mathbf{H}_0$,

$$\mathbf{H}_0\underline{N}(t) = \sum_{k=1}^{m} N_k = N^{Total} \tag{5.31}$$

This output can be used for "feedback" to constrain the modeled system behavior. From here on we will assume that all output matrices include this appended row of ones. Based on the definitional constraints on $\mathbf{A}(t)$ associated with being a conservative Markov chain, each column of $\mathbf{A}(t)$ must sum to one. This means that $\mathbf{H}_0\mathbf{A} = \mathbf{H}_0$ for all valid state transition matrices.

$$\mathbf{H}_0\mathbf{A} = \mathbf{H}_0 \tag{5.32}$$

$$\mathbf{H}_0\mathbf{A}^{-1} = \mathbf{H}_0$$

These constraints have some practical implications on the output covariance of the system due to uncertainty in the state estimate. Specifically, the uncertainty as to the total number of agents is unaffected by the time evolution process. This is a direct consequence of the definitional constraints on $\mathbf{A}$. Recall that the component of the output covariance $\mathbf{S}(t)$ due to *a priori* estimation uncertainty is determined by $\mathbf{H}$ and $\mathbf{P}$. For the pseudo-measurement, the component of the output covariance can be written as a matrix $\mathbf{S}_0(t)$,

$$\mathbf{S}_0(t) = \mathbf{H}_0\mathbf{P}(t|t-1)\mathbf{H}_0^T \tag{5.33}$$

Substituting (5.22) into $\mathbf{P}(t|t-1)$, the output covariance can be expressed in terms of the prior estimation covariance and $\mathbf{Q}$ or $\tilde{\mathbf{Q}}$,

$$\mathbf{S}_0(t) = \mathbf{H}_0\mathbf{P}(t-1)\mathbf{H}_0^T + \mathbf{H}_0\mathbf{Q}(t)\mathbf{H}_0^T \tag{5.34}$$

The added uncertainty per stage caused by time evolution is the term containing $\mathbf{Q}(t)$. Expanding this in terms of the definition of $\mathbf{Q}$, we find that the identities from (5.32) simplify this expression,

$$\begin{aligned}\mathbf{H}_0\mathbf{Q}(t)\mathbf{H}_0^T &= \mathbf{H}_0\,\mathrm{diag}\{\mathbf{A}(t)\underline{N}(t)\}\mathbf{H}_0^T - \mathbf{H}_0\mathbf{A}(t)\,\mathrm{diag}\{\underline{N}(t)\}\mathbf{A}(t)^T\mathbf{H}_0^T \\ &= \mathbf{H}_0\,\mathrm{diag}\{\mathbf{A}(t)\underline{N}(t)\}\mathbf{H}_0^T - \mathbf{H}_0\,\mathrm{diag}\{\underline{N}(t)\}\mathbf{H}_0^T\end{aligned} \tag{5.35}$$

Because $\mathbf{H}_0$ is a row of ones, post-multiplying a diagonal matrix by $\mathbf{H}_0^T$ has the effect of "undoing" the diagonalization,

$$\begin{aligned}\mathbf{H}_0\mathbf{Q}(t)\mathbf{H}_0^T &= \mathbf{H}_0\mathbf{A}(t)\underline{N}(t) - \mathbf{H}_0\underline{N}(t) \\ &= \mathbf{H}_0\underline{N}(t) - \mathbf{H}_0\underline{N}(t) \\ &= 0\end{aligned} \tag{5.36}$$

A similar result can be obtained for $\tilde{\mathbf{Q}}$, because nowhere in (5.35) or (5.36) was the value of $\underline{N}(t)$ important. Assuming that there are N^{Total} agents in each state therefore has no effect on the validity of the identity.

5.4 Properties of the Kalman filter for recruitment systems

Now that we have a good idea of how to construct models for estimation covariance update and for output covariance prediction, we can assemble a Kalman filter to

observe the recruiment system:

$$\begin{aligned}
\mathbf{P}(t|t-1) &= \mathbf{A}(t)\mathbf{P}(t-1)\mathbf{A}(t)^T + \tilde{\mathbf{Q}}(t) \qquad (5.37)\\
\mathbf{S}(t) &= \mathbf{H}\mathbf{P}(t|t-1)\mathbf{H}^T + \tilde{\mathbf{R}}(t)\\
\mathbf{K}(t) &= \mathbf{P}(t|t-1)\mathbf{H}^T\mathbf{S}(t)^{-1}\\
\underline{\hat{N}}(t) &= [\mathbf{I} - \mathbf{K}(t)\mathbf{H}]\,\mathbf{A}(t)\underline{\hat{N}}(t-1) + \mathbf{K}(t)\underline{Y}(t)\\
\mathbf{P}(t) &= [\mathbf{I} - \mathbf{K}(t)\mathbf{H}]\,\mathbf{P}(t|t-1)
\end{aligned}$$

The properties of this filter must now be examined to ascertain whether or not the filter is adequate to the task of producing estimates with bounded covariance.

5.4.1 Underlying Covariance Bounds

In general, LTV systems with additive noise are not bounded in either the first or the second moment. However, the underlying recruitment process is guaranteed to be bounded, both in terms of $\underline{N}(t)$ and in terms of $\underline{Y}(t)$. There are a finite number of agents, each with states having bounded outputs. The summed distribution has compact support. Consequently, we would expect that any estimator should output a prediction with bounded covariance. For example, if a constant state transition matrix were applied to the graph, it would be possible to arrive at a reliable *a priori* prediction of $\underline{N}(t)$ based on the right-hand eigenvalues of $\mathbf{A}(t)$, as was shown in §3.4. Because $\mathbf{A}(t)$ is a matrix of nonnegative elements whose columns sum to one, the Perron-Frobenius theorem guarantees that the equilibrium probability of being in any state can be represented by the elements of $\underline{u}_1$,

$$\mathbf{A}\underline{u}_1 = \underline{u}_1 \qquad (5.38)$$

$$\lim_{t\to\infty} \Pr\{s(t) = i\} = u_{1i} \qquad (5.39)$$

The expected state distribution and covariance can again be calculated:

$$\lim_{t\to\infty} \mathrm{E}\{\underline{N}(t)|\ \underline{x}(t), \mathbf{A}\} = N^{Total}\underline{u} \tag{5.40}$$

$$\lim_{t\to\infty} \mathrm{Cov}\{\underline{N}(t)|\ \underline{x}(t), \mathbf{A}\} = N^{Total}u_{1i}\left(\delta_{ij} - u_{1j}\right) \tag{5.41}$$

These expressions are clearly bounded, as each element of $\underline{u}_1$ is bounded. It would be reassuring if the Kalman filter formulation of this problem preserved the boundedness of the estimate, in terms of first and second moments, as well as the overall scaling relationship that both are proportional to N^{Total}. Fortunately, both of these properties can be demonstrated for a system with a constant state transition graph.

5.4.2 Examining the Convergence Behavior of the Filter

We would like to show that the moment-baased model converges in the absence of any *a posteriori* information to the same expected value as the underlying, intrinsically bounded random process. Further, we would like to show that the covariance converges to a bounded value. It is acceptable to converge to a bounded value that is larger than the value found in (5.41), but it must exhibit the same scaling relationship, that is, each element of $\mathbf{P}$ must scale with N^{Total} in the steady state. The *a priori* moment-based model predicts that the expected value of $\underline{N}(t)$ should evolve according to the linear dynamics

$$\mathrm{E}\{\underline{N}(t+1)\} = \mathbf{A}\underline{N}(t) \tag{5.42}$$

Clearly, the equilibrium value of $\underline{N}$ is equal to the value obtained in (5.40),

$$\underline{N}(ss) = \mathbf{A}N^{Total}\underline{u}_1 = N^{Total}\underline{u}_1 \tag{5.43}$$

Finding the equilibrium covariance is a little more complicated. In (5.22) above, the time evolution of the *a priori* estimation covariance was found to be:

$$\mathbf{P}(t+1) = \mathbf{A}(t)\mathbf{P}(t)\mathbf{A}(t)^T + \tilde{\mathbf{Q}} \tag{5.44}$$

The standard method for finding equilibrium values of a Ricatti equation (this is a special, simplified case of the Riccati equation) is to use a fractional decomposition, as in §4.9.2 of Grewal and Andrews [17],

$$\mathbf{P}(t|t-1) = \mathbf{U}(t)\mathbf{V}(t)^{-1} \tag{5.45}$$

Under this decomposition, the time evolution of $\mathbf{P}$ can be written as a linear equation,

$$\begin{aligned} \begin{bmatrix} \mathbf{U}(t+1) \\ \mathbf{V}(t+1) \end{bmatrix} &= \mathbf{\Psi}(t) \begin{bmatrix} \mathbf{U}(t) \\ \mathbf{V}(t) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A} & \tilde{\mathbf{Q}}\mathbf{A}^{-T} \\ 0 & \mathbf{A}^{-T} \end{bmatrix} \begin{bmatrix} \mathbf{U}(t) \\ \mathbf{V}(t) \end{bmatrix} \end{aligned} \tag{5.46}$$

The values of $\mathbf{U}$ and $\mathbf{V}$ that produce a steady state value of $\mathbf{P}$ are those satisfying the eigenvalue problem:

$$\mathbf{\Psi}(t) \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix} = \begin{bmatrix} \mathbf{UD} \\ \mathbf{VD} \end{bmatrix} \tag{5.47}$$

Here $\mathbf{D}$ is a diagonal matrix of eigenvalues. The complete solution to this eigenvalue problem is difficult, but we can easily determine a value for $\mathbf{D}$, because $\mathbf{\Psi}$ is in block diagonal form,

$$\mathbf{AU} + \tilde{\mathbf{Q}}\mathbf{A}^{-T}\mathbf{V} = \mathbf{UD} \tag{5.48}$$

$$\mathbf{A}^{-T}\mathbf{V} = \mathbf{VD} \tag{5.49}$$

It is clear from (5.49) that $\mathbf{V}$ is related to the eigenvalues of $\mathbf{A}$, and that $\mathbf{D}$ is related to the eigenvectors of $\mathbf{A}$. In fact, we can show that $\mathbf{V}$ is the matrix of right-handed

eigenvectors of $\mathbf{A}$, and that $\mathbf{D}$ is equal to the inverse of the corresponding eigenvalues:

$$\begin{aligned}
\mathbf{A} &= \mathbf{\Theta}\mathbf{\Lambda}\mathbf{\Theta}^{-1} \\
\mathbf{A}^{-1} &= \mathbf{\Theta}\mathbf{\Lambda}^{-1}\mathbf{\Theta}^{-1} \\
\mathbf{A}^{-T} &= \mathbf{\Theta}^{-T}\mathbf{\Lambda}^{-1}\mathbf{\Theta}^{T} \\
\mathbf{A}^{-T}\mathbf{\Theta}^{-T} &= \mathbf{\Theta}^{-T}\mathbf{\Lambda} \\
\mathbf{V} &= \mathbf{\Theta}^{-T} \\
\mathbf{D} &= \mathbf{\Lambda}^{-1}
\end{aligned}$$

Motivated by this observation, we will express $\mathbf{P}$ in a different set of coordinates, corresponding to the eigenvectors of $\mathbf{A}$,

$$\mathbf{\Gamma}(t) = \mathbf{\Theta}^{-1}\mathbf{P}(t)\mathbf{\Theta}^{-T} \tag{5.50}$$

On an element-by-element basis, this new matrix $\mathbf{\Gamma}$ can be written in terms of pre- and post-multipliction of $\mathbf{P}$ by right-handed eigenvectors of $\mathbf{A}$,

$$\underline{e}_i\mathbf{A} = \lambda_i\underline{e}_i$$

$$\Gamma_{ij}(t) = \underline{e}_i^T\mathbf{P}(t)\underline{e}_j \tag{5.51}$$

The time evolution of any element of $\mathbf{\Gamma}$ can be traced by pre- and post-multiplying (5.22) by $\underline{e}_i$ and $\underline{e}_j$,

$$\mathbf{P}(t+1) = \mathbf{A}\mathbf{P}(t)\mathbf{A}^T + \tilde{\mathbf{Q}} \tag{5.52}$$

$$\underline{e}_i^T\mathbf{P}(t+1)\underline{e}_j = \underline{e}_i^T\mathbf{A}\mathbf{P}(t)\mathbf{A}^T\underline{e}_j + \underline{e}_i^T\tilde{\mathbf{Q}}\underline{e}_j \tag{5.53}$$

$$= \lambda_i\underline{e}_i^T\mathbf{P}(t)\underline{e}_j\lambda_j + \underline{e}_i^T\tilde{\mathbf{Q}}\underline{e}_j \tag{5.54}$$

$$\Gamma_{ij}(t+1) = \lambda_i\lambda_j\Gamma_{ij}(t) + \underline{e}_i^T\tilde{\mathbf{Q}}\underline{e}_j \tag{5.55}$$

Thus, this change of coordinates decouples the time evolution of each element of $\mathbf{\Gamma}$. If each element converges to a bounded value in steady state, then $\mathbf{P}$ must also be

bounded. The eigenvalues of $\mathbf{A}$ are all between -1 and 1. The largest eigenvalue, λ_1, is equal to one, and unique by the Perron-Frobenius theorem [15]. Thus, for all elements except Γ_{11}, the recursive sequence of values converges to some steady state,

$$\lim_{t\to\infty} \Gamma_{ij}(t) = \frac{\underline{e}_i^T \tilde{\mathbf{Q}} \underline{e}_j}{1 - \lambda_i \lambda_j} \tag{5.56}$$

For the special case element Γ_{11}, this does not hold. However, The identities from (5.35) turns out to be quite useful. Remember that the left-handed eigenvector corresponding to an eigenvalue of 1 is equal to a column of ones. Using this, we can rewrite the time evolution of Γ_{11}:

$$\begin{aligned} \Gamma_{11}(t+1) &= 1 \cdot 1 \cdot \Gamma_{11}(t) + \underline{e}_1^T \tilde{\mathbf{Q}} \underline{e}_1 \\ &= \Gamma_{11}(t) + \mathbf{H}_0 \tilde{\mathbf{Q}} \mathbf{H}_0^T \\ &= \Gamma_{11}(t) + 0 \end{aligned} \tag{5.57}$$

Thus, the value of Γ_{11} is not affected by time evolution. The corresponding intuitive statement here is that the uncertainty in the total number of agents does not change with time, and is consequently equal to the initial uncertainty. If the initial variance is bounded, then it is bounded for all time.

5.4.3 In Summary

In this section, we have shown that under a few modeling approximations, the recruitment process can be represented in a form that can be observed using the Kalman filtering framework. The information needed to construct the filter is not difficult to obtain, consisting mainly of empirical variance data from the physical system. In the latter half of this chapter, we will show how this can be applied to improve the performance of real systems.

5.5 Experimental Example

5.5.1 Decreasing Sampling Time

One important abstraction that was adhered to in chapters 3 and 4 is the discrete-state abstraction. This guaranteed that the force or displacement produced by each motor unit is solely a function of the discrete agent's state. Figure 5-4 shows this simple two-state model. This approach sacrifices speed and responsiveness for simplicity. In order to credibly make the assumption that the output of the SMA element is determined by the discrete state of the agent, the sampling interval for measuring the ensemble output and issuing commands must be long enough to allow any transient continuous-state behavior to settle down, as we discussed in §3.2.1, and illustrated in Fig. 3-3. Because of the discrete-state abstraction, all of the performance metrics given in the previous chapters are benchmarked against the characteristic response time of the local continuous-state behaviors. For example, all of the expected convergence times computed in §4.5.5 had a hard minimum value of 1 outside of the target set, as was shown in Fig. 4-14.

