A Gestural Language For A Humanoid Robot

by

Aaron Ladd Edsinger

B.S., Stanford University (1994)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Science in Computer Science and Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

Author .....
Department of Electrical Engineering and Computer Science
May 11, 2001

Certified by .....
Rodney Brooks
Fujitsu Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

A Gestural Language For A Humanoid Robot

by

Aaron Ladd Edsinger

Submitted to the Department of Electrical Engineering and Computer Science
on May 11, 2001, in partial fulfillment of the
requirements for the degree of
Masters of Science in Computer Science and Electrical Engineering

Abstract

This thesis describes work done at the MIT Artificial Intelligence Laboratory on the humanoid robot platform, Cog. Humanoid research has long been concerned with the quality of the robot's movement. However, obtaining the elusive tempo and grace of the human motor system has proven to be a very difficult problem. The complexity of controlling high degree of freedom (DOF) humanoid robots, combined with insights provide by neurophysiological findings, has lead researchers to look at motor primitives (Williamson 1996) as an organizing methodology. We propose a data-driven approach to motor primitives in building a motor language for Cog. The proposed model is implemented on Cog and applied to the task of human motor mimicry.

Thesis Supervisor: Rodney Brooks
Title: Fujitsu Professor of Computer Science and Engineering

Acknowledgments

I would like to thank my advisor, Rod Brooks, for creating and supporting the strange and wonderful world that is the Humanoid Robotics Laboratory. And of course the living, breathing, creatures that in inhabit it and provide a constant source of stimulation: Jess, Paulina, Eduardo, Paul, Artur, Charlie, Lijin, Juan, Scaz, BP, Maddog, Cindy, Una-May, and Gorgio.

It goes with out saying that I could never have made the long journey from the hayfields of Enumclaw to the Artificial Intelligence Lab at MIT without the continual support and understanding of my family: Mom (aka Yo!), Dad (aka Hayseed), Eric (aka Walking-Disaster), Craig (aka Hagar), and Carrie (aka Lynner).

Of course Jess deserves a special mention for all of her help and for pointing our sights at the blue sky. And Kelly for long wine filled nights, hairy dogs, and everything else good. And Jeff for god knows what.

And finally, the supporting cast, without whom I'd still be living in a van on the streets of SF: Willie Nelson, Merle Haggard, Dostoyevsky, Flanders, Johnny Cash, Carlos Rossi jug wine, Dirty Red, Kris Kristofferson, Minnie, peanut butter, the office of Ron Wiken, Godspeed, Camus, Henry Moore, Tinguely, Rodin, Michaelangelo, Goya, the food truck, Palace, McMaster Carr, Berg Farm Supplies, the 65 Plymouth Valiant, the 76 Grumman step van (aka Shag), good bread, old cheese, warm beer, and cold women...

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

This thesis describes work done at the MIT Artificial Intelligence Laboratory on the humanoid robot platform, Cog. Humanoid research has long been concerned with the quality of the robot's movement. However, obtaining the elusive tempo and grace of the human motor system has proven to be a very difficult problem. The complexity of controlling high degree of freedom (DOF) humanoid robots, combined with insights provide by neurophysiological findings, has lead researchers to look at motor primitives (Williamson 1996) as an organizing methodology. We propose a data-driven approach to motor primitives in building a motor language for Cog. The proposed model is implemented on Cog and applied to the task of human motor mimicry.

1.2 Approaches to Humanoid Motor Control

In this chapter we describe the issues encountered when developing motor control systems for robots and for humanoid robots in particular. The set of problems that are posed by humanoid robotics are in some sense unique to the field in general. These problems both constrain and complicate research in this area. The scope of investigation for this thesis is motivated this set of problems.

Humanoid robotics is an endeavor inherently concerned with realism. As researchers in the field, we are attempting to model, simulate, and approximate what may myopically be considered one of nature's greatest accomplishments. In building humanoid robots we are tacitly asserting the following: human level intelligence and interaction with the world requires both a physical presence in the world and a human morphology. As (Brooks, Brezeal et al. 1998) have noted, we have human intelligence by virtue of having a human body.

However, it is not clear to what degree we need to model and mimic natural systems. The level at which we choose to approximate nature certainly varies depending on the area of investigation and on the purpose of that investigation. In the domain of humanoid motor control, this raises a number of issues.

Clearly, we would like our robots to move in a human-like manner. Nature's animals, regardless of our perception of native intelligence, never fail to astound us with the grace and dexterity of their movements. This overarching motivation has pushed researchers to a detailed study of biological motor control, ranging from physiological examination, to biomechanical models, to cognitive theories.

Implementing a motor control system with a similar quality of movement on a humanoid robot can be considered one of the grand challenges of the field. It is still an open question, however, what method or approach is suitable for this goal.

Researchers working towards this end have recognized that a simple recreation of human movement, as is done in the field of animatronics, is not enough in itself. The aesthetic of robot movement is closely tied to the morphology of the robot, the electro-mechanical dynamics of the robot, and most importantly, a tight coupling between the robot's environment, perceptual system, and motor system. Naturalism for humanoid robot movement must not just come from the well-coordinated execution of movement trajectories, but also from movement that is in appropriate response to the environment.

Unfortunately, the best of our attempts have been hampered by the physical technologies available to us. Actuators such as DC servo motors are bulky, heavy, and consume inordinate amounts of power. The sensorimotor feedback available on

our current systems pales in comparison to the massively parallel feedback employed in nature.

Regardless of our current technical limitations, there is still an interesting set of questions to explore. In fact, much of the grace and dexterity we admire in animals may not be a direct result of sophisticated musculature and sensory systems. We propose that a humanoid robotic system with limited sensorimotor faculties can still exhibit interesting and naturalistic motor behavior given the following conditions:

- The system is tightly coupled to a complex environment.
- There exists an organizing principle for the motor systems whereby complex movements can be generalized from simpler movements

The work in this thesis follows from this working assumption.

1.2.1 The Biological Motivation

The term "biologically motivated" has become widely used and perhaps overextended within the field of humanoid robotics. Clearly there are advantages to looking to nature for questions and for answers, particularly in a naturalistic domain such as this. However, the human body has not been carefully engineered the way, say, a car has, and it doesn't lend easily to a clean decomposition. Studies in pyschophysics, cognitive science, physiology, ethology, etc. can be illuminating as well as confusing. It can be difficult to move from a set of results gained from studies of a system to a well formed model of that system. Despite the insights that can be gained, it is not clear what level of biomimicry is useful. A connectionist approach, starting at the neural level, may leave one with a grossly simplified or intractable system. On the other hand, traditional approaches that treat the natural system as a set of compartmentalized boxes ignore the greater complexity of the whole, influenced in part by hormones, the environment, and biomechanics of the complete system.

Because the physical substrate, actuators, and sensors on a robot are fundamentally different than those in a natural system, a middle ground is perhaps necessary.

For example, for the purposes of this thesis, the rich and complex musculature system of the human body will be approximated by a spring and damper system (Williamson 1995). The true contribution of a biologically inspired approach to this work, however, is that of presenting a plausible organizing principle for humanoid motor control.

1.2.2 An Organizing Principle

While the literature review in Chapter 2 provides the biological justification for this approach, we find it instructive to begin with an overview of why, and for what, an organizing principle is useful in humanoid motor control.

The difficulty of the posed problem comes in part from the complexity of the human motor system. The human arm, for example, contains seven degrees of freedom and 26 muscle groups (Berniker 2000). It is a highly redundant system. A task such as pointing to an object in the world can yield an infinite number of solutions, yet the human system utilizes just a few. This in itself suggests that it is necessary to look for methods to simplify and structure the problem in such a way that it becomes tractable.

Luckily, the domain of human movement represents a constraint on the problem, limiting the large space of possible motor movements to those that are naturally occurring.

Additionally, humans have the ability to generate a large range of novel movements that adapt to the environment and task at hand. Models used in humanoid research cannot predict or learn all possible outcomes. Instead, the models need to be built upon an organizing principle which formalizes a method for generalizing movements from one domain to another.

Implicit in the choice of an organizing principle is a choice of representation for motor acts. If we construct an representational structure for motor actions, then a central component of this representation involves assessing the similarity or dissimilarity of its members. Such an ability requires that the relevant features of the domain are selected, or perhaps learned. It also requires that an encoding for the representative members of the domain be selected so that we can store, retrieve, and

compare these elements.

Finding a good representation is a challenge and can be task dependent. For example, human reaching in free space may lend itself to a Cartesian, exocentric representation. Contradicting this representation, bipedal walking is largely parameterized in terms of forces exerted at the feet. Thus the two tasks demand different forms of representation.

As both the kinematic and task complexity of a robot increase, a good representation of movement becomes necessary. A humanoid robot must be able to generalize from a small set of movements to a large repertoire of possible motor acts. Not only that, but it must be able to correlate the global nature of a motor act with perceptual stimuli. But assessing the global nature of a motor act requires an abstraction away from simple joint trajectories. It requires embedding the motor act in a structure such that the perceptual stimuli can apply to related motor acts as well.

The approach taken in this work can be considered a variant of what are commonly referred to as motor primitives (Bizzi, Accornero, Chapple & Hogan 1984). The crux of this approach, supported by biological findings, is that we compose complex motor acts out of a small set of simpler motor acts. This small set of motor primitives provides the building blocks of our complete set of movements. It also provides a means for representing movements such that they can be compared and categorized in learning situations.

1.2.3 A Gestural Language

We use the term "gestural language" as a metaphorical means of conveying the intent of our approach. It is used analogously to the way the written language utilizes alphabets, verbs, nouns, and adverbs to formalize ideas. In a similar fashion, we can view motor primitives as the canonical elements of an alphabet from which more complex elements can be constructed. From this vantage, the problem can be decomposed into the following subproblems:

- Determine a useful encoding for a movement primitive.

- Specify, or learn, the basis members of a gestural alphabet corresponding to the canonical movements used in human motor acts.
- Specify, or learn, a grammar for the language such that basis elements can be combined in a meaningful manner to form more complex movements.
- Apply the language to a real world problem on the robot by using perceptual stimuli to activate a viable gestural sentence in response to the given stimulus.

The effect of this model is that the robot can remain tightly coupled to the environment yet still have a complex repertoire of motor actions with which to respond and act. It is important to note that the application of this model allows us to create a motor representation without creating any explicit models of the environment or the motor system.

1.2.4 Related Approaches

The general structure of this approach can be found in application in many disciplines of science. Clearly it is advantageous to decompose a problem in such a manner that its complexity can be approximated with a collection of simpler and better understood parts. The relationship of this organization to verbal language is direct, and there have been proponents of the idea that our verbal language is built upon representational structures originally used by the motor system (Allot 1995). This type of organizing principle has also been investigated with respect to visual processing (Riesenhuber & Poggio 1999). The work of (Arkin 1998) with behavior schemas is conceptually similar but deals with a domain of a larger granularity: behavior integration and representation.

1.3 Scope of Investigation

Humanoid motor control is a very broad and active area of research. For the purposes of this thesis, a small subset of the problems found in this domain will be studied.

Our approach is to propose a general framework for motor action organization based on the concept of motor primitives. Within this framework we will investigate various approaches to both learning basis elements of the gestural language and learning the grammatical structure of the language. Finally, we experimentally test our proposal through its application on the humanoid platform, Cog.

This work is done in the context of a larger problem: human motor mimicry and imitation. Learning through imitation stems from a developmental approach to humanoid robotics. It is essential to provide our humanoid platforms with the sensorimotor experiences necessary for the robot to learn motor tasks and the correlation between motor action and its impact on the physical world. During the critical developmental phases of infants and young children, imitation is an important tool for enabling exploration of the world through parental guidance.

A preliminary, yet essential, step towards human imitation is motor mimicry. Motor mimicry is a simple and direct kinematic emulation of caregiver's actions by the robot. The intent of the action is not understood. This thesis applies the concept of a gestural language to this task. Our goal is to implement a system which can accomplish coarse mimicry of broad human gestures. This allows the mimicry to occur without precise kinematic knowledge of the human. It also allows the robot to construct general gestures out of the language while avoiding the problem of precise trajectory formation needed in other domains such as manipulation.

The motor mimicry application can be more directly engineered through traditional approaches such as inverse kinematics (Craig 1989) or through the postural primitive approach used by (Williamson 1996). However, these approaches do not develop a representational structure. Consequently, they would be very difficult to extend to the more general problem of imitation. Imitation requires not just the ability to mimic motor actions, but also the ability to generalize the mimicry in the context of a goal. In order to perform this type of generalization, the motor action must be abstracted away from, and represented in a manner that allows it to be related to the intent of the human initiator.

1.4 Review of Thesis Contents

The thesis is organized as follows:

- **Chapter 2** provides a survey of background material supporting this work, including work in neuroscience, unsupervised learning, and humanoid motor control.
- **Chapter 3** describes in detail the implementation of the gestural language as currently implemented.
- **Chapter 4** describes an application of the gestural language to a specific domain: human motor mimicry.
- **Chapter 5** describes experiments evaluating the performance of the system in the motor mimicry task, and discusses the merits of the approach in general.
- **Chapter 6** provides conclusions and directions in which to extend this work.

Chapter 2

Literature Survey

2.1 Overview

This chapter surveys the wealth of literature that has provided the foundation for this thesis. We start by looking at the imitation framework for humanoid robotics. Then we discuss the split between dynamic and kinematic models of motor control as well as the motor primitive model gained from neurophysiological findings. We describe how these findings have been applied to the fields of graphical agents and humanoid motor control. We conclude with a brief survey of unsupervised learning techniques as they relate to this work.

2.2 Imitation and Motor Mimicry

The recent discovery of mirror neurons in macaque monkeys has spawned a swell of interest in imitation, mainly because they provide a neural location for imitation. Mirror neurons are distinguished by their activation due to the perception of a motor action and also by the execution of the same motor action. A motor action is a more holistic movement than a simple activation of a group of muscles. Instead it is tied to categories of movement such as grasping, reaching, and manipulation. A good introduction to this work can be found in (Gallese & Goldman 1998).

Imitation promises to be a powerful tool for teaching humanoid robots new skills.

The goal is to be able to teach by demonstration. There are a host of perceptual and cognitive obstacles to reaching this goal (Breazeal & Scassellati 1998). The problem from the motor control perspective is a bit more tractable.

There are a number of approaches to the motor mimicry problem. One method is for the robot to gather the complete kinematic information of the agent from the environment and then replicate it. This is a very difficult perceptual problem and most approaches require specialized hardware to accomplish it (Ude, Man, Riley & Atkeson 2000*b*). Another method is to use the 2D or 3D path of the end effector as the feature for mimicry. The use of this feature has biological support (Berniker 2000, page 20). This is a simpler problem, adequate for capturing the general nature of the movement. It benefits from not having to find the mapping from the kinematic structure of the agent to that of the robot. A combination of these approaches may prove to be the most robust.