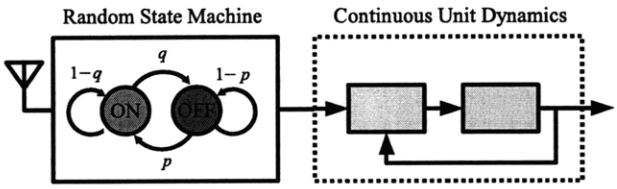

Figure 5-4: The hybrid dynamics imposed on a single unit consist of a random finite state machine that changes the command given to an SMA unit, whose continuous-state dynamics are assumed to settle after some transient.

It seems quite reasonable to object to this as a hard constraint on the performance of the control system. In principle, it should be possible to respond immediately to a measured disturbance, without having to wait for the output to settle. However, simply decreasing the sampling interval, with no other changes to the model or control law, is disastrous. If this is done, the algebraically determined state estimates from (4.1) do not account for the transient response. The classical control analogy here would be as if there were an unmodeled real zero in the plant one was trying to

control; this zero would become an additional pole in the system response which could hurt performance, or simply make the system unstable. Figures 5-5 and 5-6 show the result of taking the 60 unit parallel SMA actuator and attempting to control isometric force, having cut the sampling time from 2 seconds to 0.5 seconds without making any other alterations. The natural result is that the system overshoots its desired target. settling only after a long, oscillatory decay. The control laws are still convergent in probability in this case – there is still a chance that the system output could pause long enough on the desired value, as demonstrated in the figures. However, performance is very degraded and practical tracking may be useless.

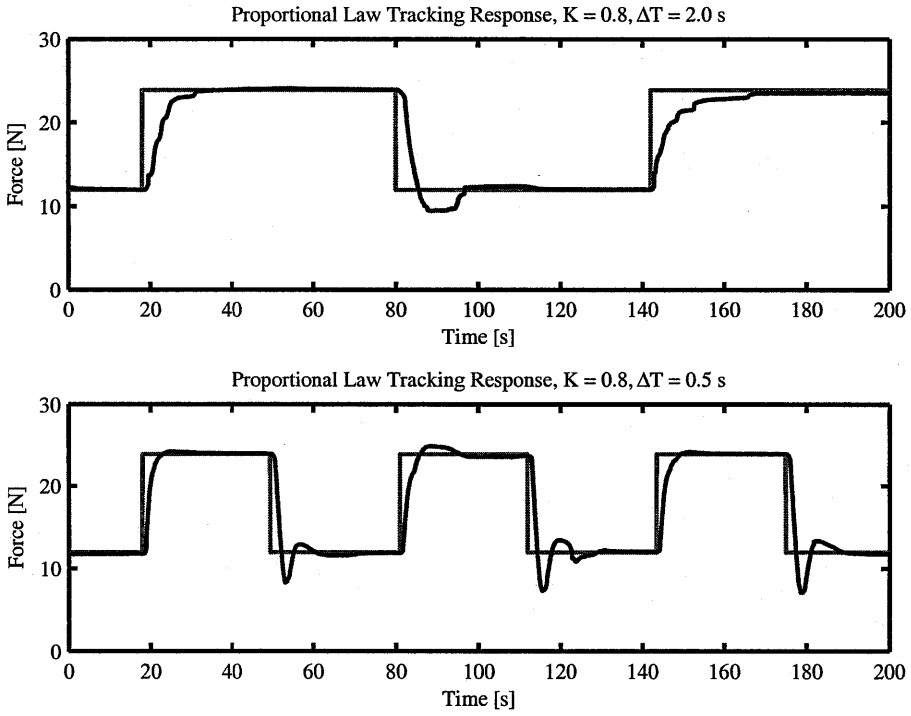

Figure 5-5: The tracking results of a proportional law (top) when the sampling time is chosen to agree with the transient settling behavior of the SMA units, and (bottom) when the sampling time is much less than the transient settling time of the SMA units.

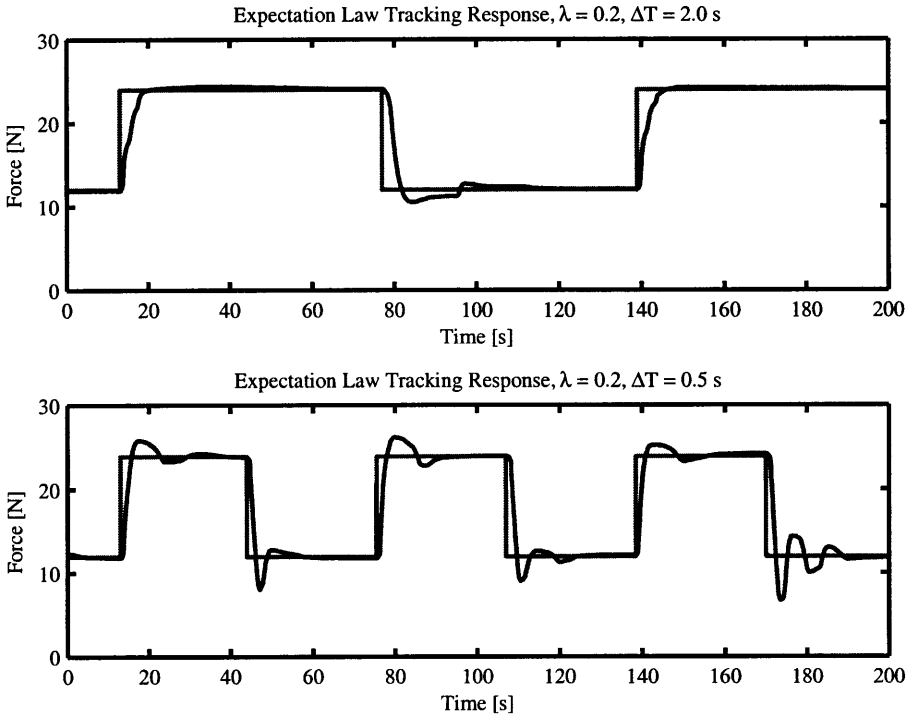

Figure 5-6: The tracking results of a moment-based law (top) when the sampling time is chosen to agree with the transient settling behavior of the SMA units, and (bottom) when the sampling time is much less than the transient settling time of the SMA units.

5.5.2 State Augmentation

There is a way to improve upon the behavior of the two-state model, if the continuous-state output of the active material units during state transitions is repeatable. If this is the case, then one could subdivide the discrete time intervals on which the control system operates, assigning output values at the intermediate time intervals based on measurements of the transient behavior, as shown in Fig. 5-7. These intermediate output values could then be assigned to augmented delay states, depicted in Fig. 5-8. A Kalman filter could then be used to keep track of the number of units that have recently made state transitions, thus compensating for their partial output. One potential difficulty lies in the behavior of units which transition from *off* to *on* and then rapidly back to *off* again, within the settling time of the unit. In this case, some kind of partial transition model would be needed, corresponding to the graph edges leaving the intermediate states in Fig. 5-8. Some combination of carefully choosing the destination of these graph edges and tuning the output model should be able to adequately approximate rapid switching behavior.

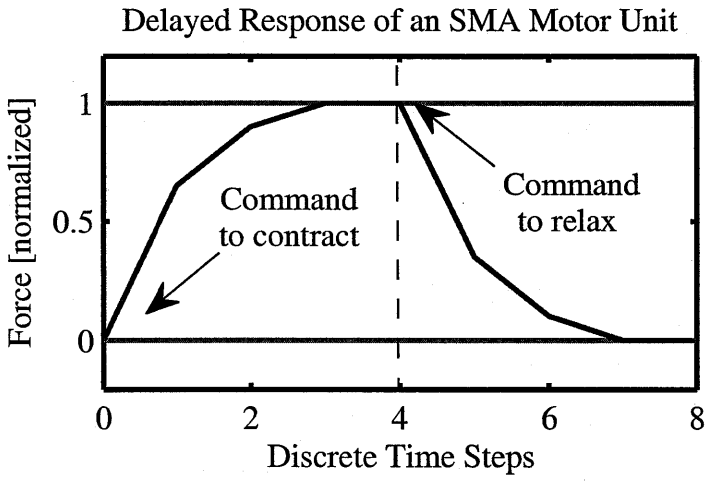

Figure 5-7: An example of unmodeled continuous dynamics. A controller that assumes that each motor unit has two states will not account for the physical delay associated with activating the SMA.

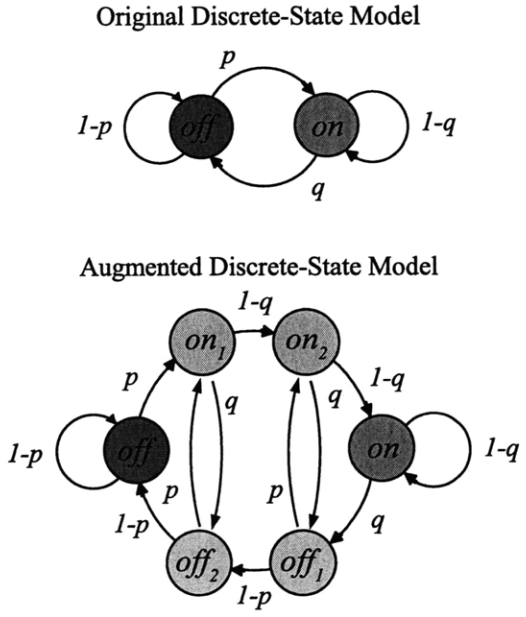

Figure 5-8: One way of addressing unmodeled dynamics is to introduce additional states into each agent that can be used to produce a more fine-grained model of the agent's continuous-state behavior. Notice that units are free to turn *on* or *off* with exactly the same probability; the only system behavior that changes is the value of the predicted transient output.

5.5.3 Control Law Modifications

Because the modified system state transition graph still has only two parameters, $p(t)$ and $q(t)$, the control law used to take advantage of the finer-grained dynamic model is relatively simple. One strategy is to look ahead at past the transient dynamics at what will happen if no state transitions are commanded. The augmented state transition matrix corresponding to the graph in Fig. 5-8 is shown as an example:

$$\mathbf{A}(t) = \begin{bmatrix} 1-p(t) & 0 & 0 & 0 & 0 & 1-p(t) \\ p(t) & 0 & 0 & 0 & 0 & p(t) \\ 0 & 1-q(t) & 0 & 0 & p(t) & 0 \\ 0 & 0 & 1-q(t) & 1-q(t) & 0 & 0 \\ 0 & 0 & q(t) & q(t) & 0 & 0 \\ 0 & q(t) & 0 & 0 & 1-p(t) & 0 \end{bmatrix} \tag{5.58}$$

If no state transitions are commanded, i.e. $p(t) = q(t) = 0$, then the only state transitions are due to the delay states. This state transition matrix will be called $\mathbf{A}_0$,

$$\mathbf{A}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.59}$$

From any state, all of the transients will have settled after two time steps. Motivated by this observation, $\mathbf{A}(t)$ could be chosen so that the state distribution, once settled, matches the desired distribution,

$$\mathbf{A}_0^2 \mathbf{A}(t) \underline{N}(t) = \underline{N}^{ref} \tag{5.60}$$

This strategy produces very stable control laws. Another strategy which yields faster but less stable performance is the "overdrive" strategy, choosing $\mathbf{A}(t)$ so that the one-step-ahead output is equal to the desired output,

$$\mathbf{H}\mathbf{A}(t)\underline{N}(t) = \underline{Y}^{ref} \tag{5.61}$$

Because this policy has a tendency to command more state transitions than needed in order to minimize one-step-ahead error, it may produce unpredictable results if there are pure delays in the system output, such as a unit which takes a moment between when it is activated and when it begins to exert force. It will also be sensitive to output inaccuracy. A solution somewhere between these two would be a control law based on limited lookahead LQR, in the form:

$$\mathbf{A}(t) = \underset{\mathbf{A}(t),\mathbf{A}(t+1),\mathbf{A}(t+2)}{\arg\min} \left(\sum_{k=0}^{2} \left| \mathbf{H} \prod_{j=0}^{k} \mathbf{A}(t+j)\underline{N}(t) - \underline{Y}^{ref} \right|^2 \right) \tag{5.62}$$

The control law chosen for implementation was the lookahead law from (5.60), in the interest of stability.

5.5.4 System Identification

For the experimental actuator, the dual problem of identifying a discrete state transition model and an output model for the transient states was solved by first identifying the outputs from the intermediate transitioning states, assuming that the system was fully settled. The two-state actuator model was augmented to include 9 transient states between *off* and *on*, and 9 more transient states between *on* and *off*, for a total of 20 states. It is possible to activate all of the units at once by sending a command with $p(t) = 1$ and $q(t) = 0$, and vice versa. The output force, shown in Fig. 5-9, was fit to a 1×20 output matrix, assuming that each agent's identical output was contributed to the overall force. Figure 5-10 shows a plot of the output model.

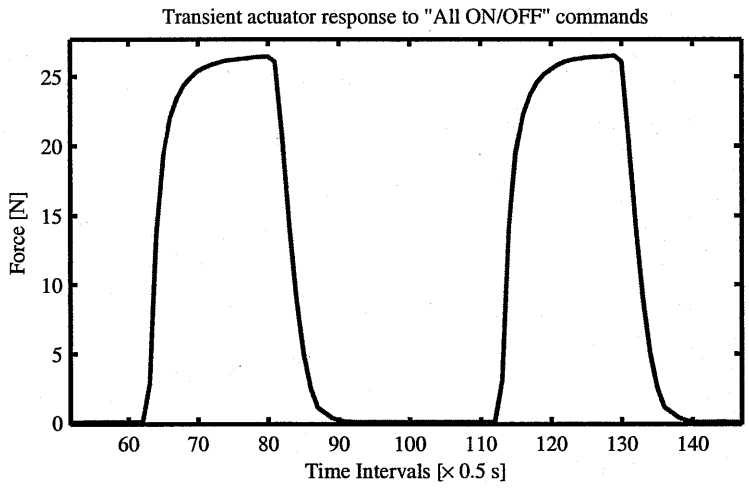

Figure 5-9: This data shows the transient response of the actuator to commands to turn all units *on* or OFF. It was used to train an augmented state transition model with 9 delay states.

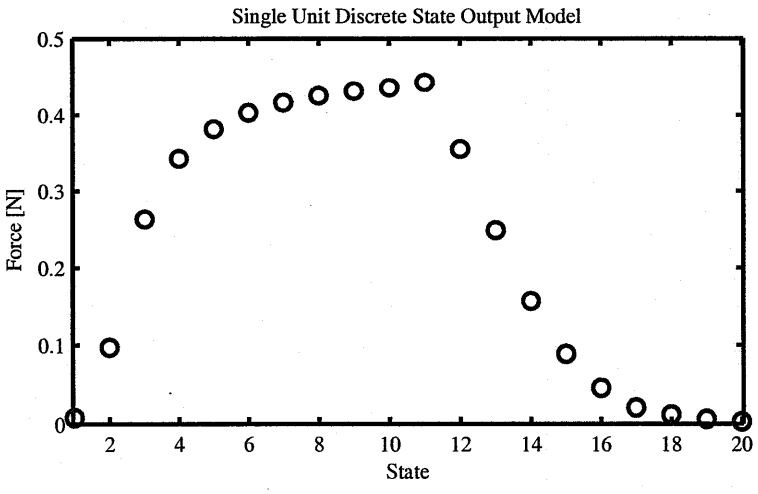

Figure 5-10: The modeled output of a single agent, fit with least squares to the state trajectory inferred from the ON/OFF switching sequence broadcast to the actuator.