2.3 Models of Motor Control

The nature of the controller used by natural systems to control motor actions is debatable. The existence of mirror neurons suggest that the representation of motor actions should be amenable to a representation of the perceived movement.

A primary problem in motor control is understanding the mechanisms used in moving a limb from one spatial location to another. One of the most fundamental distinctions that can be made in looking at control models is that between kinematic and dynamics based models (Flash, Hogan & Richardson 2000). In a kinematic model, the planning and representation of movements is in terms of joint angles, either in an egocentric or an exocentric frame of reference. In a dynamic model, it is the joint torques that provide the basis features of representation and planning. This distinction has important ramifications as to the proper representational form needed by the gestural language. Recent work has demonstrated that natural systems may decompose the task hierarchically, with trajectory planning occurring in a kinematic framework which is then used to compute the necessary joint torques.

This type of decomposition lends nicely to a common assumption made when modelling the musculature system. To a first order approximation, our muscles and individual joints behave as force controlling spring-damper systems (Williamson 1995). Consequently, we can model the controller as:

$$\tau = K(\theta - \theta_{setpoint}) + B(\dot{\theta}) \tag{2.1}$$

where τ is the commanded force, θ is the joint angle, and $\theta_{setpoint}$ is the equilibrium setpoint of the spring-damper system. This can be interpreted as a standard PD controller. This approximation is useful in that it provides a simple, linear method of mapping joint angles to joint torques. The model proposes that humans are controlling the setpoint of a spring and damper system when they move their bodies through a trajectory.

(Bizzi et al. 1984) supports the equilibrium point hypothesis with studies on rhesus monkeys. In the equilibrium point hypothesis, the system is predicted to be largely feed-forward. This holds especially for large inertia systems such as the arm in which the spring-damper model is more viable. Given the relatively large feedback latencies between the muscles controlling our limbs and our higher control centers, this proposition makes sense.

Additionally, (Flash et al. 2000) demonstrates that in controlling arm movements, we control our hand position from an exocentric frame of reference. They demonstrate that the central nervous system appears to optimize the smoothness (i.e., minimizing jerk), in Cartesian coordinates, of the movement over the course of its trajectory.

Finally, it interesting to look at the work of Cole and Gallagher (Cole, Gallagher & McNeill 1998) as it relates to developing a gestural langauge. They studied an individual, IW, who had lost all proprioceptive feedback from the neck down. IW had severe difficulty with instrumental actions (i.e. locomotion, reaching tasks) and depended almost completely on visual feedback to accomplish these tasks. However, his gestural actions were largely unaffected. Cole and Gallagher draw a number of interesting conclusions from this case:

- IW seems to compose his gestures out of a repertoire of known gestures.
- Gesture can be performed without any type of feedback.
- Gesture may be under control of a system other than the controller used for instrumental actions.

These conclusions suggest that gestural motor actions should provide a good match to the organizational method we are proposing.

Stein (Stein 1982) provides a comprehensive treatment on the hypothesized muscle parameters which may be controlled by the central nervous system. While there are significant alternative points of view on this issue, the work reviewed by Stein lends support to the premise that we can represent movement in a feed-forward kinematic framework for our gestural language.

2.4 The Motor Primitive Approach

Motor primitives provides a simple and decomposable explanation for a complex behavior. Though their appeal for humanoid robotics is clear, there are still questions remaining as to the extent that they can cover the diverse behavior of natural motor systems. In any case, motor primitives contribute a simple and decomposable model for humanoid motor control.

The motor primitives approach is hierarchical, allowing control to remain mostly local to the spinal cord. This type of local model has merit in that the long communication delays between the muscles and the brain do not need to be accounted for. Work by (Bizzi, Mussa-Ivaldi & Gistzer 1991), (Mussa-Ivaldi 1997), and others have attempted to understand this hierarchy.

Bizzi's approach involved microstimulation of a deafferented frog's premotor spinal cord. The stimulation triggered a motor activation in the frog's leg, and Bizzi recorded the force vector from various points in the leg workspace. The results showed a smooth force field across the workspace, drawing the leg to an equilibrium position. Different convergent force fields (CFF) were found at different stimulation points. Through

simultaneous stimulation Bizzi, found that the independent CFFs summed together to create a novel CFF with a new equilibrium point.

(Mussa-Ivaldi 1997) provides a computational model for the superposition of non-linear fields. This model explains the more complex movements we observe in natural systems as the result of simple scalar summation of the fields. He proposes that motor learning can be partially equated with the learning of these scalars.

More recently, (Thoroughman & Shadmehr 2000) have suggested that humans learn the dynamics of reaching movements through primitives that include a gaussian tuning function.

While support for the motor primitives model is growing, the debate continues about what form the primitives actually take. Theories range from synergistic muscle contractions, equilibrium postures, and simple movement strokes. This work investigates the application of the latter in the construction of a complex repertoire of motor gestures.

2.5 The Application of Motor Primitives

At this point it is beneficial to survey the influence motor primitives have had in the fields of robotics and computer graphics. In many cases, the work detailed here bears a direct relation to the previously cited works; in some cases it is only the gestalt of the model that has been utilized.

An early and influential application of motor primitives to a humanoid robot was done by Williamson(Williamson 1996). This work, closely linked to the earlier work of Bizzi (Bizzi et al. 1991), used a small set of equilibrium posture points as primitives. The primitives spanned the corners of the arm workspace and summation of the primitives provided a means to interpolate within the workspace. This characterization of primitives was proven effective in its simplicity, yet that very simplicity seems to have limited the ability to generate more complex motor behavior.

Mataric et al. have done the most extensive application of this approach to humanoid robots and simulated avatars. This work, also done within an imitation

framework, bears the strongest relation to this thesis and deserves closer attention.

In (Fod, Mataric & Jenkins 2000), the group takes an unsupervised learning approach to deriving the motor primitives from actual human motion data. The collected data was segmented into discrete motions and then projected onto a lower dimensional space via Principle Components Analysis (PCA) (2.6.1). A k-means approach to clustering was then employed. In the end, a set of generic, prototypical movements were discovered within the data. The quantitative results given are difficult to asses because they are highly dependent on the data set. However, the group did report that the data set could be fairly represented with around 100 primitives of 30 dimensions. The validation of this work was done in simulation and not on a physical robot, making it difficult to compare with the work here.

(Mataric, Zordan & Williamson 1999) draws upon the previously described work of Mussa-Ivaldi (Mussa-Ivaldi 1997) in formulating a joint-space force-field approach. This promising approach utilizes a nonlinear joint space controller, as opposed to the earlier work by Williamson with a linear controller. The nonlinearity reduces torque at high errors, allowing smooth transitions between equilibrium points. The controller primitive for joint i is defined as:

$$\phi_i = -k(\Theta_{actual} - \Theta_{desired})e^{-k(\Theta_{actual}-\Theta_{desired})^2/2} - k_d\dot{\Theta}_{actual} \tag{2.2}$$

This primitive defines a nonlinear force field attracting the joint to $\Theta_{desired}$. While this primitive attracts to a static point in kinematic space, superimposing a second primitive ψ_i allows modification of the trajectory followed to that static point. The two primitives are weighted by smooth step function ω_i and smooth pulse function υ_i. This yields the joint space controller:

$$\tau_i = \omega_i(t)\phi_i(t, \Theta_{actual}, \dot{\Theta}_{actual}) + \upsilon_i(t)\psi_i(t, \Theta_{actual}, \dot{\Theta}_{actual}) \tag{2.3}$$

This work provides a worthwhile comparison between this type of controller, a linear controller, and an impedance controller. The nonlinear approach, often called pulse-step control, has strong neurophysiological support (Flash & Hogan 1985, Kositsky

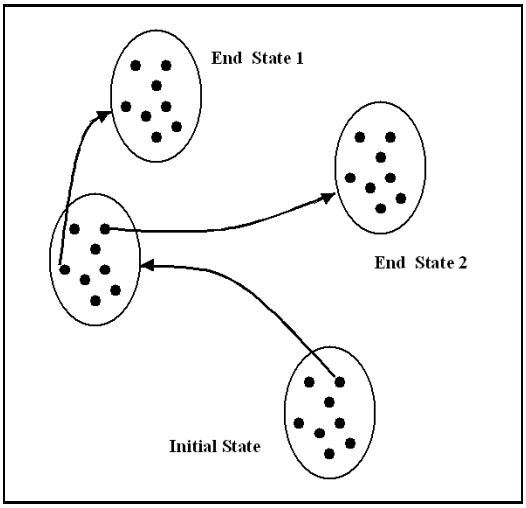


Figure 2-1: In the cluster chain model, trajectory transition points are clustered from a data set. A task trajectory can be found by searching the cluster graph.

1998). The simplicity of the primitive encoding for this controller, combined with its inherent interpolation between set points prove to be strong points of this approach.

Later work by this group places the primitive approach in an imitation framework (Jenkins, Mataric & Weber 2000). The hand trajectory, projected onto a 2D plane, is used as the fundamental unit of imitation. The primitives are hand coded with simple trajectories and the input trajectory is represented in terms of a sequence of primitives. An avatar is made to imitate this trajectory through an impedance controller. Their work presents a strong initial problem domain for application of the work described in the rest of this thesis. While their work is done entirely from an exocentric frame, our approach uses an egocentric frame of reference.

(Kositsky 1998) presents a decomposition of the pulse-step primitive into a cluster chain. A cluster chain is a method of specifying a generalized trajectory, as is depicted in Figure 2-1 . A reaching movement from Θ_{start} to Θ_{end} can be viewed as a series of via points along the joint trajectory. In Kositsky's model, a graph is built from a movement data set generated from a 2-DOF planar arm simulation. Valid trajectories for the arm are then encoded in the graph structure. Whether or not this approach can extend to a high DOF robot without an inordinate amount of data is an area of investigation for this thesis.

The computer graphics community has long been interested in simulating the elu-

sive human tempo of movement. Aside from time-consuming hand animation techniques, physical simulation and inverse-kinematic approaches bear on the work proposed here. Of particular interest are data-driven approaches using motion-capture techniques. (Bodenheimer & Rose 1997) presents a method of mapping motion capture to a wire-frame skeleton, and (Ude, Atkeson & Riley 2000*a*) extends this approach to a humanoid robot using a B-Spline wavelet encoding. Closely related to the work developed in this thesis is (Rose, Bodenheimer & Cohen 1998). Their work describes an organizing methodology for motions based on verbs (motions) which are controlled via adverbs (expressive parameterizations). By constructing extended motions using a verb graph similar to Kositsky's cluster chains, their work provides a framework for formalizing expressive behaviors.

2.6 Unsupervised Learning Techniques

(Hinton & Sejnowski 1999) provides a good survey of the primary approaches in unsupervised learning . As we will elaborate on later, the approach of this thesis is to derive the gestural language from a human movement data set using unsupervised learning techniques. In this manner the humanoid can learn the basis elements of the language and the allowable grammar with which they can be utilized.

Unsupervised learning is a suitable tool for this type of problem. It is especially useful in uncovering hidden features of a data set while not requiring prior knowledge or labelling of the data. Through techniques of clustering and dimensionality reduction, we can find the hidden features of a motor action data set. We would hope that these features bear a relationship to the actual motor primitives used in natural systems.

A disadvantage of this approach is that it typically requires a large data set, especially if the data lies in a high dimensional manifold (Hinton & Sejnowski 1999, Ch.1). Large human movement data sets are difficult to generate and hard to come by.

2.6.1 Dimensional Analysis

Dimensionality reduction is a standard statistical technique for obtaining compact representations of data by reduction of statistically redundant dimensions in the data. In general, this can viewed as a pair of maps:

$$\begin{aligned} g &: \Re^n \rightarrow \Re^m \\ f &: \Re^m \rightarrow \Re^n \\ n &> m \end{aligned} \tag{2.4}$$

This allows a mapping from n dimensions to m and back. The normalized reconstruction error, a measure of the success of the reduction on the data set, is:

$$\varepsilon norm = \frac{E_x[\|\mathbf{x} - f(g(\mathbf{x}))\|^2]}{E_x[\|\mathbf{x} - E_x\mathbf{x})\|^2]} \tag{2.5}$$

The classic approach in this area is Principle Components Analysis (PCA). (Hinton & Sejnowski 1999, Ch. 18) provide a succinct description of PCA:

> In PCA, one performs an orthogonal transformation to the basis of correlation eigenvectors and projects onto the subspace spanned by those eigenvectors corresponding to the largest eigenvalues. This transformation decorrelates the signal components, and the projection along the high-variance directions maximizes variance and minimizes the average squared residual between the original signal and its dimension-reduced approximation.

The simplicity of PCA has lead to its widespread application in engineering. However, it is a linear technique. It finds the lower dimensional hyperplane that best fits the higher dimensional data. Typically, even if the high dimensional manifold is smooth, it most likely is not planar.

This limitation led to a locally linear approach to PCA (Hinton & Sejnowski 1999, Ch. 18) which partitions the manifold into regions that can be better approximated as

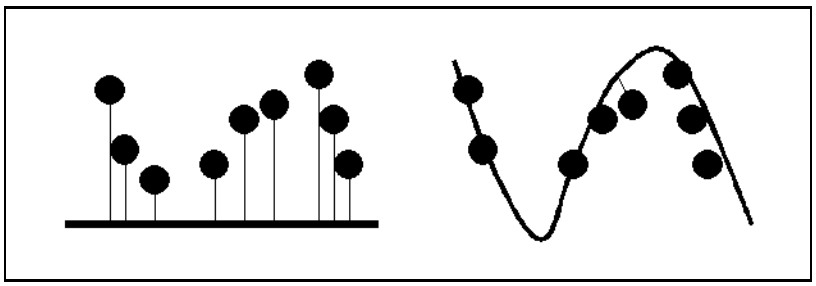


Figure 2-2: Linear and nonlinear dimensional analysis. The linear reduction maps the data down to a planar surface (left). The nonlinear analysis can conform the mapping to arbitrary smooth surfaces (right).

linear. The partitioning is typically done through some variation of clustering. Locally linear PCA improves the reconstruction error on nonlinear manifolds. However each subregion now has its own coordinate frame, and an attempt must be made to stitch the lower dimensional manifold together.

Nonlinear approaches to dimensionality reduction also exist in the form of neural-networks (Oja 1982) and the fitting of principle surfaces (Hastie & Stuetzle 1989). Most recently, (Roweis & Saul 2000) introduced the Locally Linear Embedding (LLE) technique. LLE has produced impressive results on smooth nonlinear manifolds. Importantly, it retains the topological structure of the manifold and embeds the data into a single global coordinate system. However, the reconstruction function f from Equation 2.4 is not easily obtained in LLE.