5.5.5 Identifying Transient Behavior

Rather than go through a complex, iterative process designed to minimize the number of states needed to most closely fit the output, we settled on an *ad hoc* method of approximating the transient system behavior. The time intervals between commands to turn *on* and commands to turn *off* were varied slowly, so that the whole ensemble of units passed through all of the possible transient states while transitioning. The most appropriate state transition behavior for these partial state transitions was determined by picking the output state minimizing the prediction error. The data from the transient switching is shown in Fig. 5-11, along with the prediction gleaned from the model. The edges obtained are shown in Fig. 5-12. This method of system identification was not by any means optimal. It was chosen because it was simple to construct and simple to reproduce. More in-depth methods of model order choice, and output approximation would definitely yield better results.

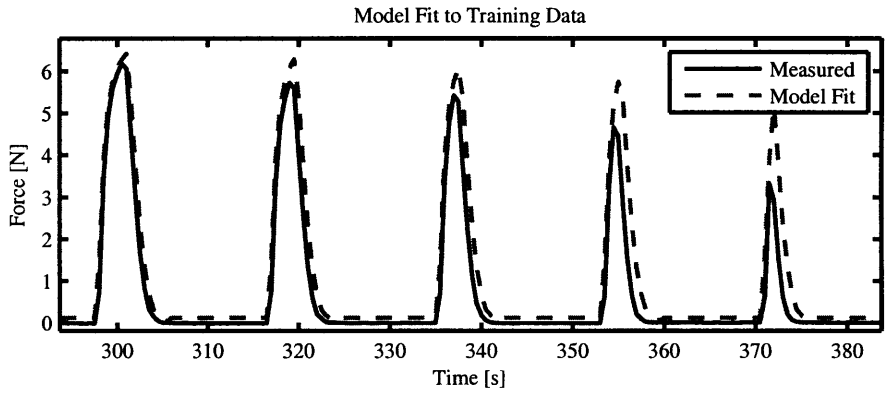

Figure 5-11: The system identification data for the fast transient switching behavior, showing the model prediction.

5.5.6 Observer-Controller Performance

The tracking performance of the observer-controller is shown in Fig. 5-13. While the tracking behavior is not that much faster than the results for the top plot in Fig. 5-6, the ability to respond quickly to disturbances is improved. The bottom plot of Fig. 5-13 is interesting because it shows that the model does not always have the correct estimate of the output, even when the output error is zero. This is much like any

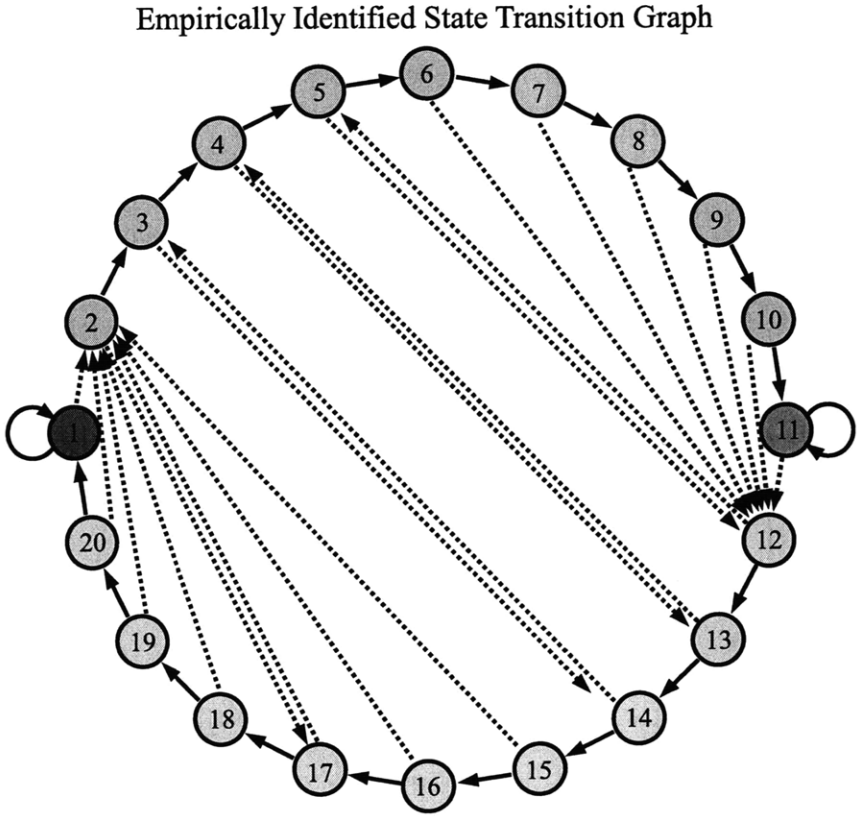

Figure 5-12: This graph was identified by measuring the settling time of the actuator when turned *on* or *off* with controlled delays. The solid lines represent the default behavior of each unit, i.e. the behavior if no state transition is commanded. The dashed lines represent the transition that will be made if a transition is commanded.

other problem in adaptive control. Because the output used for control is the same as the output used for observation, an unobservable error is not a problem because the controller will not respond to it.

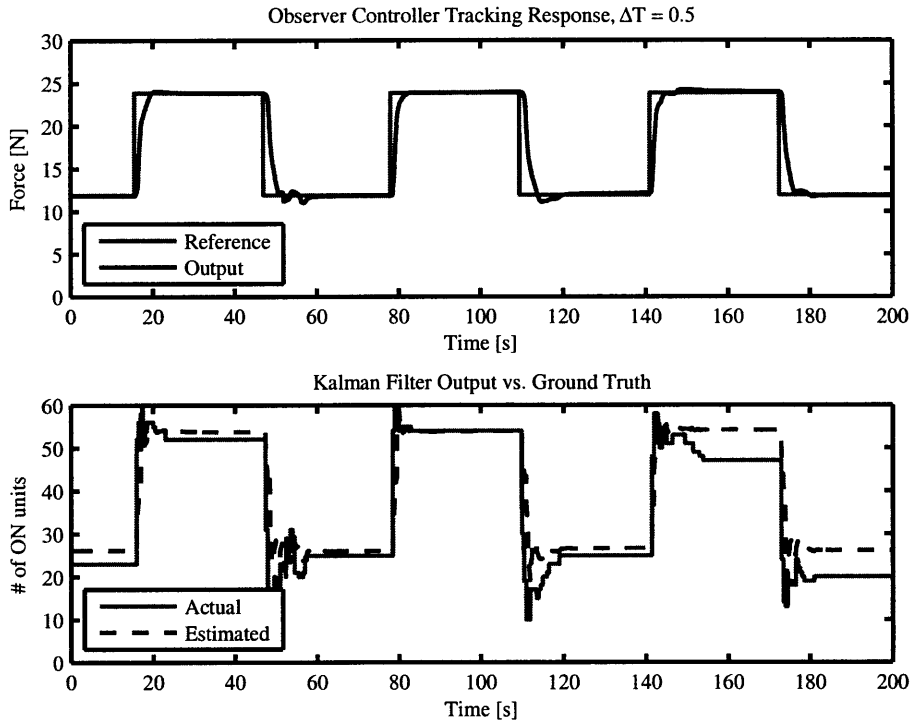

Figure 5-13: The top plot shows the tracking behavior of the observer controller. The bottom plot shows the estimated number of *on* units comared with the actual number of *on* units.

5.6 Conclusion

The example in this section demonstrates that even for a two state actuator, a state observer can be a useful tool for control. For the larger problem of controlling random state machines, the usefulness is much broader. A central controller dispatching tasks among a swarm of robots could be greatly aided by estimates of the robots' state based on partial measurement. A more complex gene regulation problem than the two-state lac operon controller proposed by Julius et Al [27] could be regulated even in the absence of rich outputs. This particular aspect of the work in this thesis is

the newest and most exciting in its potential application to other, similar distributed control problems.

Chapter 6

Application: An Antagonistic, Variable Stiffness Robot Arm

6.1 Introduction

This chapter offers an example demonstrating how recruitment-based actuators can be used to control a robot arm using recruitment control. The physical outputs produced by active materials, as discussed in chapter 2, are not ideal force or displacement sources, because of the material's intrinsic elasticity. In fact, some actuator configurations, such as the parallel configurations discussed in §2.3.2, produce no change in actuator displacement as units are recruited, only a change in stiffness. Producing stable and controllable motion with such an actuator is possible using an approach called equilibrium point (EP) control.

A joint that is actuated by antagonistic, elastic actuators, shown in Fig. 6-1, will have an equilibrium position, that is, a joint angle at which the spring-like forces exerted by the two antagonistic actuators sum to zero. The ability of the joint to resist disturbances is determined by the sum of the stiffnesses of the two actuators, plus whatever active force is exerted using feedback. If the passive stiffnesses and resting lengths of the elastic actuators can be specified, then this could be used as a technique for controlling not only the position of a joint, but also the passive joint stiffness. A variable stiffness robot arm could be used for tasks previously requiring

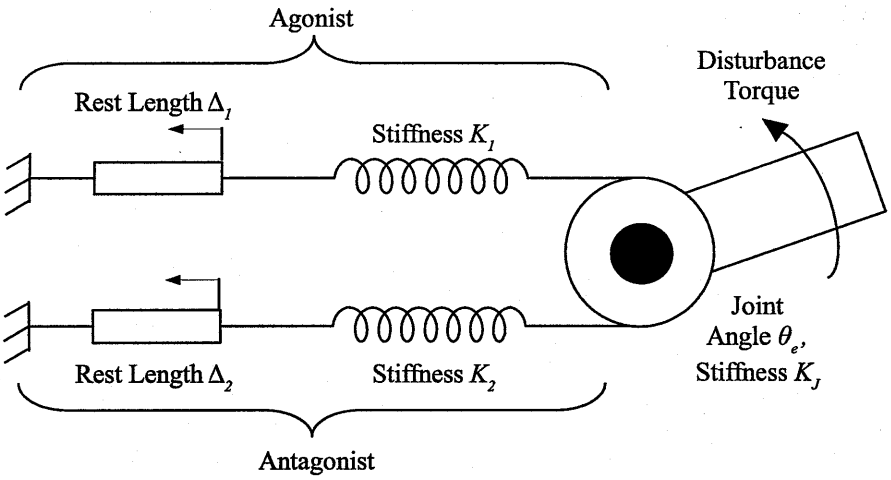

Figure 6-1: A schematic diagram of equilibrium point motor control. Two antagonistic actuators are shown here as equivalent stiffnesses and displacements, which change based on the control inputs. The control inputs to the system vary D_1, D_2, K_1 and K_2. The equilibrium point achieved by the two actuators will be a stable point of attraction for the joint.

active impedance control, or it could be used to alter its passive dynamics, such the resonant frequencies of the arm.

The idea of using equilibrium point control for robot arms has been discussed for several years in the robotics community, and recently several groups have produced actuators that implement equilibrium point control using cams and variable stiffness springs [65], [49]. Hitherto, there has been no report in the literature of any high-resolution, variable stiffness active material actuators capable of equilibrium point control, although the notion of making such an actuator has been discussed by the authors and others [46], [36]. This work is based on an implementation of recruitment-based stiffness modulation in active material actuators. Two actuators made from 60 parallel SMA elements are configured antagonistically about a joint, whose equilibrium point is varied with feedback control. The ability of this control scheme to recover from disturbances quickly and without exciting undersirable closed-loop behaviors is demonstrated.

Section 6.2 gives a brief background on the history and theory behind equilibrium point control in muscles, and a discussion of how EP control can be used to improve

the performance of active material actuators, similar to ways theorized to improve the performance of muscle. Section 6.3 introduces the 1-DOF arm constructed to demonstrate equilibrium point control. The results of the experiment are shown in section 6.4.

6.2 Equilibrium Point Control of a Single-DOF Joint

6.2.1 Overview and History of EP Control

One remarkable feature of the musculo-skeletal system is the use of passive muscle stiffness as a stabilizing mechanism. It is well known that the brain's control over limb motions is accomplished using more than neural feedback loops. Bizzi and Polit reported an experiment in which monkeys could still perform simple pointing tasks in the presence of disturbance forces after their proprioceptive feedback loops were cut at the spinal column [52]. The best explanation for this phenomenon is that the antagonistic muscles pulling on each joint act as tunable springs in equilibrium when the limb is in its desired posture, as shown in Fig. 6-1. The antagonistic muscles can be co-contracted to produce more or less net stiffness on the joint, while remaining in the same equilibrium position. Many variants on the equilibrium point hypothesis have been proposed; a good summary can be found in an article by McIntyre and Bizzi [41].

Equilibrium point hypotheses provide a good explanation for the ability of muscles to resist high-frequency disturbance forces, which the body would be unable to do using muscle reflexes [22]. Biological feedback systems suffer from time delays due to the inherently slow chemical processes governing nervous action and muscular activation. However, the elastic restoring force about the joint's co-contracted equilibrium point does not depend on sensory feedback, and so it can respond more quickly than any reflex mechanism.

6.2.2 Applying EP Control to Active Material Actuators

The same control architecture which explains the improved closed-loop response of muscles can also be used to improve the performance of artificial muscles. Many artificial muscle materials suffer from time delays due to the physics of energy delivery, such as ion diffusion in conducting polymers or thermal phase change energies in shape memory alloys. A closed-loop controller for a single-joint antagonistic actuator, shown in Figure 6-2, will exhibit a great deal of phase lag in the forward path associated with force production, and so the loop gain must be limited to reduce the risk of instability. This will greatly reduce the ability of the control loop to resist high-frequency external disturbances.

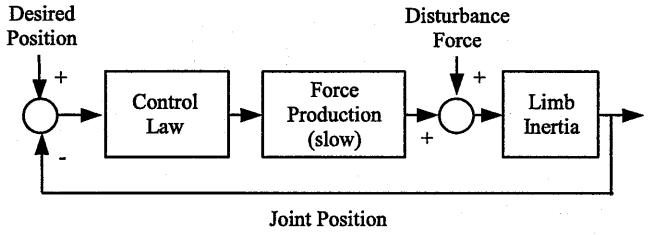

Figure 6-2: A block diagram for a typical SISO controller for simple disturbance rejection. Notice that any delays associated with force production in the actuator are inside the position feedback loop.

Instead of using a standard SISO feedback loop to control the joint position, one could use artificial muscles as tunable springs rather than as force generators to mimic a biological equilibrium point control law. The block diagram in Fig. 6-3 shows a controller that uses the EP framework to specify the equilibrium point of the joint and the stiffness about this point. The time delays associated with activating the active material still affect the rate at which the antagonstic actuator stiffness can be changed. However, this affects only the rate at which the equilibrium point of the actuator system responds to a reference input. The elastic restoring force about the equilibrium point will still respond quickly to any disturbance.

The key to implementing an equilibrium point control law is a mechanism for reliably changing the stiffness of two antagonistic actuators. The models developed in chapter 2 demonstrate that an actuator made up of many parallel units is well

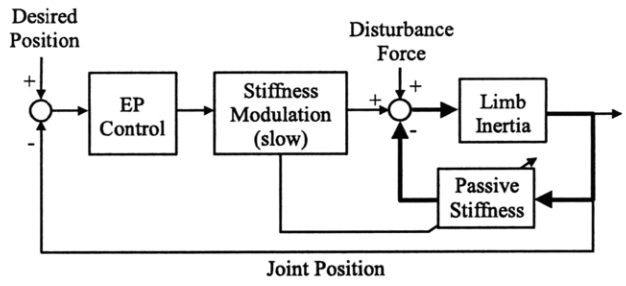

Figure 6-3: A block diagram for an equilibrium point controller. A slow, task-oriented control loop (labeled "EP Control") determines the stiffness and desired equilibrium position of the plant. The passive actuator stiffness then provides a restoring force similar to a proportional feedback loop, shown in bold lines. The time delays associated with stiffness production occur outside this virtual proportional feedback loop.

suited to this task. Figure 6-4 shows a diagram of an artificial muscle actuator made up of many small parallel units, much like a muscle. Each small unit is held in a binary state, either relaxed or contracted. Because the contracted units govern the overall length of the actuator, the relaxed units will slacken and will not contribute to the overall stiffness of the actuator. The stiffness is therefore proportional to the number of active units, $N_{on}(t)$, as previously shown in (2.11).

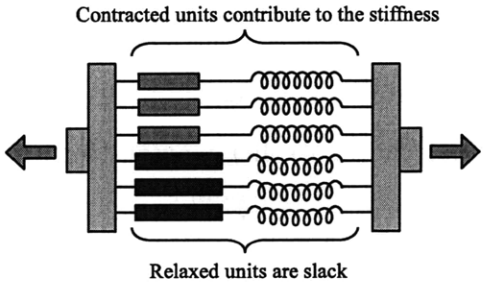

Figure 6-4: This is a figure of a single actuator, composed of many parallel springs. The relaxed springs are long enough that they do not contribute any tension to the actuator output.

6.2.3 Calculating EP Parameters using Recruitment

In order to control the equilibrium position of a joint with antagonistic, variable-stiffness actuators, a controller must first compute the actuator stiffnesses and dis-

placements in the equivalent circuit model that will produce a desired equilibrium angle and joint stiffness. These stiffnesses and displacements can then be expressed in terms of the number of SMA springs in each actuator that must be contracted in order to achieve these properties. Each actuator is modeled as having a rest length Δ and a stiffness K. If the two actuators are attached to the joint at a radius r, then the rest position of the joint θ_e can be calculated by setting the net torque around the joint to zero,

$$\theta_e = \frac{K_1\Delta_1 - K_2\Delta_2}{r(K_1 + K_2)} \tag{6.1}$$

The total stiffness of the joint K_J is also a function of the actuator stiffnesses,

$$K_J = \frac{d\tau}{d\theta} = r^2(K_1 + K_2) \tag{6.2}$$

By manipulating equations (6.1) and (6.2), the joint equilibrium position and the joint stiffness can be independently specified using K_1 and K_2,

$$K_1 = \frac{K_J}{r^2}\frac{\Delta_2 + r\theta_e}{\Delta_1 + \Delta_2} \tag{6.3}$$

$$K_2 = \frac{K_J}{r^2}\frac{\Delta_1 - r\theta_e}{\Delta_1 + \Delta_2} \tag{6.4}$$

These stiffnesses are then modulated by controlling the number of recruited units, using (2.11). In this study, the controller sought to maximize the stiffness of the actuator at the desired equilibrium angle. This was done by always keeping one of the two actuators maximally contracted, adjusting the position by selectively relaxing units on the other actuator. The maximally contracted actuator was determined by separating the joint space down the middle at the point when the two actuators are maximally contracted,

$$\theta_{sep} = \frac{K_{1,max}\Delta_1 - K_{2,max}\Delta_2}{r(K_{1,max} + K_{2,max})} \tag{6.5}$$

If the desired equilibrium point θ_{ref} is greater than this separating angle, then K_1 is maximized while K_2 is modulated by relaxing some units,

$$K_{2,ref}(t) = K_1(t)\frac{\Delta_1 - r\theta_{ref}}{\Delta_2 + r\theta_{ref}} \tag{6.6}$$

The feedback law is in the form of (4.26), using the current stiffness divided by the maximum stiffness as an estimate of the number of contracted units,

$$p(t) = \begin{cases} (1-\alpha)\frac{K_{2,ref}(t)-K_2(t)}{K_{2,max}-K_2(t)}, & K_2(t) < K_{2,ref}(t) - \epsilon \\ 0, & K_2(t) \geq K_{2,ref}(t) - \epsilon \end{cases} \tag{6.7}$$

$$q(t) = \begin{cases} (1-\alpha)\frac{K_2(t)-K_{2,ref}(t)}{K_2(t)}, & K_2(t) > K_{2,ref}(t) + \epsilon \\ 0, & K_2(t) \leq K_{2,ref}(t) + \epsilon \end{cases}$$

6.2.4 Adaptive Compensation for Slow Disturbances

The equilibrium joint angle tracking can be calculated in real time because the force $F(t)$ and displacement $D(t)$ of each actuator can be measured,

$$\begin{aligned} K_1(t) &= F_1(t)/(D_1(t) - \Delta_1) \\ K_1(t) &= F_2(t)/(D_2(t) - \Delta_2) \\ \theta_e(t) &= \frac{K_1\Delta_1 - K_2\Delta_2}{r(K_1 + K_2)} \end{aligned} \tag{6.8}$$

However, knowing this angle does not mean that the actual measured joint angle will be equal to the equilibrium angle, because of disturbance forces such as gravity. One technique which may also be useful for dealing with slow or constant disturbances might be a slow estimator which tracks the difference $\tilde{\theta}(t)$ between the actual equilibrium joint angle $\theta_e(t)$, which can be calculated by measuring the actuator stiffnesses *in situ*, and the measured joint angle, $\theta(t)$. By subtracting $\tilde{\theta}(t)$ to the reference equilibrium joint angle, the equilibrium point can be shifted to a feed-forward value $\theta'_{ref}(t)$ that reduces the output error. An estimator for this purpose could be implemented

using an exponentially weighted moving average with time constant ρ,

$$\tilde{\theta}(t+1) = (1-\rho)\tilde{\theta}(t) + \rho(\theta(t) - \theta_e(t)) \tag{6.9}$$

$$\theta'_{ref}(t) = \theta_{ref}(t) + \tilde{\theta}(t) \tag{6.10}$$

If the time constant of the estimator is chosen to be equal to or longer than the physical time constant of the actuator response, then this estimator does not hamper the high frequency disturbance rejection discussed above.

6.3 Apparatus Description

Figure 6-5 shows a photograph and a section view of the robot arm constructed for this experiment. It contains two shape memory alloy actuators, each having an array of 60 SMA units placed in parallel, each controlled with a single FET for Joule heating using a microcontroller implementation of the recruitment architecture. Each actuator consists of three stacked circuit boards, which slide back and forth in channels within the arm. The actuators are cooled by a fan, which blows air through a duct in the center of the actuator. The walls of the duct have slits which blow the air through the array of SMA units, as shown in the section view. The force of each actuator was measured with a load cell near the shoulder end of the arm. The angle of the elbow joint was measured with a potentiometer. The total range of motion in the arm is 1.8 radians.

The neutral positions of each actuator were calculated *in situ* by activating all of the units on the antagonistic actuators, and perturbing the arm. The plot of actuator torque versus displacement, shown in Fig. 6-6, was fit to a linear model, whose intersection with the joint angle axis was taken to be the actuator's neutral point. The force-displacement relationship of the actuator was measured in Fig. 6-7. Just as the models predicted, the actuator stiffness varies with the number of recruited units. The model of agonist and antagonist actuator stiffnesses were used to estimate the number of *on* units in each actuator by averaging the force and displacement from

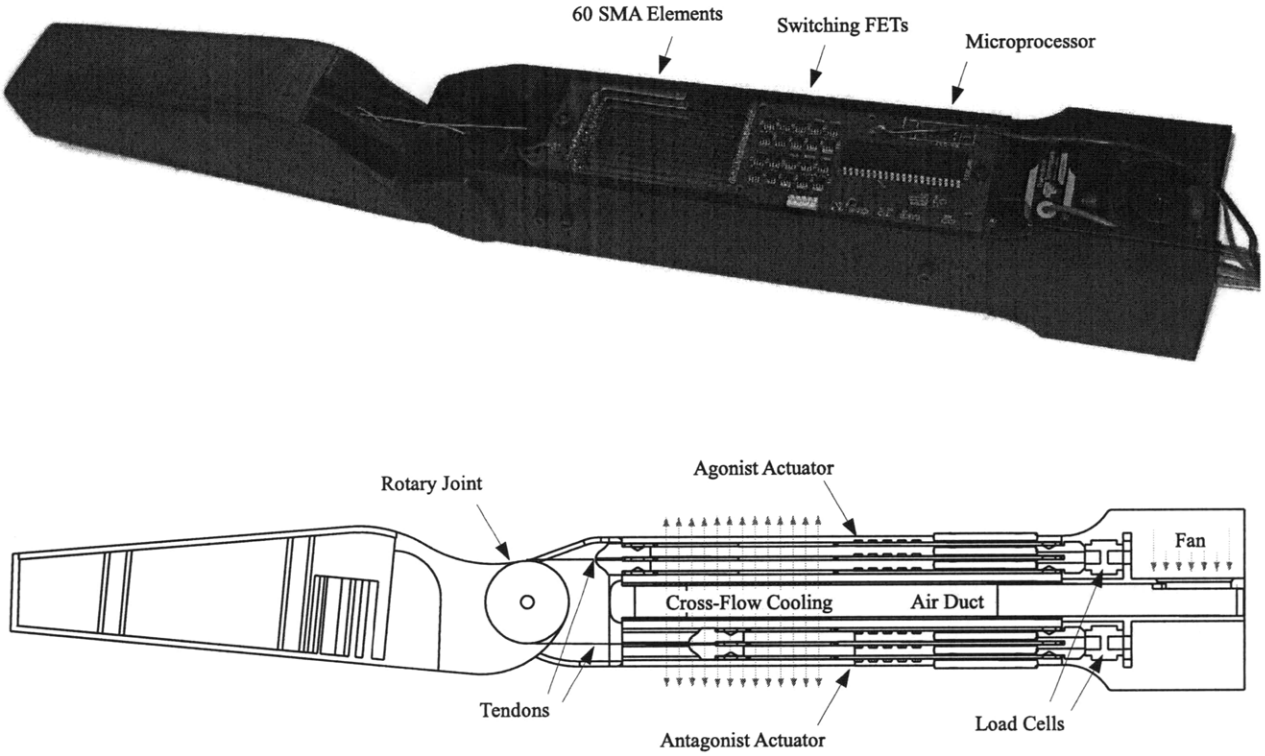

Figure 6-5: A schematic of the antagonistic recruitment-based 1-DOF arm.

the calibrated neutral position of each actuator over a group of n samples,

$$\hat{K}_i(t) = \frac{\sum_{k=0}^{n}(D_i(t-k) - \delta_i)F_i(t)}{\sum_{k=0}^{n}(D_i(t-k) - \delta_i)^2} \tag{6.11}$$

These values were then used with the control law from (6.6) and (6.7) to control the equilibrium position. An estimator was implemented based on (6.9) and (6.10) to compensate for slow disturbances.

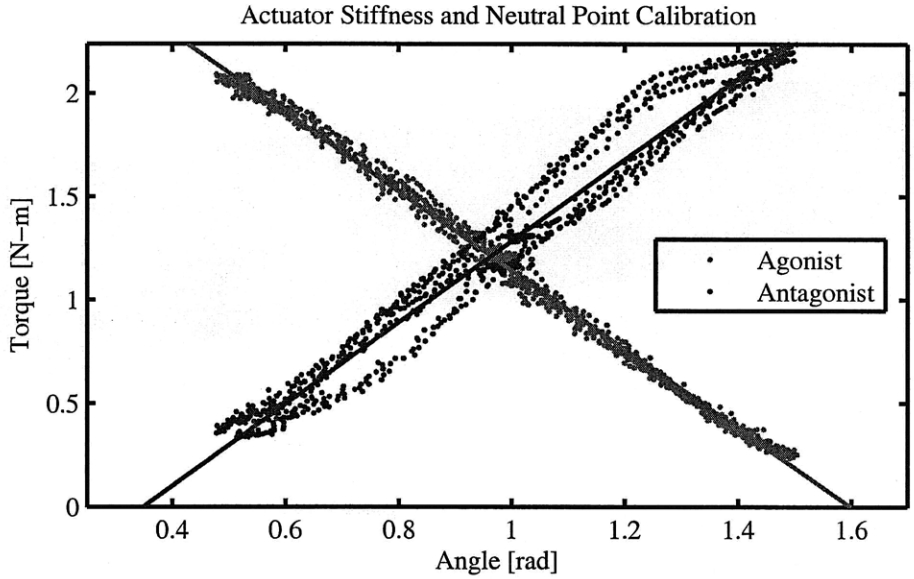

Figure 6-6: Each actuator's torque as a function of the joint angle, with all units in the *on* state.

6.4 Experiment

6.4.1 Equilibrium Point Control

For the first experiment, the arm was clamped in the horizontal plane, as shown in Fig. 6-8. A reference equilibrium position was sent to the central controller, which then used the force sensor data and the joint angle of the arm to drive the equilibrium point to the desired value, without any adaptive compensation for constant disturbance forces. Figure 6-9 shows the results of the equilibrium point tracking the desired position. As predicted, the tracking response is slow because it is limited by the

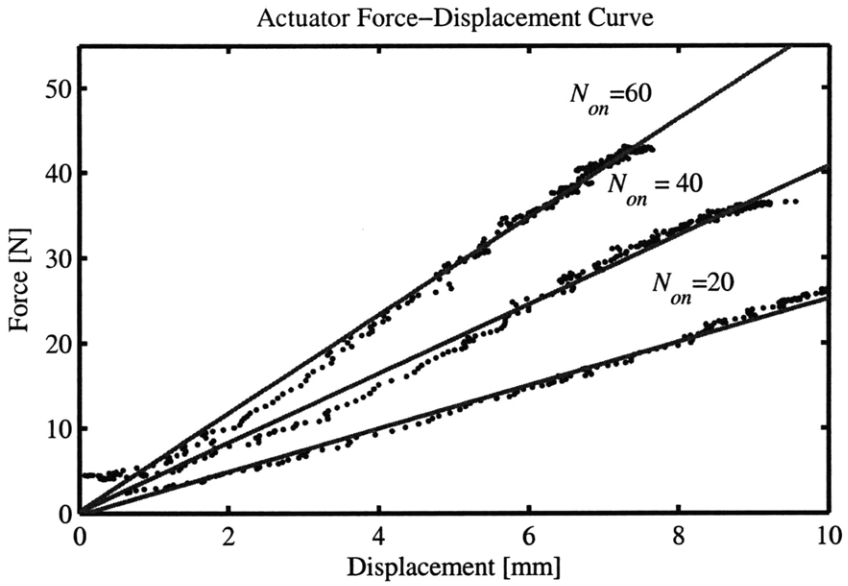

Figure 6-7: The stiffness of a single actuator as a function of number of recruited units is shown for N_{on} = 20, 40 and 60 units.

activation speed of the SMA. The difference between the equilibrium point angle and the actual joint angle is due to an unmodeled disturbance force. The overshoot exhibited by the feedback control law is a result of using the measured stiffness of the completely controlled actuator to determine the commands to be sent to the other actuator, as in (6.6). This solution was chosen because it was very stable and robust to modeling errors introduced by friction between the actuator circuit boards and the channels they slide in. However, because commands sent to the arm are based on tracking a moving target (a reference proportional to the changing stiffness of the other arm), some overshoot was inevitable. A better law would coordinate the two actuators so that their tracking was decoupled, at least when a large change in the reference angle is commanded.