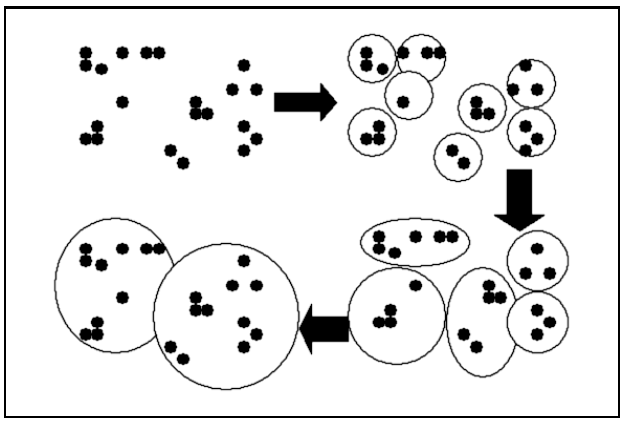


Figure 2-3: Clustering a data set. Given a distance metric, clustering iteratively groups points in the set. A threshold ϵ can be set such that all points lie within a hypersphere of diameter ϵ.

2.6.2 Clustering

Another unsupervised learning technique often used in conjunction with dimensionality reduction is clustering. Its goal is to group together similar inputs and let the groupings characterize the similarities in the features. As seen in Figure 2-3, clustering reduces the number of data points by sequentially joining clusters based on their similarity.

The success is highly dependent on the distance metric utilized and on the features chosen. A Euclidian distance metric is typically used. However, two data points may be qualitatively similar yet be far apart in their Euclidean distance (Hinton & Sejnowski 1999).

A standard clustering technique is UPGMA (Ooyent 2001). UPGMA takes the following approach: Take the cluster k which is formed by joining clusters $\{i, j\}$. The dissimilarity between k and a test cluster l is:

$$D_{k,l} = \frac{N_i D_{l,i} + N_j D_{l,j}}{N_i + N_j} \tag{2.6}$$

,where N denotes the number of members in the cluster and D is the dissimilarity. In practice, a combination of clustering and dimensional analysis has proven effective in finding the underlying structure in unlabelled data. As we will see in Chapter 3, this

approach can be applied to a data set of human movements. However it is an open question whether or not a smooth high dimension manifold is sufficient to capture the motor primitives employed by nature.

Chapter 3

Implementation

3.1 Overview

In this chapter we cover the details of the implementation. The general approach is to create a large motion data set and learn the gestural language from it. As we will describe, this is accomplished by:

- Acquisition of the motion data set using the humanoid robot.
- Segmentation and encoding of the data set into a form such that it can be treated computationally.
- Decomposition of the motions into kinematic subspaces.
- Derivation of the base elements, or primitives, of the gestural language using unsupervised learning techniques.
- Reconstruction of the data set in terms of the gestural language primitives.
- Development of the language grammar through transition graphs.

We are proposing a kinematic model of motor control. Accordingly, a gestural primitive is a specification of joint trajectories over time. A joint trajectory is a vector of equilibrium points moving a joint from Θ_{start} to Θ_{end}. In this chapter, as we

describe the implementation of the gestural language, we are essentially describing a method of organizing these trajectory vectors in a meaningful and useful manner.

3.2 The Development Platform

The work in this thesis is implemented on a humanoid robot. The embodiment of the robot in the physical world allows for a tight coupling to a complex environment not available in simulation. It is our belief that this is a critical component to achieving naturalistic movement.

The platform is a 26-DOF torso humanoid robot named Cog (Figure 3-1). Cog has a 7-DOF active vision head with two foveal CCD cameras and two wide angle CCD cameras. Each arm has 6-DOF: three in the shoulder, one at the elbow, and two at the wrist. The 3-DOF torso has pitch and roll at the waist and yaw at the shoulders. In addition, Cog has a pair of 2-DOF hands. All degrees of freedom are driven by DC servo motors. The work done in this thesis involved an 8-DOF kinematic chain starting at the hips and ending at the wrist of the right arm. The final wrist joint of the arm is not used. This is illustrated in Figure 3-1. The bilateral symmetry of the robot allows the gestural language to apply equally to both arms.

The force control hardware of Cog is critical for testing of biologically inspired motor control hypotheses. While the active vision head has only position feedback via optical encoders, the arms, torso, and hands all provide joint force feedback. The force feedback in the arms and hands is provided through Series Elastic Actuators (Pratt & Williamson 1995, Williamson 1995). The actuator places a spring element in series with the motor. Deflection in the spring allows compliance in the joint and provides a linear measure of force at the joint. Because the loads in the torso are much higher than in the arms, the torso utilizes torsional load cells in series with the motor. By using a standard PID controller on the force signal, we can control the force at each joint. Feedback position of each joint is also available, allowing us to experiment with control of joint position through many of the techniques described in Section 2.5. The work in this thesis uses the spring-damper control law (Equation 2.1) to control

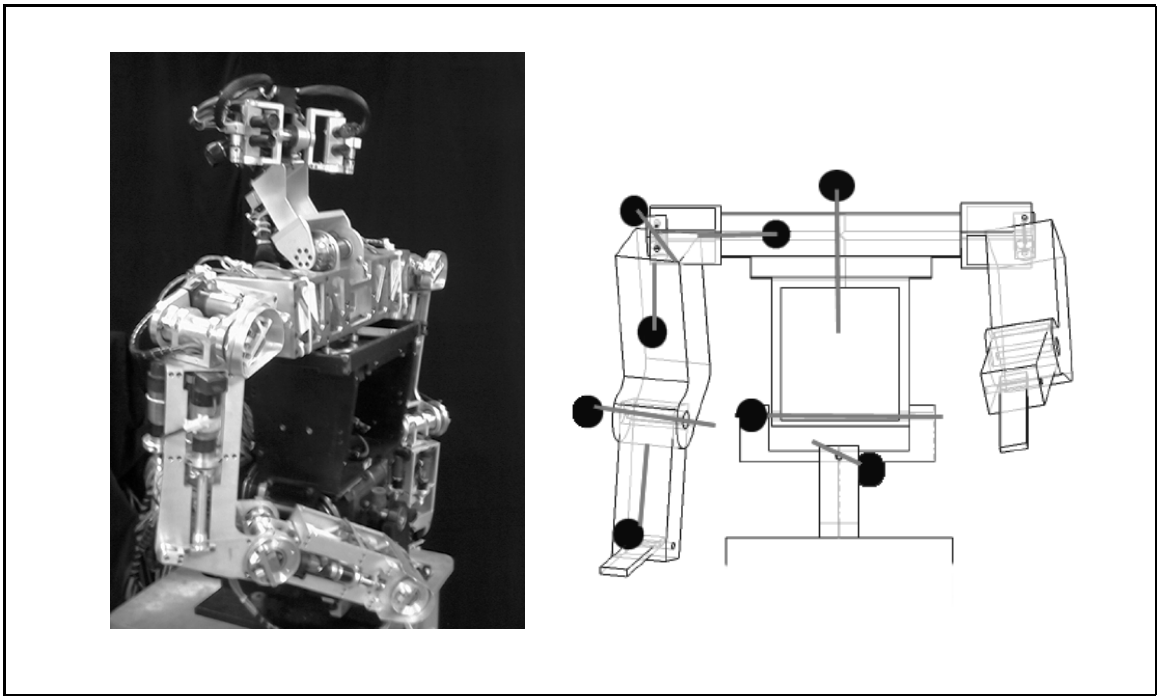


Figure 3-1: (left) Cog, the humanoid research platform. (right) A kinematic schematic of the 8-DOF used in this thesis.

position by setting the spring equilibrium point. Although we also investigated the nonlinear control law of Equation 2.2, it did not yield a significant improvement in the system performance. As of this writing, Cog's computational system consists of a 24 Pentium processors networked together and running the real time operating system, QNX4.

3.3 Dealing with the Data

We are taking an inherently data driven approach. We hope to learn the gestural language from movement data, as opposed to deriving the primitives by hand or by modelling the system as a complex controller. The quality of the data is critical to the success of the system. The data we are interested in is a time series of joint angles. This provides enough information to reconstruct joint velocities and Cartesian coordinate trajectories using a forward kinematic model.

3.3.1 Motion Capture

Acquiring the data invariably requires a motion capture system. There are number of commercially available systems employed in the animation industry. The most common technology is a suit, outfitted with optical or hall-effect position feedback sensors, which is worn by a human performer (Bodenheimer & Rose 1997).

Another technique involves mounting an array of cameras around a staged area. Markers placed on a performer allow tracking of limb position. A time series of joint positions can then be calculated off-line.

(Ude et al. 2000*b*) take a vision based approach to motion capture. Using minimal body markers, they automatically generate a kinematic model of the performer which is mapped to the body of an avatar or humanoid robot. This type of system is ideal from the imitation perspective. Unfortunately, the technology is not yet mature enough to be used without a large research investment.

Some motion capture systems can be tailored to match the kinematics of the robot or avatar. In most cases, a kinematic mapping between the performer and the avatar must be constructed. This causes a loss in the complexity of the motion captured if the kinematics of the robot are less complex than those of the performer.

To avoid these pitfalls and to simplify the process, we chose to use the robot itself as a motion capture device. This technique has several advantages:

- The hardware is already in place.
- The data is already in terms of the robot's kinematic structure.
- Physically unreproducible motions can't be generated.
- It allows the possibility to learn or adapt the gestural language online.

The unique virtual spring control on the robot arms allows a human to interact with the robot in a physical therapy type manner and consequently generate joint trajectories that are recorded for the data set.

A time series of joint positions is obtained by guiding the robot's hand through a trajectory and recording the joint angles via a data acquisition board. We used a

100Hz sampling rate of the joint position. Joint velocity was determined computationally and subjected to a low-pass filter.

The disadvantages of this approach will be discussed in Chapter 5.

3.3.2 Segmentation, Normalization, and Encoding

1. Θ denotes a joint angle and $\vec{\Theta}_k$ is a vector of joint angles.
2. n is the number of points used to specify a single joint trajectory.
3. M_i refers to a joint trajectory, encoded as a vector, in the data set.
4. S_j refers to a continuous sequence of joint trajectories in the data set.
5. DS refers to the entire motion capture data set.
6. $D(X, Y)$ is the Euclidean distance between vectors X and Y.

Given an 8-DOF kinematic chain, where the values of q and r are dependent on the data set , we can say that:

$$\begin{aligned} \vec{\Theta}_k &= \langle \Theta_1, \Theta_2, \Theta_3, ..., \Theta_n \rangle \\ M_i &= \langle \vec{\Theta}_1, \vec{\Theta}_2, ..., \vec{\Theta}_8 \rangle \\ S_j &= \langle M_1, M_2, ..., M_q \rangle \\ DS &= \{S_1, S_2, ..., S_r\} \end{aligned} \tag{3.1}$$

Figure 3-2: Notation and variables used in the data set encoding.

In Figure 3-2 we provide a notational reference for the data set encoding.

A difficult step in the data acquisition process is motion segmentation. The stream of joint positions needs to be segmented into individual motions. Because the gestural primitives will be short movement strokes, we need to exclude protracted motions as well as spurious short motions from the data set.

Ultimately, we want to be able to build protracted motion out of the primitives. To do this we also need to ascertain which motion strokes are continuous and which

are disjoint.

Several approaches to segmentation were tested: zero acceleration crossing (Bindiganavale & Badler 1998), zero velocity crossing, and sum of squares velocity (Fod et al. 2000). These methods assume that the segmentation point occurs when the trajectory uniformly experiences a change in direction or comes to rest. The zero velocity crossing approach provided the best results. For our 8-DOF movement, zero velocity crossing is defined as:

$$Z(t) = \Sigma_{i=1..8}(|\dot{\Theta}_i(t)|) < \epsilon \tag{3.2}$$

When a motion segment is distinguished, it is normalized to unit time and a standard dimensionality n. This is accomplished by encoding the time sequence using a cubic spline (Flannery, Teukolsky & Vetterling 1988). The spline is then evaluated at n evenly spaced points along its length, resulting in the following encoding for a movement M_i:

$$\vec{\Theta}_k = \langle \Theta_1, \Theta_2, \Theta_3, ..., \Theta_n \rangle \tag{3.3}$$

$$M_i = \langle \vec{\Theta}_1, \vec{\Theta}_2, ..., \vec{\Theta}_8 \rangle \tag{3.4}$$

The n elements of $\vec{\Theta}_k$ can be viewed as a trajectory of via points for the kth joint, occurring at evenly spaced time intervals. The spline encoding is beneficial in that it allows for simple and smooth interpolation between the via points.

The unit time normalization makes the motion segments invariant to time. Joint velocity was not included in the encoding because, it can be argued, velocity information is redundantly included in the position vector.

The trajectory of a single joint is a vector of size n. We represent a 8-DOF movement as a vector formed from the concatenation of the eight single joint trajectories. For the work covered here, the number of via points per joint trajectory, n, is between 3 and 5. Thus, we can also view M_i as a vector of dimensionality between 24 and 40.

While M_i can represent a simple movement stroke, we also want to formalize a

continuous sequence of movement strokes that occur in the data set. If the ending kinematic configuration of M_i and the starting configuration M_{i+1} match, then $\langle M_i, M_{i+1} \rangle$ is a continuous sequence. We represent a continuous sequence of q movement strokes as:

$$S = \langle M_1, M_2, ..., M_q \rangle \tag{3.5}$$

Thus the entire data set of r disjoint motion sequences can be represented as:

$$DS = \{S_1, S_2, ..., S_r\} \tag{3.6}$$

3.4 Learning the Gestural Language

3.4.1 Overview

Learning the gestural language from the data set is the central component of this thesis. This involves two general steps: deriving the gestural primitives from the data, and reconstructing the data set in terms of these primitives. The reconstruction step places the data in a representational framework. The framework organizes the data such that, given a perceptual stimulus, a suitable response gesture can be composed.

3.4.2 Kinematic Spaces

It is beneficial to describe the concept of kinematic spaces early on. In Figure 3-4 we provide a reference for the notation employed in this description. Kinematic spaces are a formalism we provide to decompose a complex kinematic chain into a hierarchy of less complex chains. Thus a 8-DOF chain can be viewed as the concatenation of two 4-DOF chains. A 4-DOF chain can be viewed as the concatenation of two 2-DOF chains, and so on. Figure 3-3 illustrates this concept.