6.4.2 Frequency Response

To determine the frequency response of the closed-loop system, a slow chirp signal was fed into the arm as a joint angle reference. The Bode plot of the actuator's joint

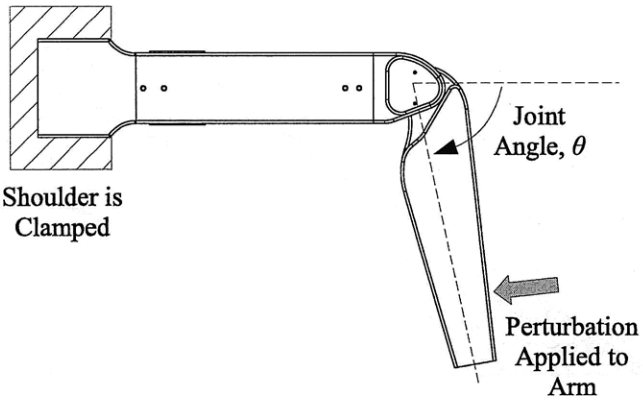

Figure 6-8: The equilibrium point of the arm was controlled while the arm was clamped at the shoulder in the horizontal plane. A perturbation was applied to the end of the arm.

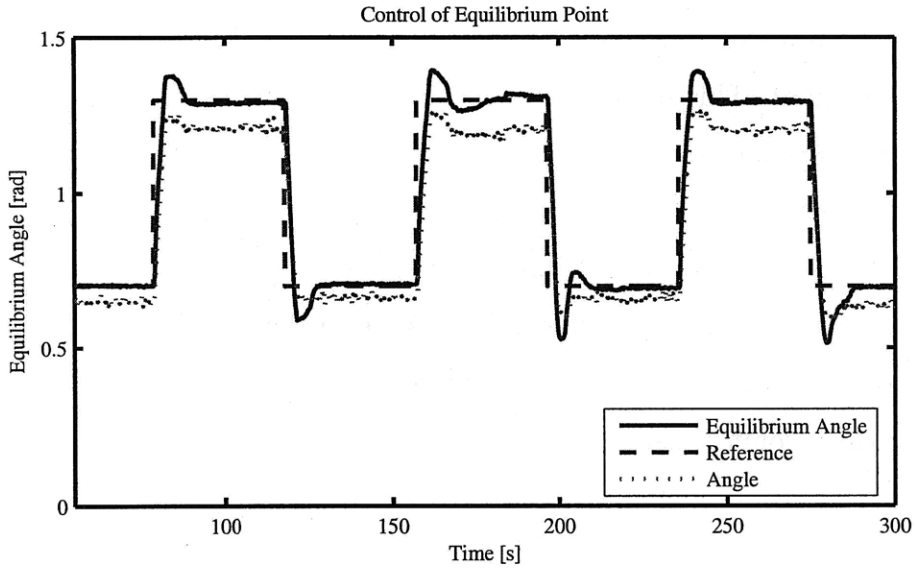

Figure 6-9: This plot demonstrates the ability to control the equilibrium point of the arm joint. The actual angle differs from the equilibrium point due to a small disturbance force.

angle to a reference is shown in Figure 6-10. The bandwidth[1] of the actuator was found to be 0.126 Hz. This seems slow, but it is commensurate with the physical response of the SMA used in the actuator. Consider, for example, that the open-loop force response of the actuator took approximately 5 seconds to settle in the experiment performed in Chapter 5. The fact that the actuator was able to produce a sinusoidal reponse on the order of 10 seconds means that the control system produced an output at remarkably close to the physical response time of the active material. The phase of the actuator's tracking response drops off linearly with frequency. This fact is attributable to the pure time delays associated with activation. The phase was measured by calculating the lag between each rising and falling zero crossing in the reference and the subsequent crossing in the response. The result, plotted in Fig. 6-11, shows that the time delay is equal to about 2 seconds, largely independent of frequency.

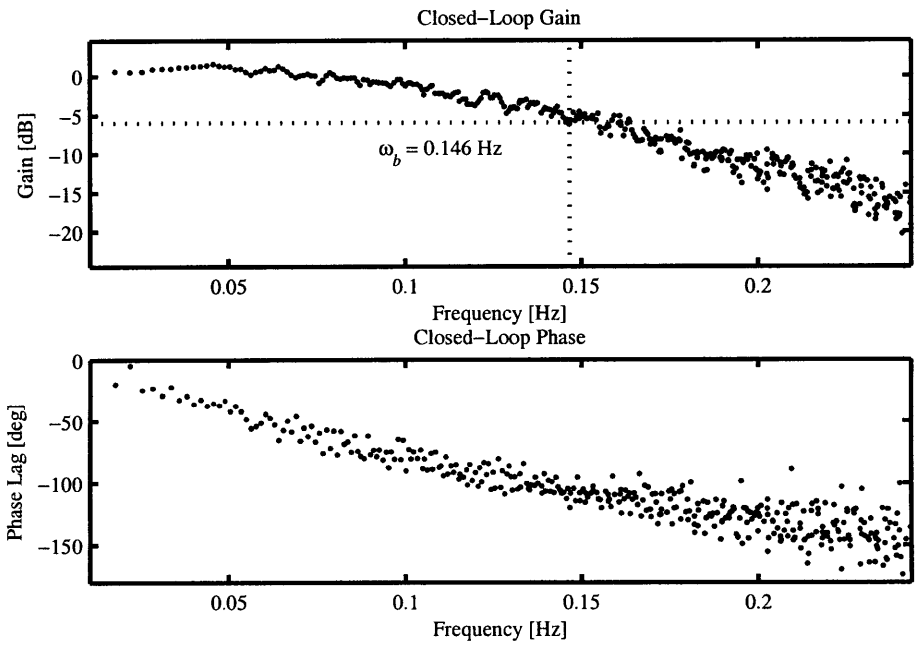

Figure 6-10: A Bode plot of the arm's closed-loop response to a frequency sweep.

[1]The bandwidth was calculated as the frequency for which the response amplitude was one half of the reference.

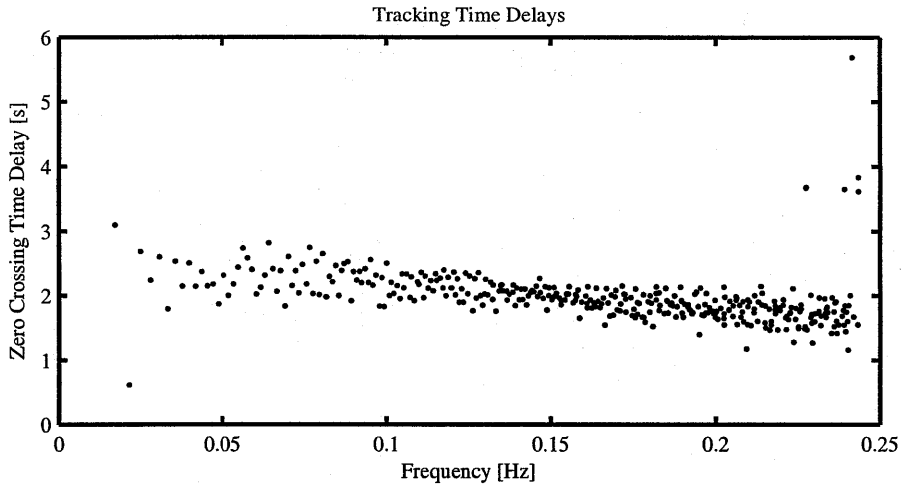

Figure 6-11: The delay between zero crossings for the sinusoidal reference joint angle and the closed loop response.

6.4.3 Fast Disturbance Rejection

The advantage of equilibrium point control can be seen when a peturbation is applied to the actuator. Rather than attempting to activate more units in respose to a change in joint angle, the controller does not see the change in angle as a change in the equilibrium position of the arm, so it does nothing. When the disturbance is removed, the arm springs back quickly to its original position, as shown in Fig. 6-12. The characteristic time for activation of the SMA wire is about 5 seconds; however, the actuator returns to its unperturbed position in 0.21-0.26 seconds.

6.4.4 Adaptive Compensation for Slow Disturbances

To demonstrate the ability of this control system to track a joint angle accurately in the face of disturbances, the arm was held in the vertical plane, and commanded to track a square wave. The estimator for $\tilde{\theta}(t)$ from (6.9) and (6.10) was used to find the offset between the equilibrium point and the actual actuator trajectory. This value was then fed forward into the recruitment controller to alter the joint stiffnesses accordingly. Figure 6-13 shows the measured joint trajectory, which converges to the desired joint angle despite a large disturbance force. The true joint angle, shown in blue, is offset by $\tilde{\theta}(t)$, which settles to a constant value at about the same rate as the

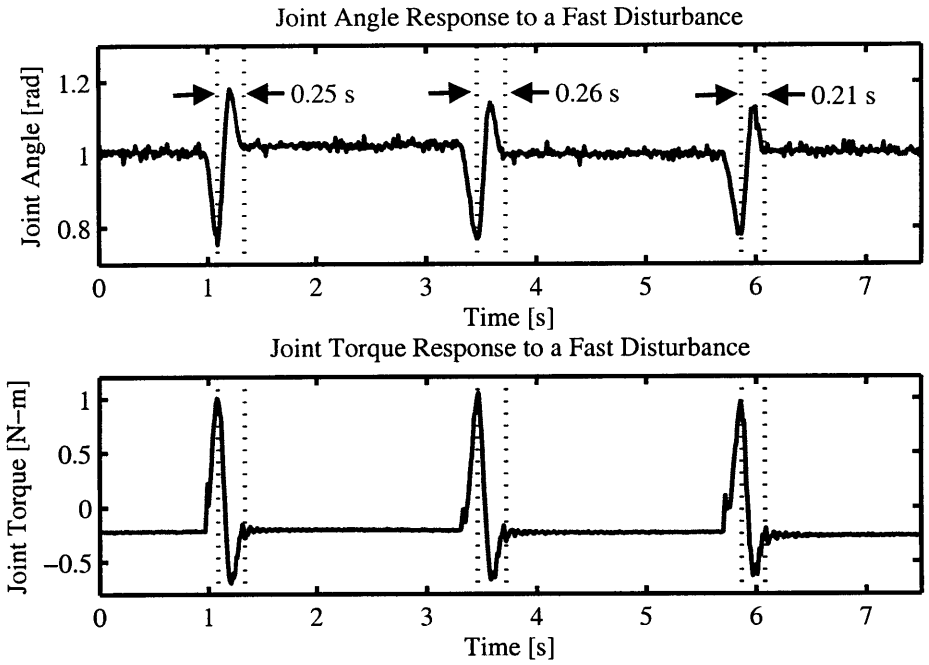

Figure 6-12: The passive elasticity of the actuators enables the arm to reject high-frequency perturbations that the closed-loop controller cannot possibly respond to.

actuator's physical reponse.

6.5 Conclusion

This chapter has shown how a simple, biologically-inspired recruitment model can be used to design and build SMA actuators having tunable stiffness. With the ability to significantly alter the stiffness of an actuator, equilibrium point control of an antagonistically actuated joint was implemented in hardware using a stochastic recruitment control system. An adaptive, feed-forward compensation scheme was developed for rejecting slow or constant disturbances without interfering with the high frequency disturbance rejection peculiar to equilibrium point control.

There is much that can be done to improve upon the results presented here. The ability to perform the same motion at varying levels of total stiffness was not discussed, although it is not hard to imagine. Also, maximizing total joint stiffness may not be the best way to elicit a rapid response from the arm, although it was simple to implement and robust to simple models and sensor noise. It could be useful

to develop feed-forward motion primitives to reduce point-to-point movement times.

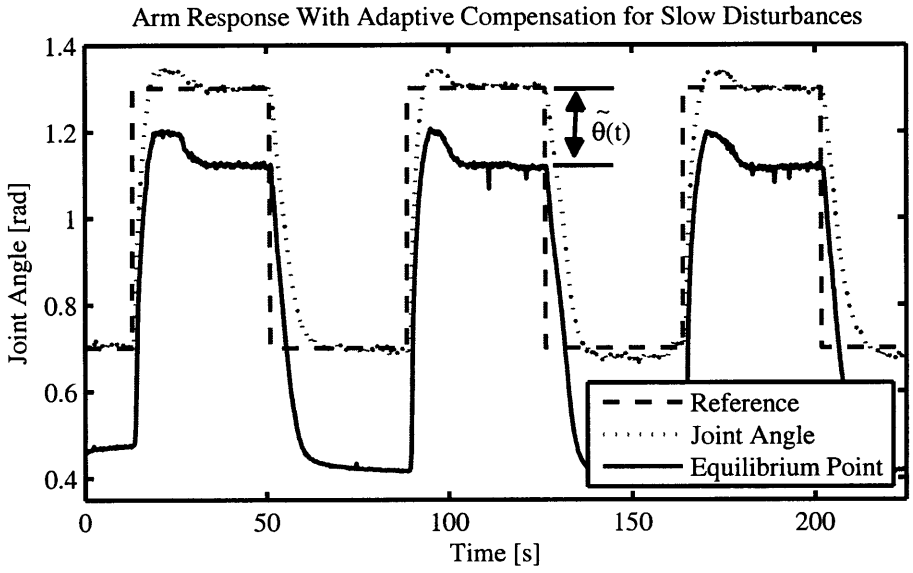

Figure 6-13: The arm can track a reference joint angle even with the arm in a vertical configuration, if an adaptive estimator can be used to offset the equilibrium point trajectory from the desired position.

Chapter 7

Conclusions

In this thesis, I presented a novel, unified framework for controlling an active material actuator made of many small elements as a noisy, single-input, single-output system through the intentional randomization of each functional unit's behavior. The significance of this is that actuators of this kind can scale up much more gracefully than those having a single monolithic element. I developed a physical model of actuators composed of many two-state functional units, showing how the ensemble system outputs could be adjusted as a function of the nuber of units "recruited" into the active state. I showed how this system can be treated as a random process if the behavior of each unit is randomized, so that it transitions between the active and inactive material states with known probabilities. Then I demonstrated that this process can be controlled in a manner similar to a classical SISO control loop, based only on the total force and displacement produced by the actuator. The ability to control the actuator improves with the number of units that are added to the actuator. I explained how higher-order discrete-state machines could be controlled by using a state space observer-controller framework. Finally, I presented a robot arm using two stochastic, recruitment-based actuators to control joint position and respond to fast and slow disturbances.

The larger significance of this work is a change in the way that distributed, multi-agent systems are seen. Many researchers assume that the solution lies in distributed computation. The resulting control architectures usually involve agents that have a

great deal of computational power. The control goals are achieved through a series of complex negotiations between each agent in the system. While the decentralized approach has obvious benefits in some contexts, many of these problems can be solved in the framework of a single central observer-controller, since often the desired system behavior is posed in terms of a global output or describing statistic. I hope that the tools for estimation and control presented here can be of use in simplifying many problems involving the control of ensembles of hybrid-state systems.

Much remains to be done in order to rigorously pose the stochastic recruitment control architecture as a general solution to the problem of controlling random, hybrid-state ensembles. This work has presented some possible approaches to the derivation of control laws assuming that the central controller has either full knowledge or limited knowledge of the ensemble state of the system. It is relatively inflexible in assuming that full control authority is available, that is, that it is possible to arbitrarily control the probability of transitions between states. However, problems of limited control authority are very much of interest. As mentioned in the conclusion of Chapter 4, some work exists in the formulation of open-loop inputs having constrained state transition probabilities. This must be extended to closed-loop control laws if this is to be presented as a general solution.

Another area that deserves some attention is the extension of the stochastic recruitment control architecture to systems exhibiting higher-order random behaviors, such as agents whose state transition probabilities are proportional to the number of agents already in one or more states. This is needed to model systems in which multi-agent interactions are important in determining system behavior. Also, neighbor-to-neighbor agent interactions may introduce some degree of correlation into the agents' behavior. Currently, the statistical models do not account for this, although it may be useful (for instance, with regard to the problem of cross-heating SMA elements). This kind of extension is absolutely necessary for properly describing spatially distributed systems such as cell cultures. In such systems, describing ensemble behavior can still be quite useful, but the local correlations may necessitate more careful modeling of how individual agent behaviors scale up.

In closing, I'd like to come back to the original motivation for this thesis, which is the development of new actuators. As roboticists, the importance of this topic is self-evident. Limits on actuation are among the key constraints placed on the problems of locomotion, manipulation, and robot-human interaction. If the path toward the actuators of the future were already known, a great number of additional applications would spring up overnight. However, the challenge of developing better actuators spans many disciplines, from materials science to biology to control systems theory. It is unlikely that a materials researcher will develop an active material remotely resembling an ideal force or displacement source. It is equally unlikely that a material will be discovered having ideal scaling properties, or an ideal set of activation dynamics. The non-ideal properties of these new materials must prompt new and creative solutions from all other contributors to the field, including roboticists who must work to revise their notion of what an ideal actuator is. In my work, I have endeavored to understand the limitations of the active materials presently available, and I have made an effort to compensate for these limitations by revising our notions of how active materials are controlled. It is at present my best attempt; I am certain it will not be my last attempt.

Appendix A

Simple Hardware Implementations of Random Finite State Machines

A.1 Introduction

This appendix outlines a design for an *on/off* state machine that can be implemented in a handful of logic gates, capable of transitioning from *on* to *off* and from *off* to *on* with controllable probabilities $p(t)$ and $q(t)$. In the experimental work presented in this thesis, these state machines were implemented in software in the experimental work using a prepackaged pseudo-random number generator. However, the argument has been put forth that this randomized architecture can be scaled down to very, very small units with minimal computational footprints. This will possbily enable the integration of this kind of state machine into micro-fabricated actuator devices, such as the printable actuators demonstrated by Kim [30]. In defense of this argument, this appendix offers one solution to the problem of producing small random finite state machines in hardware.

A good starting point for the implementation of a random two-state machine is a latch, shown in Fig. A-1. A latch is a circuit which sets the output pin, **Q**, to the value on the input **D** when a rising edge is detected on the clock pin, **E**. One way to command state transitions would be to set the desired state on **D**, while sending a clock signal into an AND gate, as shown. The other input to the AND gate is

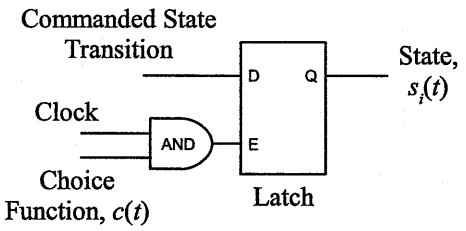

Figure A-1: A two-state machine with random state transitions could be implemented with a latch; the value of **Q** is the last value on **D** when **E** was high.

a random binary "choice" signal, $c(t)$, whose probability of being high at any point in time is controllable. Thus, each clocked command is accepted only if c(t) is high while the clock signal is sent. This method would not work to generate simultaneous transitions from *off* to *on* and from *on* to *off*, because only one command can be sent at any time. However, this example implementation is presented because it is clear, and because the feedback control laws that prove most useful make only unilateral transitions, as demonstrated in §4.5.

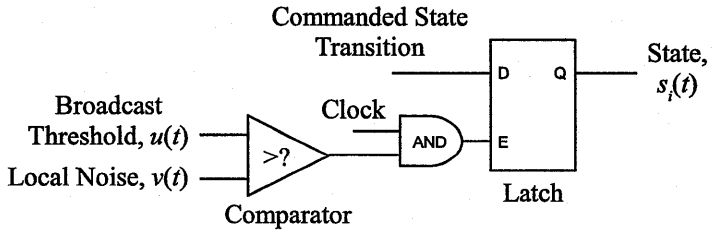

Figure A-2: Random state transition choice could be implemented with a comparator comparing a broadcast threshold to a locally generated random signal.

It is not difficult to imagine a circuit which generates a binary random choice signal $c(t)$ on a controllable probability distribution. Figure A-2 depicts one example circuit. Suppose that each unit is capable of generating a uniformly distributed random signal, say, between 0 and 1 volt, which is uncorrelated from all of the other units in the actuator. A broadcast value $u \in [0,\ 1]$ could be compared to this random signal with a comparator of some kind, generating a binary random variable $c(t)$ with

the probability distribution:

$$\Pr\{c(t) = 1\} = u \tag{A.1}$$
$$\Pr\{c(t) = 0\} = 1 - u$$

This could be sampled at regular time intervals via a broadcast clock signal to determine whether to make a state transition. In software, comparison with a uniform random variable is easy to implement, assuming that a good pseudo-random number generator is available, and that each unit's pseudo-random number generator can be independently seeded.

A.2 Hardware Random Number Generation

Each unit needs hardware that can generate a uniformly distributed random variable that is uncorrelated from all the other units. One way to think of this is as a stream of unbiased bits, that is, a stream of bits where the probability of each bit being 1 is 0.5 and the auto-correlation of the bit sequence is zero for any nonzero time delay. The equivalence of these two can be shown by looking at an interval between 0 and 2^k on a number line, where k is the number of bits sampled from the random sequence. The most significant bit of the number being one with probability 0.5 corresponds to the number lying in the upper half or the lower half of the interval with equal likelihood. The distribution of the next bit down represents the odds of lying in the upper quarter or lower quarter of each half. It is straightforward to see, then, how having an unbiased stream of random bits is equivalent to having a uniformly distributed random variable.

There are many ways of obtaining random electrical noise in a small circuit. With a good amplifier, Johnson-Nyquist noise can be measured in any conductor [25]. Shot noise, due to quantization of charge, can be measured wherever current is flowing [23]. Avalanche diode breakdown noise is easy to obtain if the diode is biased properly [21]. Any one of these noise sources could be amplified and thresholded to produce a binary

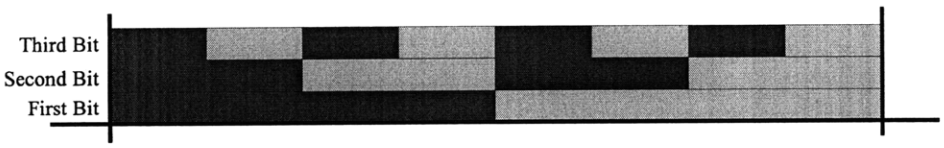

Figure A-3: A binary number made up of unbiased random bits will lie on a uniform distribution. The first bit will determine which half of the range the variable lies in; the second bit determines which half of that half, and so on.

random variable. However, in practice, these noise sources will not produce noise that is both unbiased and without some auto-correlation at nonzero time delays. Some noise sources, such as avalanche diode noise, tends to produce burst events rather than a steady stream of random output. Even amplified Johnson-Nyquist noise, which in theory is white noise, will probably pick up bias due to other parasitic signals. This is a noted problem in the field of computer security [14]. It is often solved by using a mixing function, a function whose purpose is to thoroughly scramble its input so that the output appears to be uncorrelated, unbiased noise. RFC 4086, the Network Working Group standard defining random number generation, suggests any number of mixing functions as a good way to scramble a signal in order to obtain these properties. Many of the mixing functions described in the standard are far more complicated than is required in this case. They are designed to prevent an attacker from predicting the output based on partial measurements. In this case, the only criteria of interest relate to bias and auto-correlation, so much simpler mixing functions can be used.

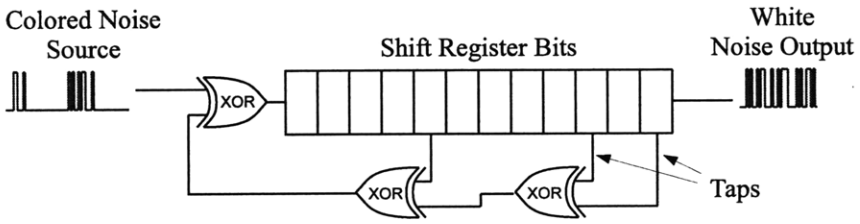

Figure A-4: A linear feedback shift register is a simple-to-implement hardware mixer made from a shift register and several XOR gates. It effectively removes bias and auto-correlation from a random source signal.

Once mixing method sufficient for producing well-behaved random numbers is

called the linear feedback shift register, or LFSR. Shown in Fig. A-4, it consists of a shift register with selected taps wired into XOR gates, fed back into the input. A binary random variable, fed into the LFSR, will emerge as an unbiased, uncorrelated sequence[1], if shift register taps used for feedback are carefully chosen. This method is used to "de-correlate" all kinds of signals. It is at the core of the cyclic redundancy checksum (CRC) algorithm, a mainstay of communications error correction protocols since the 1960s. Peterson's 1961 paper on the subject is a good reference on the properties of linear feedback shift registers, and explains how to choose the shift register taps to use for feedback [50]. Goresky and Klapper's preprint book *Algebraic Shift Register Sequences* is a much broader look at this class of mixing function [16].

A.3 Making Transition Decisions in Hardware

The flip-flop, comparator and LFSR random number generators could be combined with a digital-to-analog converter attached to all of the taps of the feedback shift register, as shown in Fig. A-5. This circuit will make state transitions with the controllable probability distribution from (A.1). However, better solutions exist that do not necessitate digital-to-analog conversion, a relatively complex process compared to the other components involved. One alternative would be to use some kind of pulse width modulation on the command signal. Figure A-6 shows a schematic for a pulse width modulated random state machine. Like before, a LFSR is used to scramble the bits from a local random source. Instead of feeding the bits into a DAC, n sequential bits could be fed into a multiple-input AND gate to construct a binary random variable with known bias, so that the probability of the AND gate output being high is 1 in 2^n. Suppose that the clock signal sent to all of the units is a pulse of variable width. By setting the pulse width to k times the clock period of the shift register, the probability of pin **E** seeing a rising edge within the width of the clock

[1]There will be some auto-correlation if the sequence runs uninterrupted through all possible states in the shift register. The number of states is large, on the order to 2^K, where K is the number of bits in the shift register.

pulse is equal to the joint probability that the biased signal was high at any time,

$$\Pr\{c(t) = 1\} = 1 - (1 - 2^{-n})^k \tag{A.2}$$

This probability can be modulated from 0 to almost 1. As we will explain in §4.3.1, this is good enough to control the behavior of the actuator.

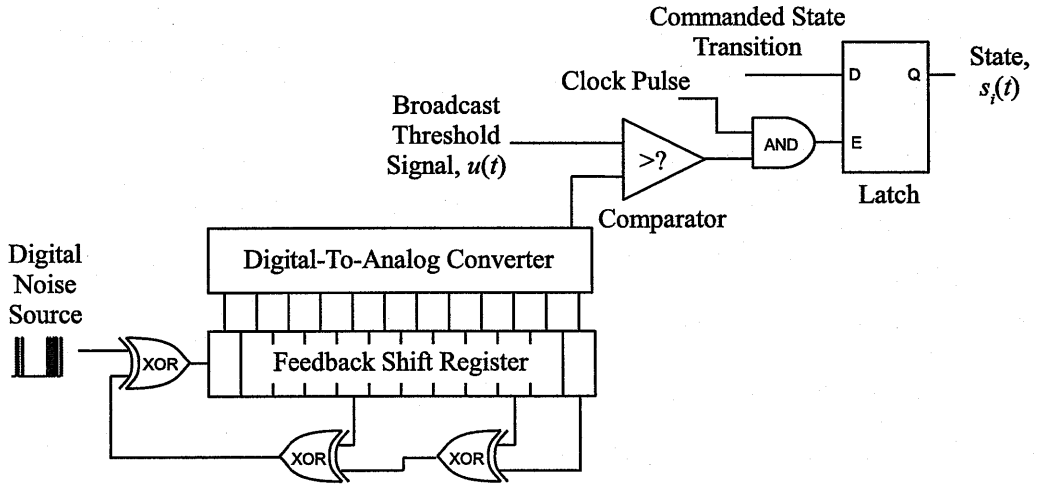

Figure A-5: A digital/analog circuit for producing random decisions with known probability. The broadcast analog threshold value determines the likelihood that the output will be high when sampled.

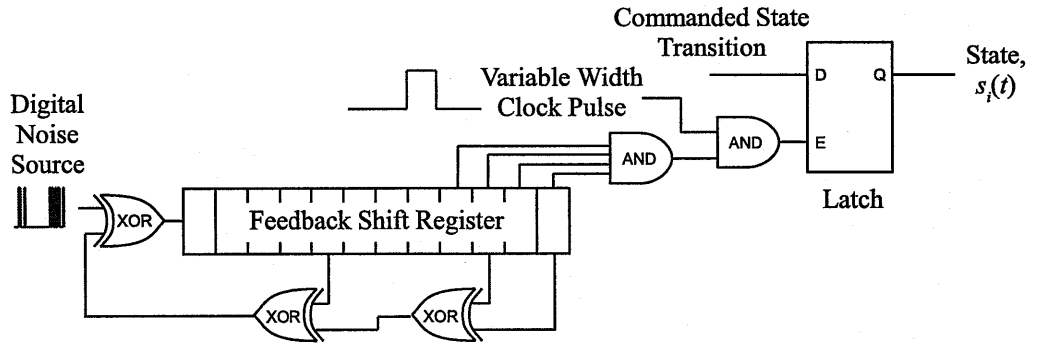

Figure A-6: An all-digital circuit for producing random decisions with known probability. By varying the width of the clock pulse, the probability that the output will be high can be controlled.

A.4 Conclusion

This brief gate-level circuit sketch shows that it is possible to make very small computational agents whose stochastic behavior is well-characterized and controllable. The purported value of using random state machines instead of distributed computation [12] or an addressed bus is that identical random state machines with a very small circuit would be ideal for scaling down to the integration of possibly very small and very fast actuator units. The end result, presented in Fig. A-6, shows that discrete, uncorrelated random behaviors are not difficult to generate with small digital circuits. Time-tested methods of mixing physical random noise can be used to produce unbiased streams of effectively random bits. These can be manipulated using logic gates to modulate the probability with which a latch-based state machine makes a transition.

Appendix B

Minimal Feedback Policies

B.1 Introduction

One of the control law candidates discussed in §4.5 was the minimal feedback control law, so called because it requires only the minimum amount of feedback information necessary to prove probability 1 convergence of the control system: knowledge that the present state is within the target set, R. This could be considered as a very rudimentary output to the system, $y(t)$, a Boolean measurement which lets the controller know if the current output has reached the destination:

$$y(t) = \begin{cases} \text{true}, & \text{if } \underline{N}(t) \in R \\ \text{false}, & \text{if } \underline{N}(t) \notin R \end{cases} \tag{B.1}$$

The proposed control strategy from §4.5.1 based only on this information is to constrain the inputs $p(t)$ and $q(t)$ so that the expected future error is zero, *not conditioned on any knowledge of the present state*:

$$p(t) = \begin{cases} \frac{N_{on}^{ref}}{N^{Total}}, & \underline{N}(t) \notin R \\ 0, & \underline{N}(t) \in R \end{cases} \tag{B.2}$$

$$q(t) = \begin{cases} \frac{N^{Total} - N_{on}^{ref}}{N^{Total}}, & \underline{N}(t) \notin R \\ 0, & \underline{N}(t) \in R \end{cases}$$

When the system is within the target state, the state transitions are commanded to cease, so that the target set is absorbing. This minimal law is interesting not because of its incredible performance, but because it serves as a good comparative case against which the performance of any other $\Pr\{1\}$ convergent control law should be measured. Control laws that assume more in-depth knowledge of the state should perform better than the minimal law. Otherwise, the additional information is not improving the performance of the control system. The purpose of this appendix is to demonstrate that the control law presented in §4.5.1 is the "best" minimal feedback control law for performance comparison.