Recall that M_i is the vector concatenation of 8 individual joint trajectories, $\vec{\Theta}_k$. We can map M_i onto a kinematic subspace $J^{(x:y)}$ such that:

$$M_i^{(x:y)} = \langle \vec{\Theta}_x, \vec{\Theta}_{x+1}, ..., \vec{\Theta}_y \rangle \tag{3.8}$$

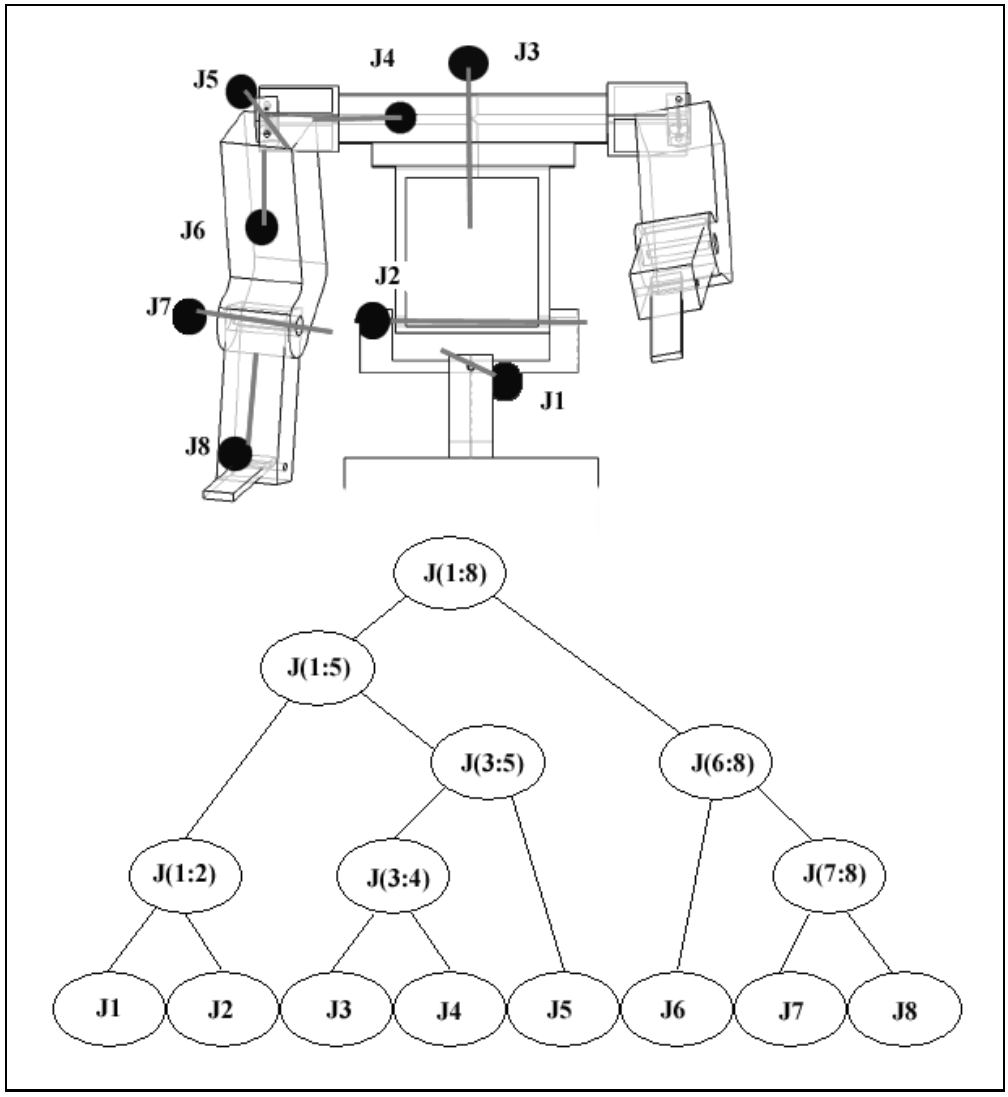


Figure 3-3: A hierarchical decomposition of the robot kinematic space. In the figure, $J^{(x:y)}$ denotes the kinematic chain starting at joint x and ending at joint y. The leaf nodes of the binary tree denote the single joint kinematic chains. We join adjacent nodes in the tree such that the root of the tree corresponds to the full 8-DOF kinematic chain.

1. We use the superscript $(x : y)$ to denote the kinematic chain from joint x to joint y on the robot.

2. Thus $M_i^{(x:y)}$ refers to the trajectory joints x through y take during the trajectory M_i.

3. We use $J^{(x:y)}$ to denote a kinematic space. A kinematic space is the workspace of joints x through y.

4. d is the dimensionality for a given kinematic space. For $J^{(x:y)}$, $d = (y - x + 1) \times n$. [a]

In our implementation, we use 8-DOF. Consequently,$M_i \equiv M_i^{(1:8)}$.
Example: if $n = 5$, then for kinematic space $J^{(4:6)}$, $d = 15$ and:

$$M_i^{(4:6)} = \langle \vec{\Theta}_4, \vec{\Theta}_5, \vec{\Theta}_6 \rangle \tag{3.7}$$

[a] n is defined in Figure 3-2

Figure 3-4: Notation and variables used in describing kinematics spaces.

For example: in our work, joints four through six are the first 3-DOF of the shoulder. The kinematic subspace $J^{(4:6)}$ describes the kinematic workspace of the shoulder and $M_i^{(4:6)}$ is the trajectory that the shoulder takes in the course of movement M_i. In our work, we use the 15 kinematic spaces described in Figure 3-3.

By decomposing M_i into a hierarchy of simpler movements, we can treat each of the simple movements individually and then recombine them to reform M_i. We illustrate this generalization process in Section 3.4.4. The intuition behind using the kinematic space decomposition is that canonical trajectories can exist in subsets of the entire kinematic chain. By finding these, we can use them to compose the more complex motions made by the full kinematic chain. It allows us to reuse and recombine the basic joint trajectory building blocks to create a motion instead of just finding the canonical full body motions.

As a final note on kinematic spaces, we should mention that we constrained the decomposition to match the morphology of the robot. We only allow consecutive

joints in the kinematic chain to form a kinematic space. Additionally, we tailored the decomposition to ensure that the 3-DOF shoulder, 3-DOF torso, and 2-DOF elbow each formed a separate kinematic space.

3.4.3 Clustering

1. ϵ is the user specified clustering threshold.
2. ζ is the user specified reconstruction threshold.
3. $\hat{\epsilon} = \epsilon \times d$ is the effective clustering threshold in a kinematic space of dimensionality d.[a]
4. $\hat{\zeta} = \zeta \times d$ is the effective reconstruction threshold in a kinematic space of dimensionality d.
5. P_k refers to a short motion trajectory (i.e. primitive) in the gestural language.
6. $P_k^{(x:y)}$ is a primitive in kinematic space $J^{(x:y)}$.
7. T_i is a binary tree.
8. Γ_i is a set of binary trees for a given kinematic space $J^{(x:y)}$.
9. L is the gestural language.

We should note that each $P_k^{(x:y)}$ is in fact a node on some tree T_i. Our implementation uses a 8-DOF kinematic chain partitioned into 15 kinematic spaces. This gives the following, where q is dependent on the data:

$$P_k^{(1:8)} = \langle \vec{\Theta}_1, \vec{\Theta}_2, ..., \vec{\Theta}_8 \rangle$$
$$\Gamma_i = \{T_1, T_2, ..., T_q\}$$
$$L = \langle \Gamma_1, \Gamma_2, ..., \Gamma_{15} \rangle$$

[a] d is defined in Figure 3-4

Figure 3-5: General notation and variables used in building the gestural language.

We use a clustering technique to find the initial gestural primitives in each kinematic space. As we will explain, the clustering approach represents each cluster as a binary tree, T_i. The root node of T_i is the centroid of the cluster, and it represents a

canonical gesture in the data set.

We begin the clustering process by first decomposing each trajectory $M_i^{(1:8)}$ into the 15 kinematic spaces described in Figure 3-3.

Now we look for canonical trajectories in each kinematic space by clustering the data. If a group of data points lie near each other within the space, then group should encompass similar types of motor actions. The assumption in clustering a kinematic space is that there is an underlying regularity in the generative process that created the data for the space, and therefore the distribution in the space is not uniform. The biomechanics of human movement should confine the kinematic space trajectories to small regions of the entire space. This is a hypothesis under investigation in this work.

The clustering algorithm works by building a binary tree over the similarity of the data elements in a particular kinematic space. It is described in Figure 3-6. The idea behind the algorithm is to find the centroid of each cluster in a kinematic space $J^{(x:y)}$. This is done by iteratively replacing each data element and its nearest neighbor with a single element, on the condition that the distance between the two elements is less than a threshold $\hat{\epsilon}$. This new element is the average of its two children.

When the algorithm terminates, the data for the kinematic space $J^{(x:y)}$ has been partitioned into a set of q binary trees:

$$\Gamma = \{T_1, T_2, ..., T_q\} \tag{3.9}$$

If $P_r^{(x:y)}$ is the root node of a tree T_i, then the kinematic trajectory that $P_r^{(x:y)}$ encodes is the canonical gesture for that cluster.

Although we can treat $P_r^{(x:y)}$ as the representative primitive of tree T_i, we can also choose some $P_k^{(x:y)}$ that is not a root node. As we will see in Section 3.4.4, this allows us to subtly vary the primitive representation for some trajectory $M_i^{(x:y)}$. In this manner we can adapt the representation to the task at hand.

1. For a kinematic space $J^{(x:y)}$.
2. Let Ψ and be $\bar{\Psi}$ be empty sets of binary tree nodes.
3. For each data set element $M_i^{(x:y)}$
   (a) create trajectory $P_i^{(x:y)} = M_i^{(x:y)}$.
   (b) let $P_i^{(x:y)}$ be a leaf node and add it to Ψ.
4. do forever:
   (a) set $\bar{\Psi}$ to be the empty set
   (b) for each $P_i^{(x:y)}$ in Ψ
      i. find a $P_j^{(x:y)}$ in Ψ such that $D(P_i^{(x:y)}, P_j^{(x:y)})$ is minimal and $i \neq j$
      ii. if the dissimilarity $D(P_i^{(x:y)}, P_j^{(x:y)}) < \hat{\epsilon}$
         A. remove $P_i^{(x:y)}$ and $P_j^{(x:y)}$ from Ψ
         B. create a new node Q and add it to $\bar{\Psi}$
         C. set the children of Q to be $P_i^{(x:y)}$ and $P_j^{(x:y)}$
         D. set the value of Q to be the average of its children: $(P_i^{(x:y)} + P_j^{(x:y)})/2$
   (c) add the elements of $\bar{\Psi}$ to Ψ
   (d) if no new elements were created on the last iteration or Ψ has only one element, terminate loop.

Figure 3-6: Algorithm for clustering of a given kinematic space $J^{(x:y)}$. The algorithm creates a set of ordered binary trees such that the root node of each tree is the centroid of a cluster in the data. Each cluster has a volume proportional to $\hat{\epsilon}$.

This approach to clustering has the following characteristics:

- In contrast to mixture model clustering techniques such as Expectation Maximization, where the number of clusters is specified a priori, the cardinality of Γ is dependent on the degree to which the data lies in clusters and therefore on the clustering threshold $\hat{\epsilon}$.
- So long as $\hat{\epsilon}$ remains small compared to the range of the space, averaging elements has the effect of creating a smoother encoded motion that is a generalization of its children.
- It can be shown that the greatest distance between any two elements $\{P_k^{(x:y)}, P_j^{(x:y)}\}$ in a tree of Γ, if the tree has a tree depth l, is $l \times \hat{\epsilon}$.
- Any node in a tree is the average of its children. Consequently we only need to store data for the leaf nodes of the trees. The vector values of the newly created nodes can be inferred from their children.

3.4.4 Data Set Reconstruction

After clustering across the set of kinematic spaces, we want to reconstruct the data set in terms of the discovered gestural primitives, or clusters. Recall that the data set was initially decomposed into separate kinematic spaces. We can now use the data set to link the spaces back together. In Figure 3-7 we provide a visualization of this process, though the general idea may be best illustrated by the following example:

- Take an original trajectory, $M_i^{(1:8)}$, from the data set.
- Consider the sub-trajectories: $M_i^{(1:4)}$ and $M_i^{(5:8)}$
- In each of the two kinematic spaces, we search the primitive trees to find the closest matching trajectories within a threshold $\hat{\zeta}$. As we search down a tree, we can think of it as shrinking the size of the cluster that we can use to approximate the trajectory. Assume we find $P_j^{(1:4)}$ and $P_k^{(5:8)}$.

- Because $M_i^{(1:8)} = \langle M_i^{(1:4)}, M_i^{(5:8)} \rangle$, we can now generalize this relationship by linking the two clusters together with a new primitive $P_l^{(1:8)} = \langle P_j^{(1:4)}, P_k^{(5:8)} \rangle$.

Intuitively, the reconstruction can be thought of as taking the relationships of the individual joint trajectories for an example motion and transferring those relationships to a more general set of canonical joint trajectories.

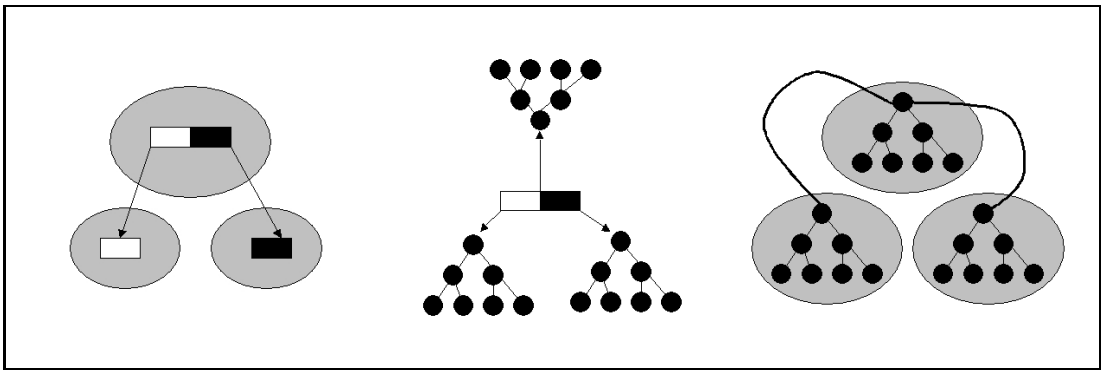
To realize the intuitive goal, we first need a method for mapping an original motor action onto a primitive. This algorithm is described in Figure 3-8. The process amounts to searching the set of trees $\Gamma = \{T_1, T_2, ...\}$ for the closest Euclidean distance match within a specified threshold $\hat{\zeta}$. If $\hat{\zeta} = 0$, then an exact match will be found because the leaf nodes of T_i are in fact the original motion vectors. As $\hat{\zeta}$ increases we are trade off a larger discrepancy in the mapping for a higher level of generalization. If $\hat{\zeta}$ is very large, then we will only be mapping to the root nodes of Γ.