B.2 Optimal Minimal Feedback Laws

As one might imagine, there are quite a number of control laws that converge using only the minimal feedback information provided by (B.1). Any open-loop input of the kind developed in §3.3.1 has some chance of entering the target set R as time progresses; more specifically, it is very likely that the state will pass through R quickly if an open-loop input is chosen so that the expected steady-state behavior as expressed in (3.25) is close to $\underline{N}^{ref}$. This appendix proves that of these many possible laws, the one presented in Chapter 4 is the best in terms of the expected convergence time. To show, this, we compute the expected time to converge, J, as before, but instead of defining this as a function of the initial state $\underline{N}(0)$, it can be defined as a function of the initial probability of each unit being on, $x_{on}(0)$,

$$J(x_{on}(0)) = E\left[\sum_{t=0}^{\infty} g(y(t)) \,\middle|\, x_{on}(0)\right] \tag{B.3}$$

where $g(y(t))$ is a cost-per-stage function given by

$$g(y(t)) = \begin{cases} 0, & y(t) = \text{ true} \\ 1, & y(t) = \text{ false} \end{cases} \tag{B.4}$$

Recall that the time evolution of a single agent's probability of being *on*, $x_{on}(t)$, can be written as an eigenvalue decomposition as in (3.23), which, for the two state case, looks like this:

$$x_{on}(t) = x_{ss} + (x_{on}(0) - x_{ss})\lambda_2^t \tag{B.5}$$

$$x_{ss} = \frac{p_c}{p_c + q_c} \tag{B.6}$$

$$\lambda_2 = 1 - p_c - q_c \tag{B.7}$$

The goal behind expressing the cost as a function of $x_{on}(0)$ is to find the control policy minimizing the expected convergence time for an arbitrary initial value of $x_{on}(0)$. In finding the control policy, we consider constant transition probabilities, $p(t) = p_c$ and $q(t) = q_c$, and determine probability values that do not need information about initial conditions. These state transitions are repeated until the destination has been reached, at which point all state transitions are commanded to cease by setting $p(t) = q(t) = 0$. The following recursive form of $J(x_{on}(t))$ is useful for analyzing the cost function. Let J_t be the cost-to-go from time t, that is, the partial sum of cost-per-stage values starting at time t:

$$J_t = E\left[\sum_{k=t}^{\infty} g(y(k)) \middle| \, x_{on}(t)\right] \tag{B.8}$$

From this definition it is clear that J_0 is equivalent to $J(x_{on}(0))$. By separating the cost-per-stage from the rest of the summation, a recursive formula of the series in (B.3) can be formed:

$$J_t = E\left[g(y(t))\right] + E\left[J_{t+1} \middle| \, x_{on}(t)\right] \tag{B.9}$$

The stochastic nature of the cost function arises from the uncertainty about when the exact number of desired *on* agents has been reached. The probability H_t of reaching the desired state can be described using (3.18). For the present proofs, R will be defined so that $R = \{[N_{on}^{ref} N^{Total} - N_{on}^{ref}]^T\}$, which makes computation of H_t

straightforward,

$$H_t = P(y(t) = \text{true} \mid x_{on}(t)) = \begin{pmatrix} N^{Total} \\ N^{ref} \end{pmatrix} x_{on}(t)^{N^{ref}} (1 - x_{on}(t))^{N^{Total} - N^{ref}} \quad \text{(B.10)}$$

Similarly, the probability of not reaching the desired state is defined as $\bar{H}_t$,

$$\bar{H}_t = P(\bar{y}(t) = \text{false} \mid x_{on}(t)) = 1 - H_t \quad \text{(B.11)}$$

Using $\bar{H}_t$, the cost-to-go J_t in (B.9) is given by

$$J_t = \bar{H}_t + \bar{H}_t J_{t+1} = \bar{H}_t[1 + J_{t+1}] \quad \text{(B.12)}$$

The second term in (B.9) is the probability of not reaching the goal at time t multiplied by the expected cost-to-go from $t+1$ assuming that the desired state has not yet been reached. It is important to note that the expected convergence time $J(x_{on}(0))$ takes a finite value for a broad range of transition probabilities p_c and q_c. The following Lemma states this.

Lemma The expected convergence time is bounded, $J(x_{on}(0)) < \infty$, for an arbitrary initial state $x_{on}(0)$ and an arbitrary desired number of *on* agents, $N_{on}^{ref} \in \{0, 1, ..., N^{Total}\}$, if the system is commanded to halt when $y(t)$ is true, and if the constant transition probabilities p_c and q_c satisfy the following conditions:

1. $\lambda_2 = 1 - p_c - q_c$ is bounded by

$$|\lambda_2| \leq 1 - \epsilon, \text{ where } \epsilon > 0 \quad \text{(B.13)}$$

2. If $N_{on}^{ref} = 0$, then $p_c = 0$.
If $N_{on}^{ref} = N^{Total}$, then $q_c = 0$.
If $1 \leq N_{on}^{ref} \leq N^{Total} - 1$, then $\exists \delta > 0$ such that $\delta \leq x_{ss} = \frac{p_c}{p_c + q_c} \leq 1 - \delta$.

The first condition, (B.13), requires that the time evolution of the probability distribution $x_{on}(t)$ converges. See (B.5). The second conditions are natural requirements. For example, if all the agents are to be recruited to the *off* state, $N_{on}^{ref} = 0$, there is no point in turning on any agent, i.e. $p_c = 0$. If at least one agent must be recruited to *on*, $N^{ref} \geq 1$, transitions from *off* to *on* should not be prohibited, i.e. $p_c \neq 0$. Considering the case when all the agents are initially in the *off* state, p_c must not be 0, as the last condition manifests. Therefore, the convergence of $J(x_{on}(0))$ is guaranteed for all p_c and q_c except for those pathological cases. It is obvious that J_t converges for all $t \geq 0$ as long as $J(x_{on}(0))$ converges. The proof of Lemma is shown in §B.3. The following propositions are useful for finding an optimal control policy with respect to the constant state transition probability parameters p_c and q_c:

Proposition 1 Suppose that N^{Total} agents are independent, identically-distributed binary-state Markov processes with transition probabilities $p(t)$ and $q(t)$. These parameters are held at constant values p_c and q_c until $y(t)$ is true, at which point they are set to 0. Let N_{on}^{ref} be a desired number of agents to recruit to the *on* state. Using the Boolean feedback signal $y(t)$ given by (B.1) alone, the cost function $J(x_{on}(t))$ in (B.3) takes an extremum with the following policy parameters for any initial distribution $x_{on}(0)$:

$$(p_c, q_c) = \left(\frac{N_{on}^{ref}}{N^{Total}}, \quad 1 - \frac{N_{on}^{ref}}{N^{Total}} \right) \tag{B.14}$$

This will be shown by demonstrating that the gradient of J with respect to p_c and q_c is zero for these parameter values.

Proof $\partial J / \partial p_c$ can be expanded as a recursive series from any point in time by taking the partial derivative of (B.12) with respect to p_c,

$$\frac{\partial J_t}{\partial p_c} = \frac{\partial \bar{H}_t}{\partial p_c}[1 + J_{t+1}] + \bar{H}_t \frac{\partial J_{t+1}}{\partial p_c} \tag{B.15}$$

Similarly, $\partial J/\partial q_c$ can be written as a recursive series,

$$\frac{\partial J_t}{\partial q_c} = \frac{\partial \bar{H}_t}{\partial q_c}[1 + J_{t+1}] + \bar{H}_t \frac{\partial J_{t+1}}{\partial q_c} \tag{B.16}$$

These series can be shown to converge under the same conditions as J_t, when $x_{on}(t)$ is bounded away from 0 and 1, using a similar argument (each term is bounded in magnitude and discounted exponentially). The sign of each term in this series is determined by the sign of the partial derivative of $\bar{H}_t$ at each point in time. The other terms in the expression are probabilities, which are positive, or truncated cost functions, which must also be positive. The derivative of H_t with respect to $x_{on}(t)$ reduces to an expression in terms of H_t. From (B.10),

$$\frac{\partial H_t}{\partial x_{on}(t)} = H_t \frac{N_{on}^{ref} - N^{Total} x_{on}(t)}{x_{on}(t)(1 - x_{on}(t))} \tag{B.17}$$

The partial derivative of $\bar{H}_t$ with respect to p_c is:

$$\frac{\partial \bar{H}_t}{\partial p_c} = H_t \frac{N^{Total} x_{on}(t) - N_{on}^{ref}}{x_{on}(t)(1 - x_{on}(t))} \frac{\partial x_{on}(t)}{\partial p_c} \tag{B.18}$$

The partial derivative of $\bar{H}_t$ with respect to q_c also contains a factor of $N^{Total} x_{on}(t) - N_{on}^{ref}$,

$$\frac{\partial \bar{H}_t}{\partial q_c} = H_t \frac{N^{Total} x_{on}(t) - N_{on}^{ref}}{x_{on}(t)(1 - x_{on}(t))} \frac{\partial x_{on}(t)}{\partial q_c} \tag{B.19}$$

The control policy of (B.14) implies that the convergence rate given by (B.7) is zero, $\lambda_2 = 0$. Therefore, the value of $x_{on}(t)$ can be found for all t using (B.5):

$$x_{on}(t) = \begin{cases} x_{on}(0), & t = 0 \\ N_{on}^{ref}/N^{Total}, & t > 0 \end{cases} \tag{B.20}$$

Therefore, the factor $N^{Total} x_{on}(t) - N_{on}^{ref}$ involved in (B.18) and (B.19) is zero for all $t > 0$, $N^{Total} \cdot N_{on}^{ref}/N - N_{on}^{ref} = 0$. Consequently $\partial \bar{H}_t/\partial p_c = \partial \bar{H}_t/\partial q_c = 0$ for all $t > 0$, by extension, it is zero for every term in the series defining $\partial J_t/\partial p_c$ and $\partial J_t/\partial q_c$. This concludes the proof.

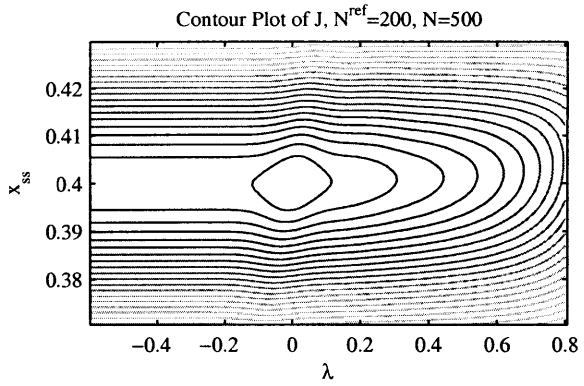

Figure B-1: The contour lines of the cost function for $N = 500$, $N^{ref} = 200$

Figure B-1 shows an example contour of the cost function $J(x_{on}(0) = 0)$ for $N^{ref}/N^{Total} = 0.4$. Parameters $x_{ss} = p_c/(p_c + q_c)$ and $\lambda_2 = 1 - p_c - q_c$ are used, which have one-to-one correspondence with parameters p_c and q_c. It should be noted that the cost increases sharply as x_{ss} deviates from the target distribution N_{on}^{ref}/N^{Total}. Along the axis of convergence rate λ_2, on the other hand, $J(0)$ varies gently and is therefore unclear where it takes its minimum. The following Proposition proves that the proposed policy is at an extremum that is minimized with respect to λ_2 at that point, though this proof does not rule out other extrema resulting from variations in both x_{ss} and λ_2.

Proposition 2 Suppose that an ensemble of N^{Total} independent agents are controlled with state transition probabilities $p(t) = p_c$ and $q(t) = q_c$ and the Boolean feedback signal $y(t)$, commanding all transitions to cease when $y(t)$ is true. Among the control policies having the steady-state probability distribution set to the desired value,

$$x_{ss} = \frac{p_c}{p_c + q_c} = \frac{N_{on}^{ref}}{N^{Total}} \tag{B.21}$$

the optimal policy that minimizes the cost function $J(x_{on}(0))$ for an arbitrary initial probability distribution $x_{on}(0)$ is given by the convergence rate of $\lambda_2 = 1 - p_c - q_c = 0$.

Proof The class of control policies under consideration can be described as $p_c = \beta N_{on}^{ref}/N^{Total}$ and $q_c = \beta(1 - N_{on}^{ref}/N^{Total})$ with $\beta = 1 - \lambda_2$. These policies are on the

line of $x_{ss} = \frac{N_{on}^{ref}}{N^{Total}}$ in the (x_{ss}, λ_2) plane. The Proposition is proved if the gradient of J with respect to λ_2 is always pointing away from the optimal point candidate $\lambda_2 = 0$. This can be shown by calculating the product Z between the gradient at an arbitrary point along the line of $x_{ss} = \frac{N_{on}^{ref}}{N^{Total}}$ and the distance of that point from the optimal point, $\Delta\lambda_2$, and showing that Z is positive for all $\lambda_2 \neq 0$:

$$Z = \frac{\partial J}{\partial \lambda_2} \Delta\lambda_2 > 0 \tag{B.22}$$

As before a recursive formula can be used for the evaluation of the partial series starting at time t, Z_t. For an arbitrary t,

$$Z_t = \frac{H_t[1 + J_{t+1}]}{x_{on}(t)(1 - x_{on}(t))} G_t + \bar{H}_t Z_{t+1} \tag{B.23}$$

where the sub-expression G_t determines the sign of each term in the series,

$$G_t = \left(N^{Total} x_{on}(t) - N_{on}^{ref}\right) \frac{\partial x_{on}(t)}{\partial \lambda_2} \Delta\lambda_2 \tag{B.24}$$

As above, the only terms in the series defining the product Z that can be negative are within G_t. G_t can be evaluated for $p_c = \beta N_{on}^{ref}/N^{Total}$ and $q_c = \beta(1 - N_{on}^{ref}/N^{Total})$, to obtain the expression:

$$G_t = tN^{Total} \left(x_{on}(0) - \frac{N_{on}^{ref}}{N^{Total}}\right)^2 (1 - \beta)^{2t} \geq 0 \tag{B.25}$$

If the initial probability distribution coincides with the desired one, $x_{on}(0) = \frac{N_{on}^{ref}}{N^{Total}}$, then the rate at which $x_{on}(t)$ approaches x_{ss} does not impact the expected convergence time. For all other values of $x_{on}(0)$, the above expression G_t is positive for all $\beta \neq 1$, that is, all $\lambda_2 \neq 0$, so the cost function is increasing at any point other than the optimal point. This concludes the proof.

B.3 Proof of Cost Function Boundedness

First we can show that H_t in (B.10) is uniformly lower-bounded.

i). When $N^{ref} = 0$, $p_c = 0$ and $x_{ss} = 0$, and H_t reduces to

$$H_t = (1 - x_{on}(t))^{NTotal} \tag{B.26}$$

From (B.5),

$$x_{on}(t) = x_{on}(0)\lambda_2^t \leq \lambda_2^t \leq (1-\epsilon)^t \leq 1 - \epsilon \text{ for } t \geq 1 \tag{B.27}$$

Using this in (B.26) yields

$$H_t \geq (1 - (1 - \epsilon))^{NTotal} = \epsilon^{NTotal} > 0 \tag{B.28}$$

ii). Similarly, when $N_{on}^{ref} = N^{Total}$, then $q_c = 0$ and $x_{ss} = 1$, and

$$\begin{aligned} x_{on}(t) &= 1 + (x_{on}(0) - 1)\lambda_2^t \geq 1 - \lambda_2^t \geq 1 - (1-\epsilon)^t \geq 1 - (1 - \epsilon) \\ &= \epsilon \end{aligned} \tag{B.29}$$

Therefore,

$$H_t = x_{on}(t)^{NTotal} \geq \epsilon^{NTotal} > 0 \tag{B.30}$$

iii). When $1 \leq N_{on}^{ref} \leq N^{Total} - 1$,

$$\delta \leq x_{ss} = \frac{p_c}{p_c + q_c} \leq 1 - \delta \tag{B.31}$$

The transient term in (B.5) is bounded by

$$|(x_{on}(0) - x_{ss})\lambda_2^t| \leq |\lambda_2^t| = (1-\epsilon)^t \tag{B.32}$$

From these two it follows that $x_{on}(t)$ is upper and lower bounded as

$$\delta - (1-\epsilon)^t \leq x_{on}(t) \leq 1 - \delta + (1-\epsilon)^t \tag{B.33}$$

Let $t*$ be the smallest integer satisfying

$$0 < \frac{\delta}{2} \leq \delta - (1-\epsilon)^{t*} \tag{B.34}$$

Solving the above for $t*$,

$$t* = \left\lceil \frac{\log \frac{\delta}{2}}{\log(1-\epsilon)} \right\rceil > 0 \tag{B.35}$$

where $\lceil z \rceil$ means the smallest integer that is larger or equal to z. Note that $0 < \delta/2 < 1$ and $0 < 1-\epsilon < 1$ assure the existence of $t*$. From (B.33) and (B.34),

$$0 < \frac{\delta}{2} \leq x_{on}(t) \leq 1 - \frac{\delta}{2} < 1, \forall t \geq t* \tag{B.36}$$

Substituting this into (B.10) yields

$$H_t \geq x_t^{N^{ref}} \geq \left(\frac{\delta}{2}\right)^{N^{ref}} \left(\frac{\delta}{2}\right)^{N-N^{ref}} = \left(\frac{\delta}{2}\right)^{N} > 0 \tag{B.37}$$

for all $t \geq t*$. Let $H_{min} = \min\left(\epsilon^{NTotal}, (\delta/2)^{NTotal}\right) > 0$. From (B.28),(B.30), and (B.37) it follows that H_t is uniformly lower-bounded for all $t \geq t*$:

$$H_t \geq H_{min} > 0 \tag{B.38}$$

or $\bar{H}_t \leq H_{max} \doteq 1 - H_{min} < 1$. From (B.12)

$$J_{t*} = \bar{H}_t + \bar{H}_t\bar{H}_{t+1} + \bar{H}_t\bar{H}_{t+1}\bar{H}_{t+2} + ... \tag{B.39}$$

$$\leq H_{max} + H_{max}^2 + H_{max}^3 + ... = \frac{H_{max}}{1 - H_{max}} < \infty \tag{B.40}$$

Now that J_{t*} is finite, we find from (B.8) and (B.12),

$$\begin{aligned} J(x_{on}(0)) &= \bar{H}_0 + \bar{H}_0\bar{H}_1 + \bar{H}_0\bar{H}_1\bar{H}_2 + ... + \bar{H}_0...\bar{H}_{t*-1}J_{t*} \\ &< \infty \end{aligned} \tag{B.41}$$

since $0 \leq \bar{H}_t \leq 1$ for all t. Therefore, $J(x_{on}(0))$ is finite for arbitrary $x_{on}(0)$. This concludes the proof.

Bibliography

[1] M.N. Adal and S.B. Chew Cheng. The number, distribution and density of muscle spindles in two wing muscles of the domestic duck. *Journal of Anatomy*, 131(3):541–548, 1980.

[2] T. Balch, F. Dellaert, A. Feldman, A. Guillory, C.L. Isbell, Z. Khan, S.C. Pratt, A.N. Stein, and H. Wilde. How multirobot systems research will accelerate our understanding of social animal behavior. *Proceedings of the IEEE*, 94(7):1445–1463, July 2006.

[3] Y. Bar-Cohen. *Electroactive polymer (EAP) actuators as artificial muscles: reality, potential, and challenges*, chapter 1, pages 1–44. SPIE press, 2004.

[4] Y. Bar-Cohen, V. Olazábal, J. Sansi nena, and J. Hinkley. *Electroactive polymer (EAP) actuators as artificial muscles: reality, potential, and challenges*, chapter 14, pages 369–401. SPIE press, 2004.

[5] S. Berman, A. Halasz, V. Kumar, and S. Pratt. Bio-inspired group behaviors for the deployment of a swarm of robots to multiple destinations. *Robotics and Automation, 2007 IEEE International Conference on*, pages 2318–2323, 10-14 April 2007.

[6] D.P. Bertsekas. *Dynamic programming and optimal control.* Athena Scientific, Belmont, MA, 1995.

[7] P. Calvert. *Electroactive polymer (EAP) actuators as artificial muscles: reality, potential, and challenges*, chapter 5, pages 123–138. SPIE press, 2004.

[8] I. Chattopadhyay and A. Ray. Supervised self-organization of homogeneous swarms using ergodic projections of markov chains. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society*, 2009.

[9] K.J. Cho. *Reduction of dimensionality of a cellular actuator array for driving a robotic hand.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2007.

[10] Toki Corporation. Biometal helix website, August 2009. `http://www.toki.co.jp/biometal/english/contents.html`.

[11] N. Correll and A. Martinoli. Collective inspection of regular structures using a swarm of miniature robots. *Proceedings of the 9th International Symposium of Experimental Robotics (ISER)*, pages 375–385, 2004.

[12] D. De Rossi, M. Suzuki, Y. Osada, and P. Morasso. Pseudomuscular gel actuators for advanced robotics. *Journal of Intelligent Material Systems and Structures*, 3(1):75–95, 1992.

[13] J.L. Doob. *Stochastic Processes.* Wiley, New York, 1953.

[14] D.E. Eastlake, S.D. Crocker, and J.I. Schiller. Rfc–4086 randomness requirements for security. *Network Working Group*, 2005.

[15] R.G. Gallager. *Discrete stochastic processes.* Springer, 1996.

[16] M. Goresky and A. Klapper. *Algebraic Shift Register Sequences.* preprint, from `http://www.cs.uky.edu/ klapper/algebraic.html`, 2008.

[17] M.S. Grewal and A.P. Andrews. *Kalman filtering: theory and practice using MATLAB.* Wiley New York, 2001.

[18] A. Halasz, M.A. Hsieh, S. Berman, and V. Kumar. Dynamic redistribution of a swarm of robots among multiple sites. *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007*, pages 2320–2325, 2007.

[19] J. Halloy, G. Sempo, G. Caprari, C. Rivault, M. Asadpour, F. Tache, I. Said, V. Durier, S. Canonge, J. M. Ame, C. Detrain, N. Correll, A. Martinoli, F. Mondada, R. Siegwart, and J. L. Deneubourg. Social integration of robots into groups of cockroaches to control self-organized choices. *Science*, 318(5853):1155–1158, 2007.

[20] E. Henneman, G. Somjen, and D.O. Carpenter. Functional significance of cell size in spinal motoneurons. *Neurophysiology*, 28(3):560–580, May 1965.

[21] M.E. Hines. Noise theory for the read type avalanche diode. *Electron Devices, IEEE Transactions on*, 13(1):158–163, Jan 1966.

[22] N. Hogan. Adaptive control of mechanical impedance by coactivation of antagonist muscles. *Automatic Control, IEEE Transactions on*, 29(8):681–690, Aug 1984.

[23] P. Horowitz and W. Hill. *The art of electronics.* Cambridge Univ Pr, 1989.

[24] I.W. Hunter and S. Lafontaine. A comparison of muscle with artificial actuators. *Solid-State Sensor and Actuator Workshop, 1992. 5th Technical Digest., IEEE*, pages 178–185, 22-25 Jun 1992.

[25] J.B. Johnson. Thermal agitation of electricity in conductors. *Phys. Rev.*, 32(1):97, Jul 1928.

[26] A.A. Julius, A. Halasz, V. Kumar, and G.J. Pappas. Controlling biological systems: the lactose regulation system of escherichia coli. *American Control Conference, 2007. ACC '07*, pages 1305–1310, 9-13 July 2007.

[27] A.A. Julius, A. Halasz, M.S. Sakar, H. Rubin, V. Kumar, and G.J. Pappas. Stochastic modeling and control of biological systems: The lactose regulation system of escherichia coli. *Automatic Control, IEEE Transactions on*, 53(Special Issue):51–65, Jan. 2008.

[28] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[29] R.E. Kalman and R. Bucy. New results in linear prediction and filtering theory. *Trans. AMSE J. Basic Eng*, pages 95–108, 1961.

[30] S.G. Kim, Z. Traina, and H.W. Lee. Folding assembly of micro-actuators. *CIRP Annals-Manufacturing Technology*, 57(1):29–32, 2008.

[31] H.J. Kushner. On the stability of stochastic dynamical systems. *Proceedings of the National Academy of Sciences*, 53(1):8–12, 1965.

[32] K. Lerman and A. Galstyan. Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, 13(2):127–141, 2002.

[33] K. Lerman, A. Galstyan, A. Martinoli, and A. Ijspeert. A macroscopic analytical model of collaboration in distributed robotic systems. *Artificial Life*, 7(4):375–393, 2001.

[34] R.L. Lieber. Skeletal muscle is a biological example of a linear electro-active actuator. *SPIE 6th Annual International Symposium on Smart Structures and Materials*, pages 3669–03, 1999.

[35] W.S. Lohmiller. *Contraction analysis of nonlinear systems.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1999.

[36] F. Lorussi, S. Galatolo, C. Caudai, A. Tognetti, and D. De Rossi. Compliance control and feldman's muscle model. *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, pages 1194–1199, 2006.

[37] N. Ma, G. Song, and H.J. Lee. Position control of shape memory alloy actuators with internal electrical resistance feedback using neural networks. *Smart Materials and Structures*, 13(4):777–783, 2004.

[38] J.D.W. Madden, N.A. Vandesteeg, P.A. Anquetil, P.G.A. Madden, A. Takshi, R.Z. Pytel, S.R. Lafontaine, P.A. Wieringa, and I.W. Hunter. Artificial muscle technology: physical principles and naval prospects. *IEEE Journal of oceanic engineering*, 29(3):706–728, 2004.

[39] P.G.A. Madden. *Development and modeling of conducting polymer actuators and the fabrication of a conducting polymer based feedback loop.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2003.

[40] A. Martinoli, K. Easton, and W. Agassounon. Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *International Journal of Robotics Research*, 23(4):415–436, 2004.

[41] J. McIntyre and E. Bizzi. Servo hypotheses for the biological control of movement. *Journal of Motor Behavior*, 25(3):193–202, Sep 1993.

[42] T.A. McMahon. *Muscles, reflexes, and locomotion.* Princeton Univ Pr, 1984.

[43] N. Michael, M.M. Zavlanos, V. Kumar, and G.J. Pappas. Distributed multi-robot task assignment and formation control. *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 128–133, May 2008.

[44] A.H. Mitwalli. *Polymer gel actuators and sensors.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1998.

[45] N.S. Nise. *Control Systems Engineering.* John Wiley & Sons, Inc., New York, NY, USA, 2000.

[46] L.U. Odhner, J. Ueda, and H.H. Asada. A Cellular Control Architecture for Compliant Artificial Muscles. *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, 2006.

[47] L.U. Odhner, J. Ueda, and H.H. Asada. Feedback Control of Stochastic Cellular Actuators. *The 10th International Symposium on Experimental Robotics*, pages 481–490, 2006.

[48] L.U. Odhner, J. Ueda, and H.H. Asada. Stochastic Optimal Control Laws for Cellular Artificial Muscles. *2007 IEEE International Conference on Robotics and Automation*, pages 1554–1559, 2007.

[49] J. Park, J. Song, and H. Kim. *Lecture Notes in Control and Information Sciences*, volume 370, chapter Safe Joint Mechanism Based on Passive Compliance for Collision Safety, pages 49–61. Springer, 2008.

[50] W.W. Peterson and D.T. Brown. Cyclic codes for error detection. *Proceedings of the IRE*, 49(1):228–235, 1961.

[51] P.V. Pillai and I.W. Hunter. Stochastic system identification of the compliance of conducting polymers. *MRS Symposium Proceedings*, 1134, 2009.

[52] J. Polit and E. Bizzi. Characteristics of motor programs underlying arm movements in monkeys. *Journal of Neurophysiology*, 42(1):183–194, 1979.

[53] R.Z. Pytel, E.L. Thomas, and I.W. Hunter. In situ observation of dynamic elastic modulus in polypyrrole actuators. *Polymer*, 49(8):2008 – 2013, 2008.

[54] T. Saburi. Ti-ni shape memory alloys. *Shape memory materials*, pages 49–96, 1999.

[55] T.W. Secord and H.H. Asada. A humanoid foot with polypyrrole conducting polymer artificial muscles for energy dissipation and storage. *2007 IEEE International Conference on Robotics and Automation*, pages 2904–2909, 2007.

[56] B. Selden, K.J. Cho, and H.H. Asada. Segmented binary control of shape memory alloy actuator systems using the peltier effect. *2004 IEEE International Conference on Robotics and Automation.*, 5, 2004.

[57] J.J.E. Slotine, W. Li, et al. *Applied nonlinear control.* Prentice-Hall Englewood Cliffs, NJ, 1991.

[58] M.J. Sobel. The variance of discounted markov decision processes. *Journal of Applied Probability*, pages 794–802, 1982.

[59] O. Soysal and E. Sahin. Probabilistic aggregation strategies in swarm robotic systems. *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, pages 325–332, 2005.

[60] K. Sugawara and M. Sano. Cooperative acceleration of task performance: Foraging behavior of interacting multi-robots system. *Physica D Nonlinear Phenomena*, 100:343–354, February 1997.

[61] P.J. Sullivan. A Kalman filter approach to catch-at-length analysis. *Biometrics*, pages 237–257, 1992.

[62] Y.H. Teh and R. Featherstone. Accurate force control and motion disturbance rejection for shape memory alloy actuators. *2007 IEEE International Conference on Robotics and Automation*, pages 4454–4459, 2007.

[63] J. Ueda, L.U. Odhner, and H.H. Asada. Broadcast feedback of stochastic cellular actuators inspired by biological muscle control. *The International Journal of Robotics Research*, 26(11-12):1251–1265, 2007.

[64] J. Ueda, L.U. Odhner, S.-G. Kim, and H.H. Asada. Distributed stochastic control of mems-pzt cellular actuators with broadcast feedback. *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, pages 272–277, 0-0 0.

[65] N. Vitiello, E. Cattin, S. Roccella, F. Giovacchini, F. Vecchi, M.C. Carrozza, and P. Dario. The neurarm: towards a platform for joint neuroscience experiments on human motion control theories. *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1852–1857, 29 2007-Nov. 2 2007.

[66] G.G. Wallace, G.M. Spinks, L.A.P. Kane-Maguire, and P.R. Teasdale. *Conductive electroactive polymers: intelligent materials systems.* CRC, 2003.

[67] R.D. Woittiez, P.A. Huijing, and R.H. Rozendal. Influence of muscle architecture on the length-force diagram of mammalian muscle. *Pflügers Archiv European Journal of Physiology*, 399(4):275–279, 1983.

[68] G.I. Zahalak and S.P. Ma. Muscle activation and contraction: constitutive relations based directly on cross-bridge kinetics. *Journal of biomechanical engineering*, 112:52, 1990.