Now we can use this mapping to reconstruct the data. To do this, we simply extend the previous example. The example linked together primitives in $J^{(1:4)}$ and $J^{(5:8)}$ through the space $J^{(1:8)}$. We can use the same approach to all levels of the hierarchy of kinematic spaces, starting with the single joint kinematic spaces $J^{(i:i)}$. In doing so, we will link together the kinematic spaces in a manner that:

- Generalizes the motor action: If we link together primitives $P_j^{(x:y)}$ and $P_k^{(y+1:z)}$ in space $J^{(x:z)}$, then we are also linking together all children nodes of $P_j^{(x:y)}$ and $P_k^{(y+1:z)}$. This allows novel movements not originally in the data set.

- Guarantees that this generalization is valid: The motion composed by $\langle P_j^{(x:y)}, P_k^{(y+1:z)} \rangle$ is valid because it was formulated from actual movement data. The novel set of movements formed by the generalization will also be valid because the children of $P_j^{(x:y)}$ and $P_k^{(y+1:z)}$ must be similar to their parents by the method of clustering.

At this point, by using the idiom of clusters and kinematic spaces, we have constructed the initial framework for the gestural language. Further implementation details extend, optimize, and apply the framework.



(a) (b) (c)

Figure 3-7: Reconstructing the data set. (a) A motor trajectory is decomposed into its kinematic spaces. (b) An equivalent primitive (or cluster) for each trajectory is found by searching the trees of the space. (c) We generalize the original trajectory to the clusters by linking the clusters together.

1. Given $M_i^{(x:y)}$, a motor action in kinematic space $J^{(x:y)}$
2. Given $\Gamma = \{T_1, T_2, ...\}$, the set of binary trees for $J^{(x:y)}$
3. Let Ψ be the set of root nodes $\forall T_i$ in Γ
4. do forever:
   (a) find the $P_j^{(x:y)}$ in Ψ such that $D(M_i^{(x:y)}, P_j^{(x:y)})$ is minimal
   (b) if $D(M_i^{(x:y)}, P_j^{(x:y)}) < \hat{\zeta}$ or $P_j^{(x:y)}$ is a leaf node, terminate loop and return $P_j^{(x:y)}$
   (c) else let Ψ be the children of $P_j^{(x:y)}$

Figure 3-8: An algorithm for mapping a motor action to a gestural primitive. It performs an ordered binary tree search for $P_j^{(x:y)}$, the closest matching trajectory to $M_i^{(x:y)}$ within a threshold $\hat{\zeta}$.

3.4.5 Dimensional Analysis

It was noted in Section 2.6.1 that dimensional analysis is often used in conjunction with clustering techniques in unsupervised learning. Dimensional analysis is useful in cases where the data lie on a smooth manifold. Otherwise, application of the technique results in a loss of local topology.

In our domain, a technique such as PCA could be used in a couple of ways. One application is to take the data set as a whole, lying in the highest kinematic space, and use PCA to map down to a lower dimensional space. In this space we could then build the gestural language. An interesting outcome of this approach would be to find that the projection exaggerates the similarity and dissimilarity of the data, making the derivation of the gestural primitives more exact.

Another approach is to apply PCA to each individual kinematic space after construction of the gestural language. In this way the language can first be built without any loss of topology. Successful construction of the language depends heavily on topological relationships in the data. Applying PCA post-hoc to each kinematic space provides a means for reducing the dimensionality of the space. This increases the speed of search through the space. Using PCA in this manner is similar to methods developed for locally linear PCA.

We investigated both approaches and discuss the results in Chapter 5. Neither approach changes the basic framework for the gestural language.

3.4.6 Transition Graphs

A second extension to the gestural language that has been explored is the construction of transition graphs. Following the work of (Kositsky 1998), a transition graph is a directed graph which encodes valid sequences of motions. This is a fairly simple notion. If we have a continuous motion sequence:

$$S_j = \langle M_1, M_2, ..., M_q \rangle \tag{3.10}$$

then we can map each M_i to a primitive P_k and form a graph with nodes $\langle P_1, P_2, ..., P_q \rangle$ and links between the nodes P_k and P_{k+1} for $0 > k < q$.

We can allow for repetitive and oscillatory motions by permitting cycles in the graph. For example, if an arm extension is followed by arm contraction, then we can represent this as a graph edge between the two representative primitives. The edges are weighted according to their frequency in the data set. Kositsky's work built the graph around clusters in the arm workspace, using a velocity based encoding. Consequently a motion in progress could terminate in a variety of locations depending on the graph. In this work we are linking together larger motion strokes so that upon completion of one primitive a naturally following second primitive may be executed.

To do this we use the approach detailed in Figure 3-9. The transition graph links together the gestural primitives based on transitions found in the actual data. However, because we are linking together clusters of trajectories and not individual trajectories, we are essentially generalizing a single example from the data to all members of the clusters.

3.4.7 Feature Search

The final step in the implementation is to perform a feature search on the gestural language. This search constitutes the actual application of the language to a real world task.

Feature searching is essentially the following idea: Given a perceptual feature generated by an external process, use the gestural language to construct an appropriate motor action in response to the feature. The appropriate motor action is determined by searching the binary trees of the gestural language for the primitive, or sequence of primitives, that best match the perceptual input.

To begin the search, we want to enable, or activate, only those primitives which have an initial joint configuration similar to the current joint configuration of the robot. This prevents the robot from having to make large interpolations between the current joint state and the primitive start state.

We use this criteria to activate the leaf node primitives in the 1-DOF kinematic

1. Given data set $DS = \{S_1, S_2, ...\}$
2. Given the gestural language $L = \langle \Gamma_1, \Gamma_2, ..., \Gamma_{15} \rangle$ over 15 kinematic spaces where Γ_j is the set of trees $\{T_1, T_2, ...\}$.
3. Given the empty graphs $G = \langle \Lambda_1, \Lambda_2, ..., \Lambda_{15} \rangle$.
4. For each $\langle M_i^{(1:8)}, M_{i+1}^{(1:8)} \rangle$ in S_k
   (a) For each of the 15 kinematic spaces $J_r^{(x:y)}$ and $r = 1...15$
      i. Map $\langle M_i^{(1:8)}, M_{i+1}^{(1:8)} \rangle$ onto $J_r^{(x:y)}$ to get $\langle M_i^{(x:y)}, M_{i+1}^{(x:y)} \rangle$
      ii. Find the closest gestural primitives to $\langle M_i^{(x:y)}, M_{i+1}^{(x:y)} \rangle$ to get $\langle P_l^{(x:y)}, P_{l+1}^{(x:y)} \rangle$
      iii. Add $P_l^{(x:y)}$ and $P_{l+1}^{(x:y)}$to Λ_r if they are not already present
      iv. Add an edge from $P_l^{(x:y)}$ to $P_{l+1}^{(x:y)}$ with a weight of 1. If the edge is already present, increment the weight by 1.
   (b) Divide the weights of all edges in Λ_r by the number of edges in Λ_r

Figure 3-9: Algorithm for building a weighted transition graph from the gestural language.

spaces. Only leaf node primitives which have an initial joint configuration close to the current joint configuration of the robot are activated. This activation is then trickled up the trees of the kinematic space through parent-child relationships. Recall that the data set reconstruction step linked the kinematic spaces together. Thus we can spread the activation across all kinematic spaces as well. After we trickle the activations across the gestural language, we have a set of active primitives to search against given the perceptual feature.

The search involves an evaluation metric $F(A_i, P_k)$ which computes the response of primitive P_k to perceptual feature A_i. This metric is task dependent. To limit the size of the search, task specific heuristics are employed such as preference for larger kinematic spaces, etc. In Chapter 4 we develop an evaluation metric for the motor mimicry task.

An alternative method to generating the activation set entails using the current robot state to index into the transition graph. Edges leaving the activated nodes lead to the activation set.

Chapter 4

Application of the Gestural Language

4.1 Overview

At this point we describe the application of gestural language to a real task on the humanoid robot Cog. It is fair to claim that the gestural language adds a level of complexity to the robot that is not necessary for some tasks, and not appropriate for others. For example, tactile manipulation is a task that requires a tight sensory feedback coupling. The type of kinematic feed-forward representation proposed here would not apply well to that domain. The postural reflexes used by the body to maintain balance are another domain where a representational framework may not be of use. However, there is an interesting and important set of problems where the gestural language can be applied.

4.2 Application Domains

The power of a representational framework for motor actions is that it provides a level of abstraction necessary for multi-modal learning. If the robotic system is to form a correlation between perception and action, then providing a means to compare "apples to oranges" is a necessary first step. Two areas of research where this representational

power could be applied are nonverbal communication and imitation.

The appearance of nonverbal communication in young infants marks an important developmental stage. (Shankar & King 2000) notes that at around four months old, the infant passes into the"immediate social world", a world of subtle communicative gestures tightly coupled to the infant caregiver. Perception of the caregiver's communicative gestures is an active area of research in humanoid robotics (Brezeal 2000). The other side of the equation, the communication of the internal state and desires of the robot, hasn't received as full of a treatment. In this domain, the gestural language can serve as a substrate to learn communicative behaviors based on perceptual stimuli and caregiver reinforcement. As reinforcement signals are provided by the caregiver, novel mappings between the perceptual features and the gestural primitive can be formed. The gestural language can then provide a basis motor competency from which perception-to-action learning can bootstrap.

In fact, the concept of a gestural language integrates well with the behavioral decomposition of complex behavior that is predominant in humanoid robotics (Brooks et al. 1998). It partitions the motor action space into subspaces of motor behaviors which exhibit global similarity. We can think of the language providing a set of motor behaviors such as "reach-in-direction" or "lean-forward". This type of behavioral decomposition lends well to developing a repertoire of gestural behaviors that could be used by the robot in nonverbal communication.

A second interesting application domain is in imitation, and this is the domain that we explore in this thesis. We have already discussed the imitation and motor-mimicry framework in Section 2.2. Scassellati (Breazeal & Scassellati 1998) provides a strong developmental framework for imitation in humanoid robotics. A fundamental component of this work is the development of mechanisms for joint-attention between the caregiver and the robot. As an outgrowth of this work , Scassellati has developed a wide range of visual perceptual abilities for Cog. Of particular interest is the theory of body module (ToBY)(Scassellati 2001). This module provides Cog with a sense of naive physics about the world, and importantly, the ability to distinguish between animate and inanimate trajectories. The ToBY module uses the spatial and

temporal features of the visual input to perform the discrimination. Using motion correspondence techniques, a moving object in the visual field of the robot provides an initial trajectory for the system. Then, by applying a mixture of experts to detect features such as trajectory energy, acceleration, and elastic collisions, the trajectory is categorized as animate or inanimate. A complete description of this work can be found in (Scassellati 2001).

Scassellati's work provides an essential perceptual cue for imitation. By integrating this cue with the gestural language, we can develop a rudimentary form of motor mimicry. The robot can then approximate animate trajectories with its own body. Through the gestural language, the trajectory is no longer represented in terms of the visual modality, but instead in terms of an egocentric framework: its body and its ability to move its body in the world. This can be accomplished in the following manner:

- The mimicry behavior receives a pixel-coordinate trajectory from the perceptual system.
- Feature search (Section 3.4.7) is employed to find a gestural action that best matches the trajectory.
- The gestural action is either executed or perhaps, in the service of a learning task, inhibited.

4.3 Application to the Motor Mimicry Task

Applying the gestural language to the motor mimicry task involves developing the evaluation metric $F(A_i, P_k^{(x:y)})$ which computes the response of primitive $P_k^{(x:y)}$ to perceptual feature A_i (Section 3.4.7). A_i is an animate trajectory over time, in pixel coordinates:

$$A_i = \langle\ (x_1, y_1, t_1), (x_2, y_2, t_2), ..., (x_n, y_n, t_n)\rangle \tag{4.1}$$

To find the gestural primitive that best matches A_i, we first must find $P_k^{(x:y)}$ in terms of the pixel coordinate frame of A_i. This algorithm is explained in Figure 4-1.

1. Given $P_k^{(x:y)}$, a gestural primitive in $J^{(x:y)}$.
2. Given $\vec{\theta}_R$, the kinematic state of the robot at the current time.
3. Given $FK(\vec{\theta})$, the forward kinematic function for the robot on joint vector $\vec{\theta}$.
4. Project $P_k^{(x:y)}$ up to the full kinematic space $J^{(1:8)}$ by setting the trajectories for joints $\langle 1, .., x-1 \rangle$ and joints $\langle y+1, ..., 8 \rangle$ to be constant at their current position, $\vec{\theta}_R$, giving $P_k^{(1:8)}$
5. Evaluate $FK(P_k^{(1:8)}(t))$ at each of the trajectory via points in $P_k^{(1:8)}$, giving the 3-dimensional Cartesian trajectory $Z_k(t)$.
6. Project $Z_k(t)$ onto the 2-dimensional frontal plane in the direction of the robot gaze, given by $\vec{\theta}_R$. This gives $\widehat{Z_k(t)}$, the mapping of $P_k^{(x:y)}$ onto the robots visual coordinate frame.

Figure 4-1: Algorithm for projecting a gestural primitive onto a visual coordinate frame.

Our evaluation metric uses a forward kinematic model of Cog, which we specified in Denavit-Hartenberg notation (Craig 1989). Using the forward kinematic model, we can map the joint trajectory of a primitive to an end-effector trajectory in Cartesian coordinates. However, doing this requires the full kinematic space to be specified. Consequently, we use the current kinematic state of the robot to project lower kinematic spaces up to the full space.

The Cartesian space trajectory, $Z_k(t)$, obtained from the forward model is projected onto the two dimensional plane perpendicular to the robot's line of sight, giving us the trajectory $\widehat{Z_k(t)}$. Now $\widehat{Z_k(t)}$ and A_i lie in the same coordinate frame. The next step is to normalize the two trajectories for comparison. This is accomplished by time normalizing both to unit time using a spline encoding similar to the technique described in Section 3.3.2. Then we subtract the mean from both and normalize the size of both trajectories so that they lie within the unit circle. This normalization scales $\widehat{Z_k(t)}$ by a factor α. The factor α will be used later as a means to heuristically guide the search.

By taking the Euclidean distance between $Norm(\widehat{Z_k(t)})$ and $Norm(A_i)$, we come upon our evaluation metric $F(A_i, P_k^{(x:y)})$, determining the response of primitive $P_k^{(x:y)}$ to feature A_i.

At this point,the motor-mimicry task can be accomplished by the execution of the feature search algorithm explained in Section 3.4.7. The search finds the closest activated gestural primitive to the visual trajectory. As a means of guiding and limiting this search, a few task specific heuristics are used. These are:

- Biasing the response $F(A_i, P_k^{(x:y)})$ towards higher kinematic spaces. This is because we prefer that the robot makes full bodied motions.

- Filter the primitives in the activated set based on the rescaling parameter α. This biases the search away from short, small gestures and towards large, longer gestures.

- Limiting the actual kinematic spaces searched so that we prefer full body gesture, full arm gestures, or full torso gestures.

We should note that by including all kinematic spaces in the evaluation metric, we can satisfy the mimicry task through any given kinematic chain of the robot. In addition, by exploiting the bilateral symmetry of the robot we are able to apply the gestural language to either arm. These characteristics provide the interesting property that Cog is able to mimic the perceptual input with either hand or by using only the torso.

The final result of this application is that given a human facilitator generating a random hand motion or some other animate trajectory in front of the robot, the robot attempts to mimic the motion by executing a gestural primitive. We hope this basic functionality will set the stage for more complex mimicry and learning possibilities. We will look at the performance of the system in the next chapter.

Chapter 5

Experiments and Discussion

5.1 Overview

At this point we are ready to move beyond a formulation of the motivation and the framework. In this chapter we look at the implementation of the system and assess its performance. We begin with an analysis of the robot-generated motion data set and its accessibility to dimensionality reduction techniques. We describe experiments to assess the formation of gestural langauge from this data set and we analyze the gestural language in application. Finally, we conclude with a general discussion of the issues uncovered during the experimentation process. In Figure 5-1 we provide an overview of the system used in these experiments.

5.2 Looking at the Data

The approach described in this thesis is data driven, and as such, the nature of the data is of critical importance to the success of this work. The overarching assumption is that the data behave nicely in the following manner:

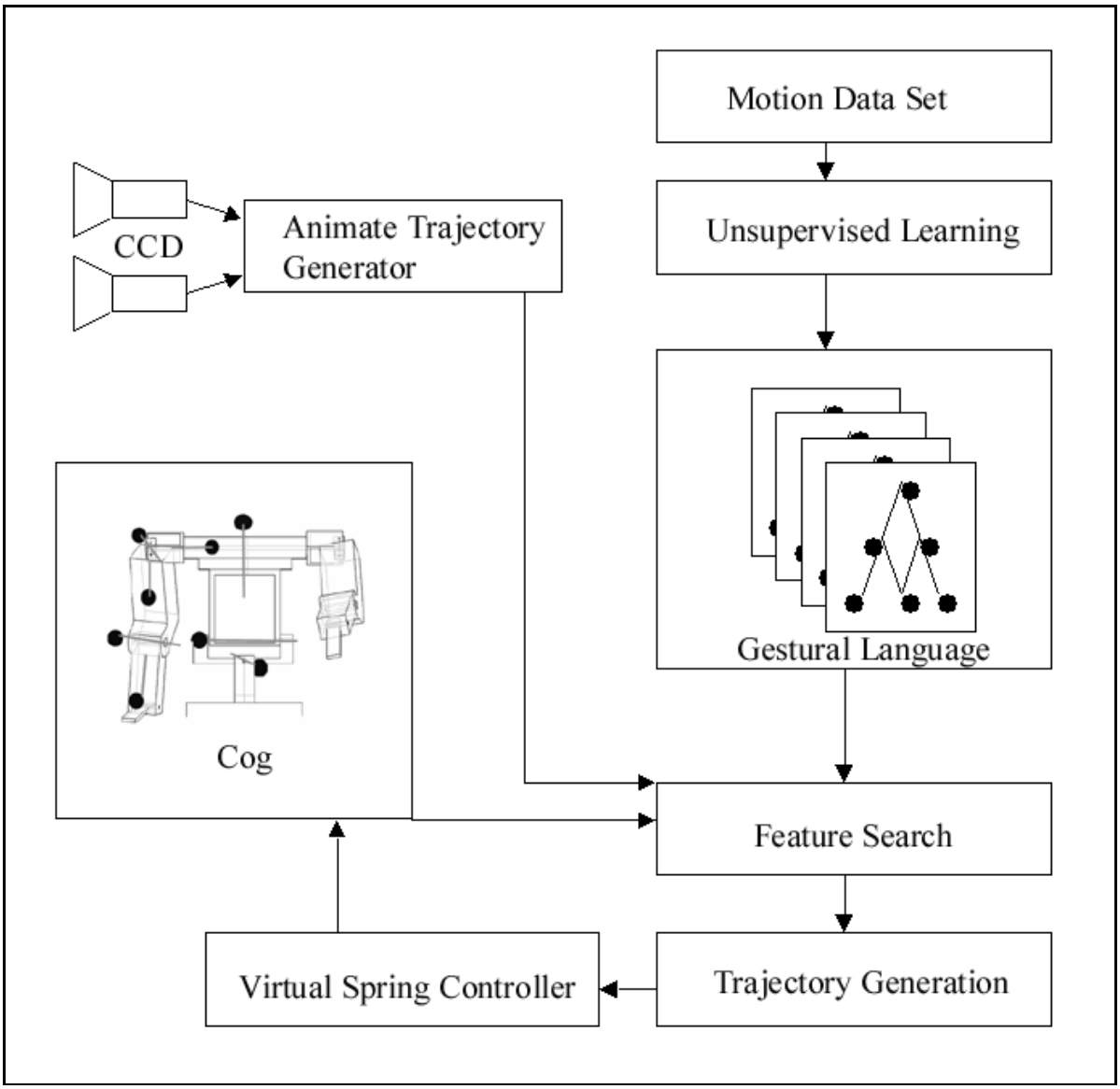


Figure 5-1: Overview of the system employed for the motor mimicry task.

- It exhibits local linearity and consequently global smoothness.
- It is derived from physical processes that exhibit strong regularity. This allows for a more compact description of the data than provided by the original space in which it was collected.
- There exists a set of features in the data that provide an encoding which is amenable to generalization. The encoding allows a distance metric to evaluate the similarity and dissimilarity of motions.

Using the method described in Section 3.3.1, we acquired approximately 500 unique gestural motions from the robot. The 8-DOF kinematic chain used in the data capture was described by 15 kinematic spaces. This resulted in a data set of 7500 data points from which the gestural langauge was built. While the kinematic space decomposition creates redundancy in the data, it is ultimately removed when the data set is reconstructed (Section 3.4.4).

Unfortunately the high dimensionality of the data makes it difficult to assess in terms of the qualitative objectives outlined. In Figure 5-2 we provide two and three dimensional views of the data by representing each single joint trajectory in terms of the start and end joint angle. Though the global motions are lost, this view does suggest that the data lie in a well behaved distribution. In Figure 5-3 we look at the standard deviation of the data set encoded as a 40 dimension vector. We see that there is a large disparity in the relative standard deviations across joints. This suggests that a subset of the 8-DOF kinematic chain may capture the predominant aspects of the motions.

5.2.1 Dimensional Analysis

The effectiveness of linear dimensionality reduction techniques such as PCA are largely dependent on the local or global linearity of the data. Following the approach of (Fod et al. 2000) we used PCA globally across the entire data set to assess the degree to which the data lie on a plane in a lower dimensional space. In Figure 5-4 we

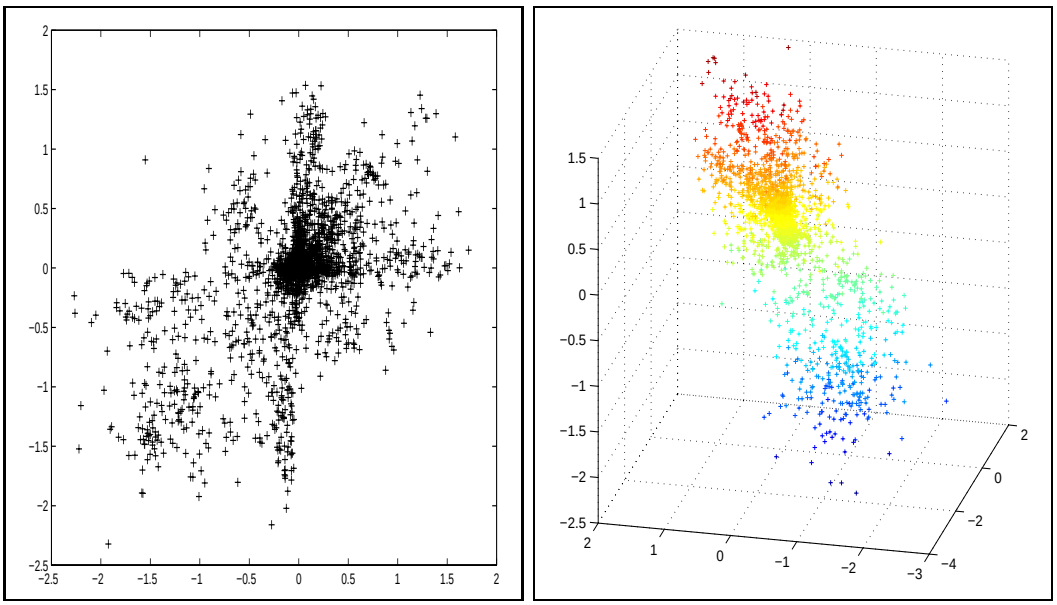


Figure 5-2: Plot of the entire data set (4016 single joint trajectories). (a) 2D projection of the start and end joint postions. (b) 3D projection including a trajectory via point.

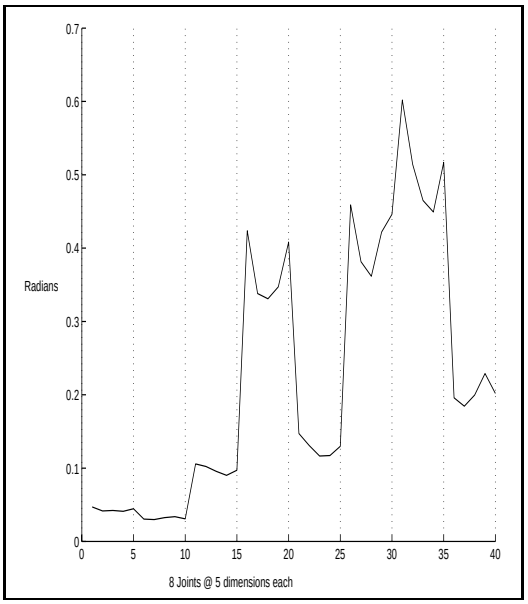


Figure 5-3: The standard deviation of the data set. A motion is represented as a 40-dimensional vector, with five dimensions for each of the 8-DOF of the robot.

look at the change in reconstruction error (Equation 2.5) as we vary the dimensionality of the data set encoding. The results suggest that our standard 40-dimensional encoding of a motion can be projected, with minimal error, onto a 20-dimensional space. We did not investigate a locally linear PCA approach because of the need for a global coordinate frame.

We also investigated LLE (Section 2.6.1) as a means to find a lower dimensional embedding of the data. The high PCA reconstruction errors found below a 20-dimensional encoding suggested that below this threshold, the data set is inherently nonlinear. LLE does not provide a simple mechanism for reconstructing the data. In PCA, we simply multiply the lower dimensional vector by the transpose of the eignevector matrix to return to the higher dimensional space. For LLE, reconstruction would involve learning the mapping from the lower to the higher dimension. We have not attempted this and consequently were not able to compare LLE to PCA based on the reconstruction error.

Instead we devised a comparison metric to measure the loss of local topology in using each technique. To measure the retention of local topology we look at the displacement of the nearest neighbors for each data point. For a data point X, we want the k nearest neighbors of X, $\{\beta_1, \beta_2, ...\beta_k\}$, to remain near X after it has been mapped to $\hat{X}$ in a lower dimension. Thus, if $D(X, Y)$ = Euclidean distance between X and Y, and ε is the topological error, then:

$$\begin{aligned} \tau &= \sum_{i=1..k} D(X, \beta_i) \\ \hat{\tau} &= \sum_{i=1..k} D(\hat{X}, \hat{\beta}_i) \\ \varepsilon &= \hat{\tau}/\tau \end{aligned} \tag{5.1}$$

Figure 5-5 compares PCA and LLE on the data set using this metric. This analysis demonstrates a clear advantage, at least in terms of this metric, of LLE over PCA. However, the difficulty in reconstructing the data with LLE remains a significant obstacle.

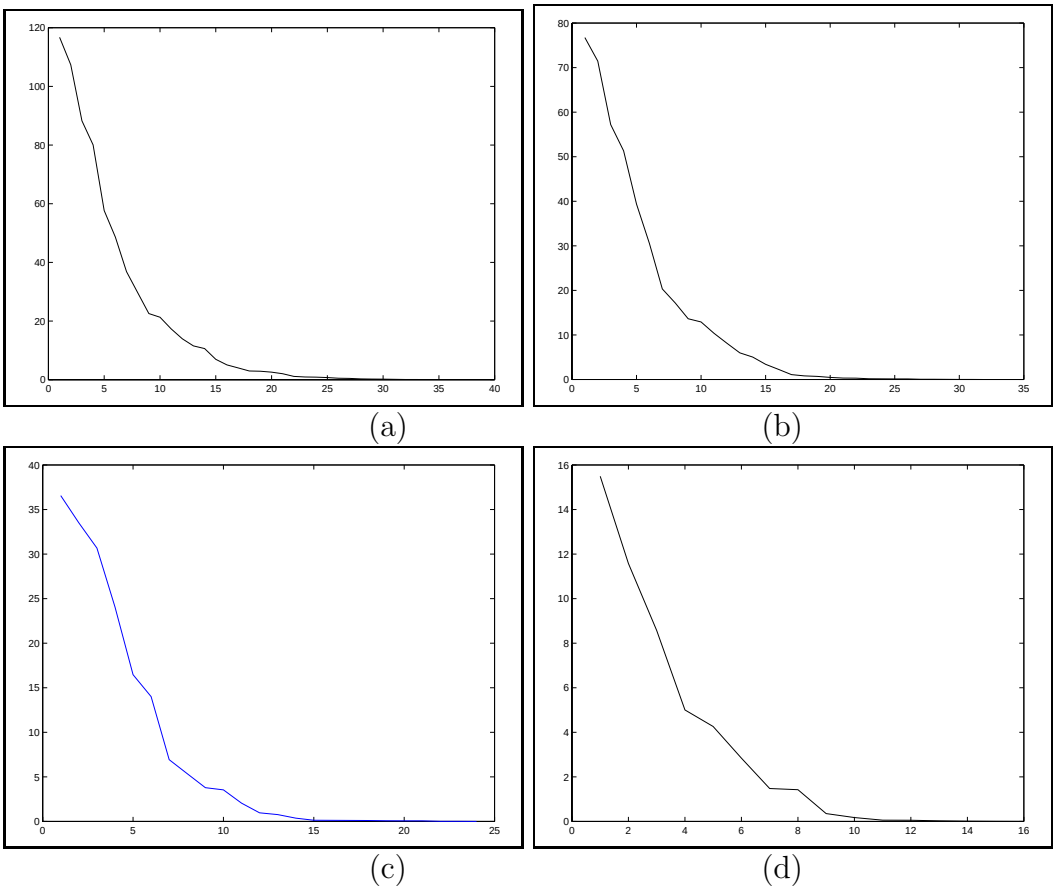


Figure 5-4: PCA reconstruction errors on the entire data set mapped into full kinematic space. (See Equation 2.5). Plotted are the errors at varying dimensional encodings of the data set.
(a) 22 eigenvectors gave fair reconstruction for a 40-dimensional vector representation.
(b) 20 eigenvectors gave fair reconstruction for a 32-dimensional vector representation.
(c) 15 eigenvectors gave fair reconstruction for a 24-dimensional vector representation.
(d) 10 eigenvectors gave fair reconstruction for a 16-dimensional vector representation

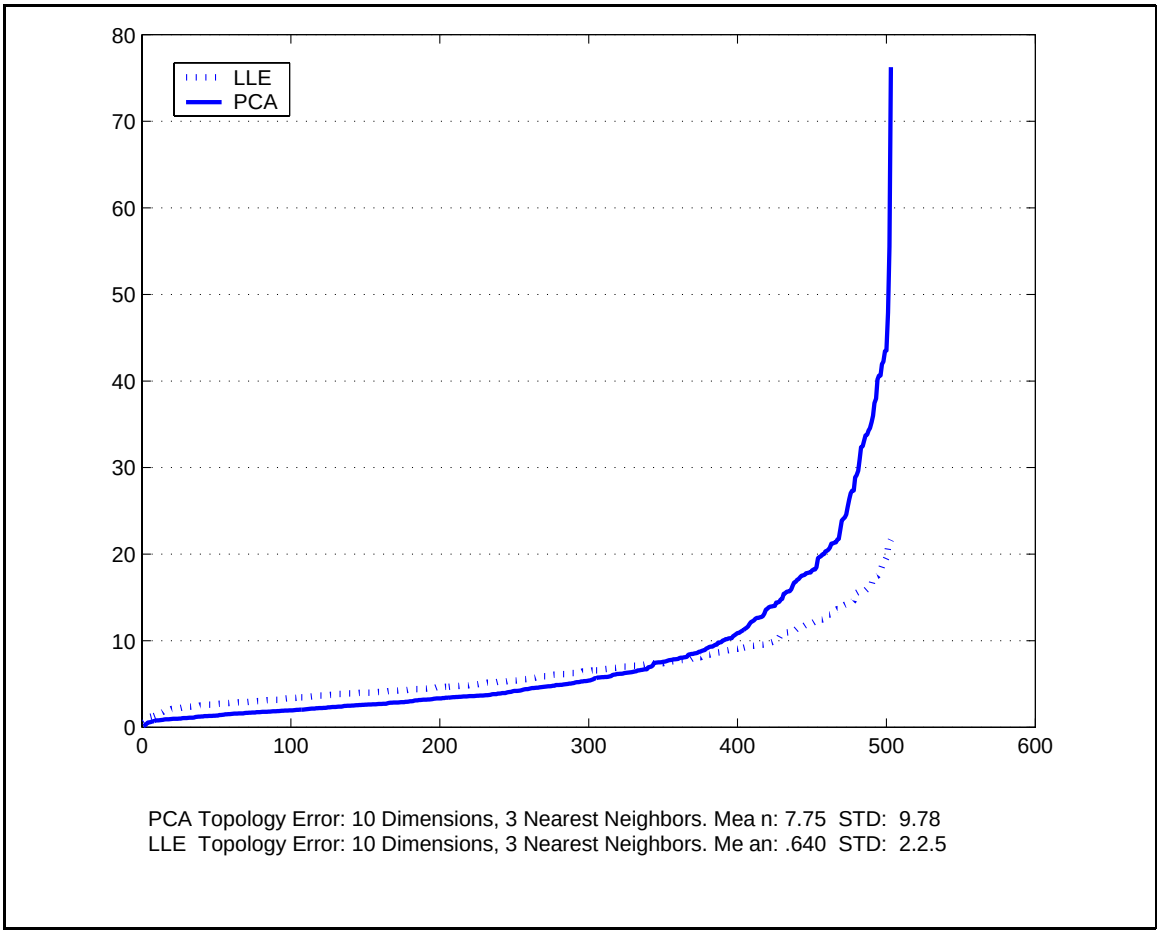


Figure 5-5: Analysis of the loss of topology in using LLE and PCA to reduce from 40 dimensions to 10 dimensions. The graph shows the induced change in the normalized distance of each data point from its nearest neighbors. The error on the 500 data points has been sorted in ascending order.

5.3 Gestural Language Analysis

It is difficult to find a suitable measure with which to analyze the gestural language. Perhaps the best method of analysis is to study the performance in application, which we will do shortly in Section 5.4. However, it is instructive to try to tease out the underlying structure of the gestural language that is formed from the data. The most direct method of doing this is to look at the nature of the primitives found in the data and how they vary as we vary the parameters used to build the language.

In Figure 5-6 we provide a visualization of some of the primitives found. They are rendered in terms of the endpoint path formed through their trajectory. It is important to keep in mind, however, that the gestural language is represented in an entirely different space (i.e. egocentric) than the visualization provided. Thus, two very similar trajectories in the figure may come from two very different types of gestures. The primitives presented represent roughly a quarter of those found in the highest kinematic space. Lower kinematic spaces are not depicted.

The primary parameter used in building the gestural language is the clustering threshold, ϵ (Section 3.4.3). In varying ϵ we are varying the volume of the clusters found in the data and consequently the number of clusters found. This effect can be seen in Figure 5-7. The number of clusters found diminishes rapidly as ϵ is increased. This allows for a more compact representation of the data. However, if ϵ is too large, then we over-generalize the data. If we group two dissimilar motions together in this case, then the resultant cluster is of little value.

Another experiment conducted was to analyze performance on a small, homogenous test data set of similar motions. Because the motions (circular hand motions is this case) were known to be similar, we could then asses the ability of the gestural language to represent this similarity. The first experiment was to project the cluster centroids into a three dimensional space for visualization using LLE (Figure 5-8). The clustering threshold ϵ was held constant. The figure shows that, in this projection, the clusters lie in a highly segregated configuration, suggesting that ϵ can be increased. Visual inspection shows that a small set of canonical gestures should

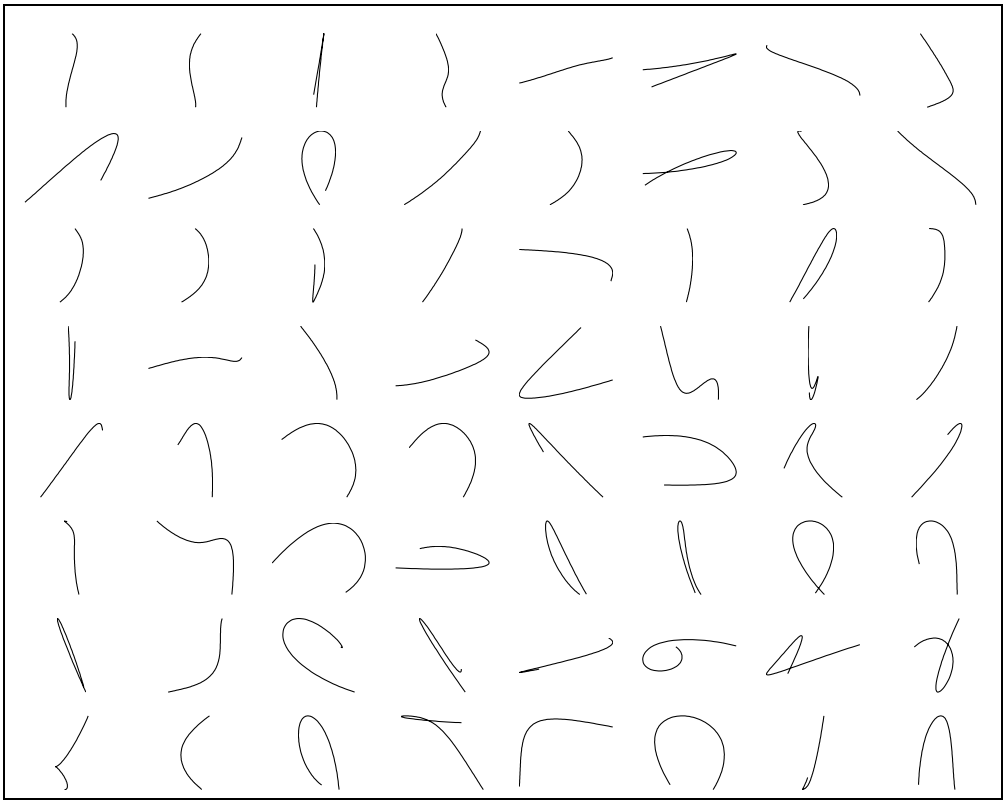


Figure 5-6: Gestural primitive hand trajectories. The gestural language was built using the complete data set. The hand trajectories displayed correspond to a subset of the gestural primitives found in the largest kinematic space.

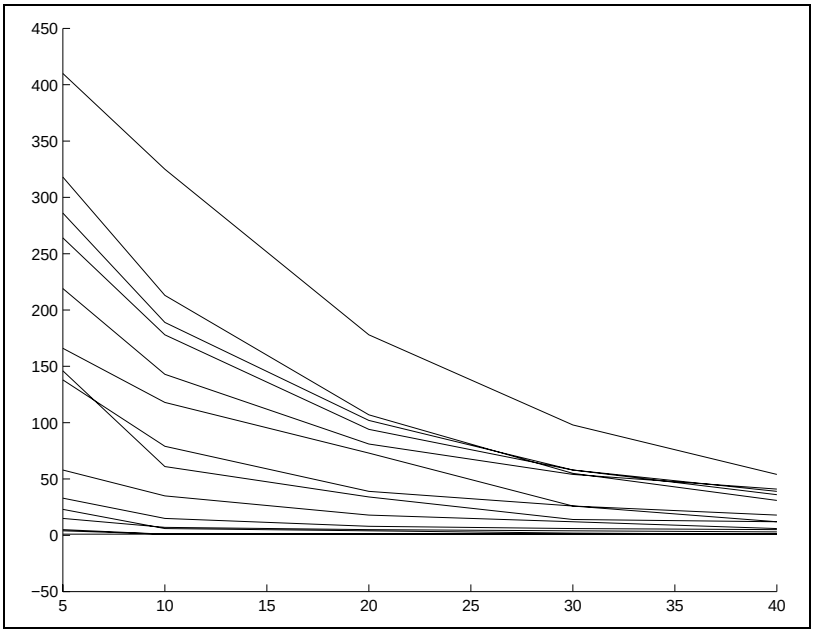


Figure 5-7: The cardinality of the set of primitives in each kinematic space versus clustering threshold. As the clustering threshold is increased, the number of primitives found is shown to decrease.

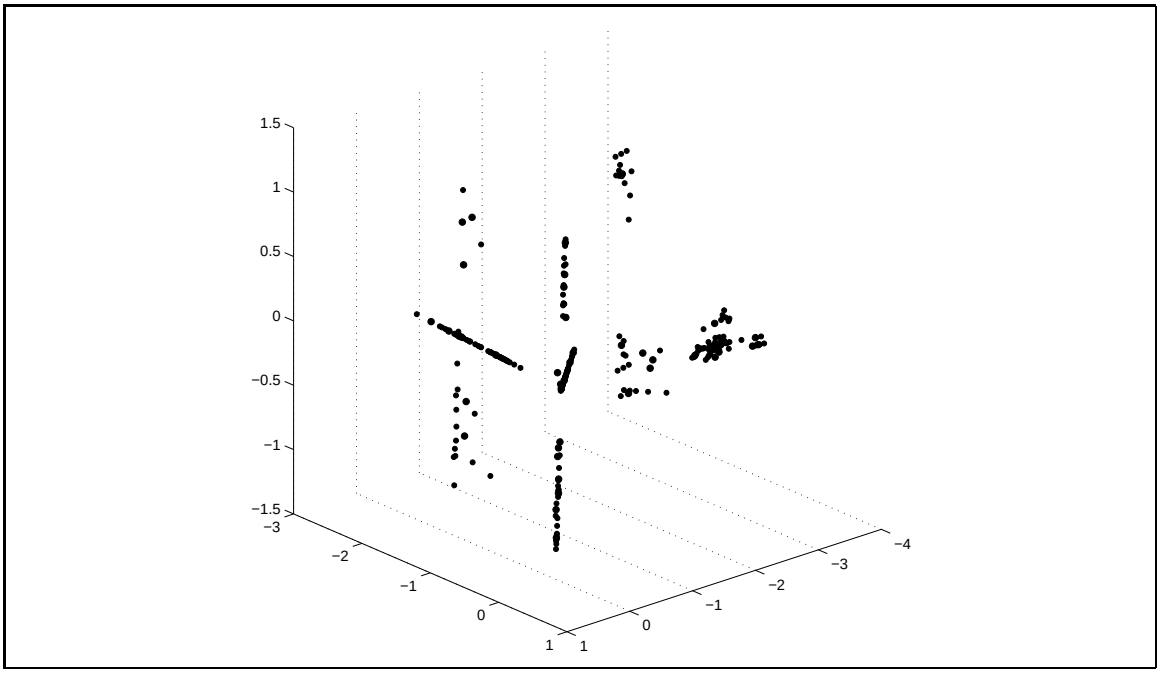


Figure 5-8: Clustering on a test data set. The primitive clusters for a set of circular hand trajectories were found and projected, via LLE, into three dimensions

exist in the data set. In a second experiment, shown in Figure 5-9, we use the same test data set. Here we decrease ϵ until the clustering converges to three canonical gestures. We provide a visualization of the convergence process in terms of the hand trajectory.

Another approach we took to assess the structure of the gestural language was to again use LLE to create a three-dimensional visualization of the primitive clusters. In Figure 5-10 we look at just the highest kinematic space corresponding to full body motions. The cluster locations are superimposed on the original data set of motions. We can see that they are fairly well-distributed across the large cluster of data. The clearly segregated cluster on the left of the figure is the set of gestures that are predominantly torso based. In Figure 5-11 we present the same type of visualization. Here, however, we are looking at the set of clusters across all kinematic spaces. We should note that a low ϵ is used in these graphs so that a large number of clusters can be visualized.

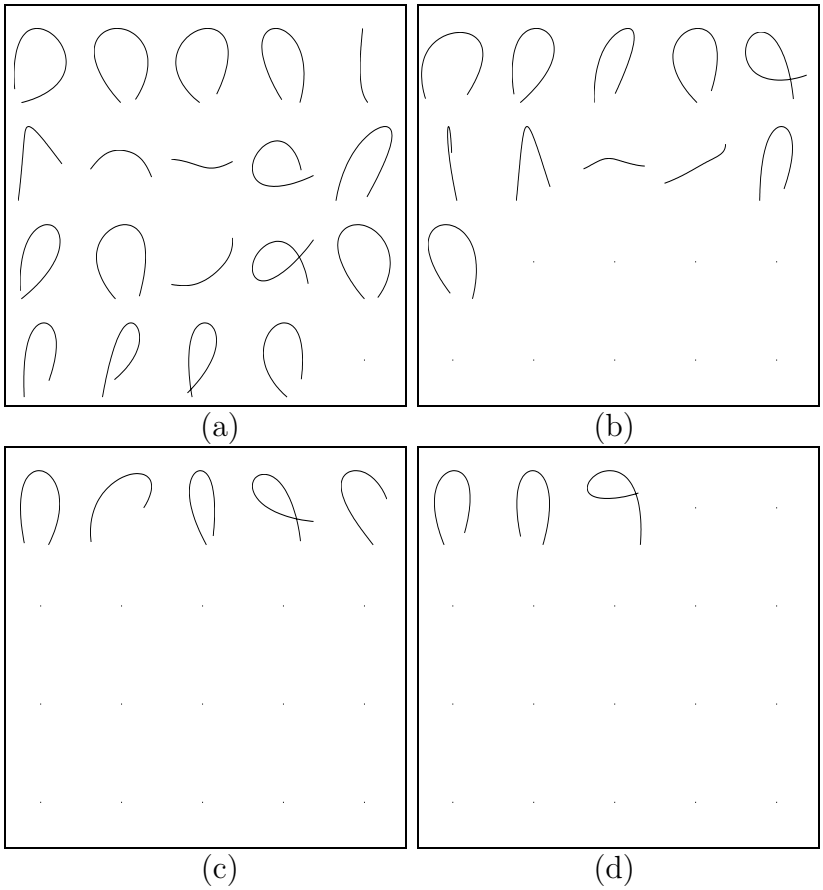


Figure 5-9: Hand trajectories of primitives. By increasing the clustering threshold, the number of gestural primitives decreases. On a test data set of circular hand motions, the clustering converges to three prototype gestures Note: Because the directionality of the trajectory is not apparent, some primitives appear identical when in fact they are not.
(a): Clustering threshold: 0.1. Roots: 19
(b): Clustering threshold: 0.2. Roots: 11
(c): Clustering threshold: 0.3. Roots: 5
(d): Clustering threshold: 0.4. Roots: 3

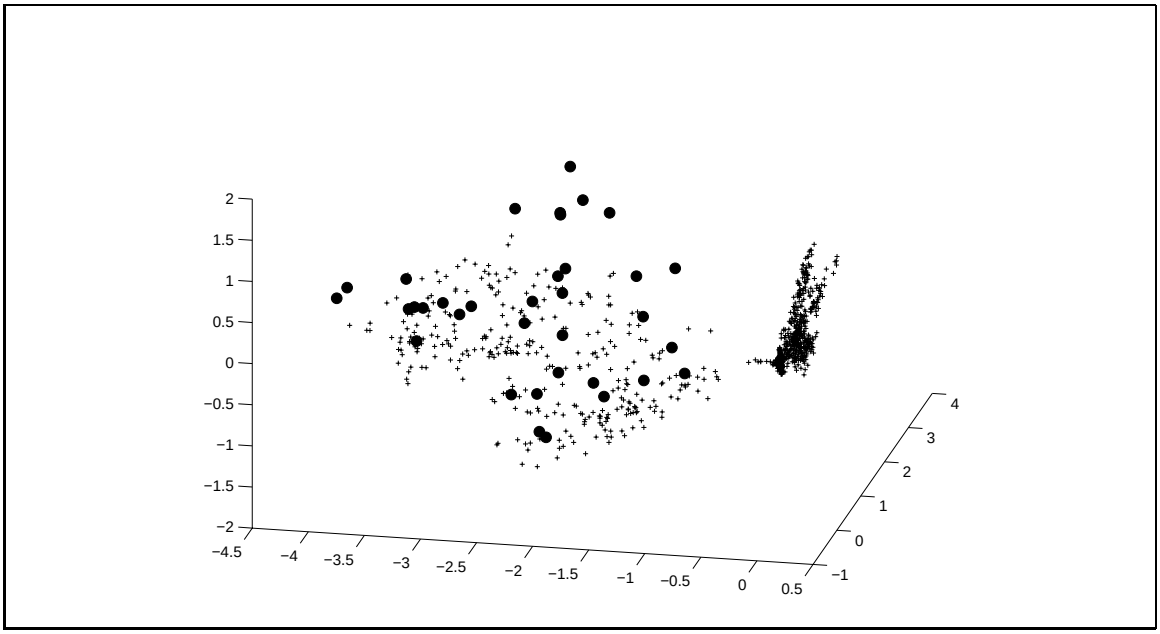


Figure 5-10: The distribution of primitive clusters for the set of full body motions (i.e., the largest kinematic space of the gestural language).

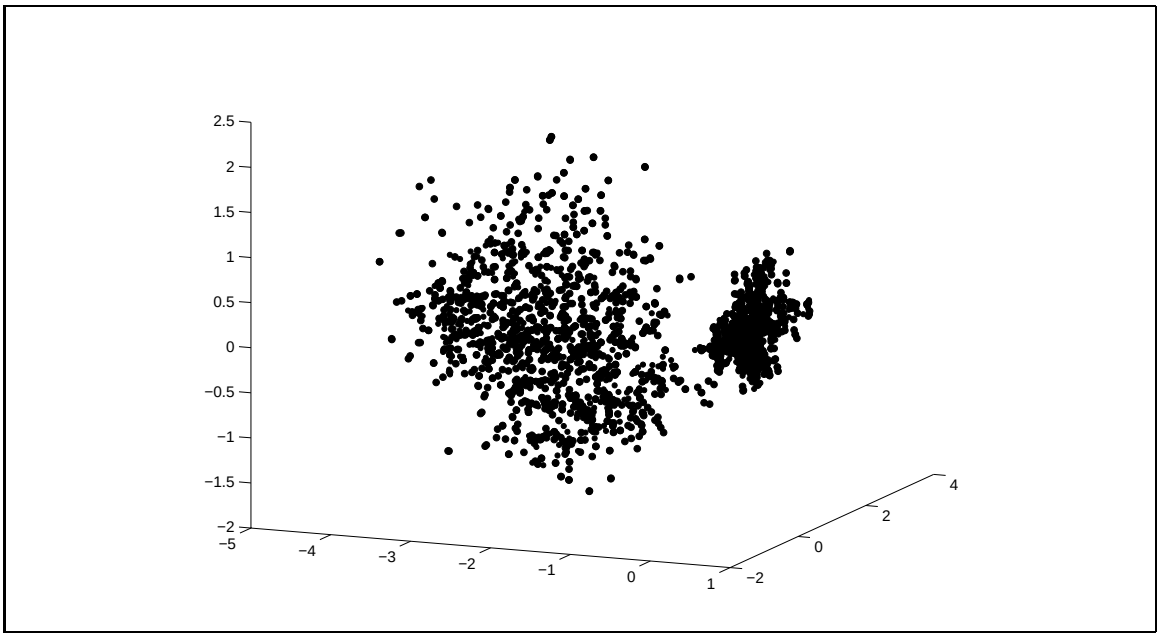


Figure 5-11: The distribution of primitive clusters for the full data set of 500 motions, across all kinematic spaces.

5.4 Task Analysis

Analysis of the gestural language performance on a given task is more direct. In the motor mimicry application, described in Chapter 4, it is simple enough to compare the desired trajectory with the generated trajectory. The scope of this thesis is limited to this application. However a full analysis of the gestural language's viability as a general organizational principle for motor control will require investigating additional applications. We have broken this analysis into two parts: performance in a simulation and performance on the physical robot. As we will see, the dynamics found in the physical application create a discrepancy between the two.

5.4.1 The Simulated Task

For the simulated task we hand generated a series of 2D trajectories. These are artificial approximations of the visually generated animate trajectories formed by the sensory unit developed by Scassellati. Using the feature search techniques (Section 3.4.7) on the animate trajectory, the gestural language formulates a motor trajectory in response. The motor trajectory is then executed on a graphical simulation of the robot for evaluation purposes.

In Figure 5-12 we can see the performance of the system on the simulated task. The figure demonstrates the adeptness of the language to replicate a variety of trajectories. While these results are promising, they do not exploit the ability to dynamically combine primitives using transition graphs (Section 5.5).

When performing the feature search, the gestural language will only activate primitives that are within a threshold of the robot's current kinematic state. We found in practice that the relative sparsity of the data required this threshold to be high. Consequently, a linear interpolation from the current kinematic state to the primitive's starting kinematic state was necessary. A second interpolation was also implemented to return the robot to a neutral posture at the end of the primitive execution.

As a second stage of the simulation task, we incorporated the real time animate trajectories from the perceptual system. These were used to drive the graphical

simulation via the gestural language. This experiment provided visual confirmation that the system behaved appropriately when using the noisier perceptual data.

5.4.2 The Physical Task

Implementation of motor mimicry task on the humanoid platform is a critical component of this work. While the simulation provides confirmation of the idealized system's ability, it is the physical implementation of the gestural language which provides the final metric of success.

As we discussed in Section 3.2, Cog's actuators introduce elasticity into the system in order to provide force feedback. A position control loop, simulating the spring and damper approximation to muscles, encloses the force control loop. For the robot to precisely follow a kinematic trajectory provided by the gestural language, we would want the joints of the robot to be very stiff. However, natural systems are not stiff in the way an industrial robot arm is, and trajectory errors naturally occur. Cog's hardware prohibits simulating an unusually stiff spring at the joint and we have not attempted to include a dynamic model in the controller. Consequently, as Figure 5-13 demonstrates, discrepancies exist between the desired trajectory and the realized trajectory. One approach to minimizing this error is to avoid high joint velocities, as they incite oscillations in the spring-damper system. From the figure we can see that the vertical range of the motion is compressed. This is due to the influence of gravity in the robot dynamics. In moving to the physical system, it soon became evident that for the robot to realize a natural quality of motion, the dynamics of the system would have to be considered in greater detail.

Finally, we tested the system as a whole. Integrating a complex system such as this into a real time platform is a challenge. Though the issues encountered do not directly relate to the work of this thesis, it should be noted that the practical aspects of the implementation certainly impact the model. For example, the latency incurred by utilizing the gestural language prevents its inclusion in a tight feedback loop with the environment. For this very reason, it is largely feed-forward. Noisy perceptual information also posed problems. If the perceptual system captures only a portion of

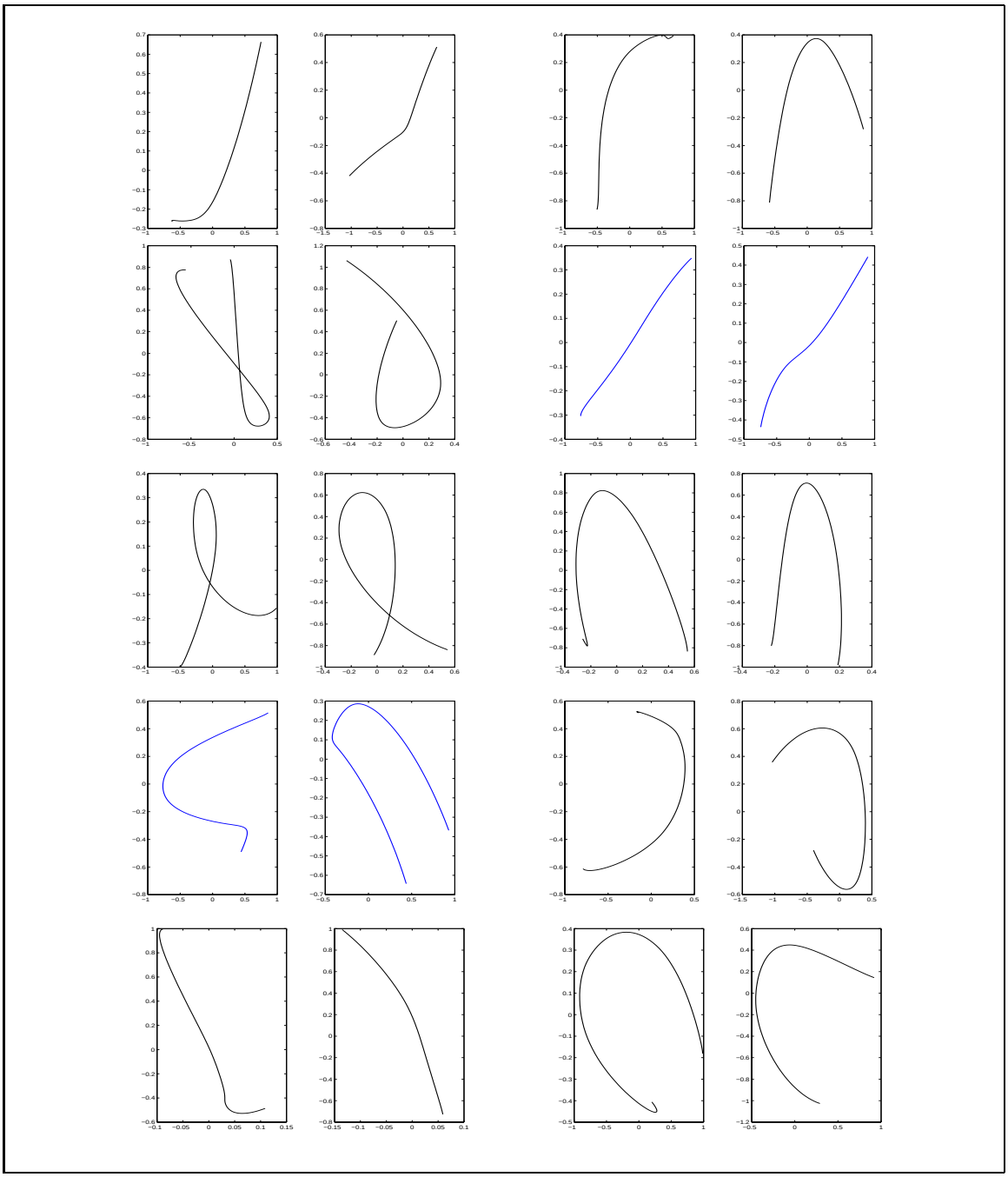


Figure 5-12: The simulated response of the gestural language to the motor mimicry task. For each pair: the left plot is the 2D perceptual trajectory to be imitated; the right plot is the 2D trajectory of the robot hand (simulated via a forward kinematic model) in response to the input trajectory.

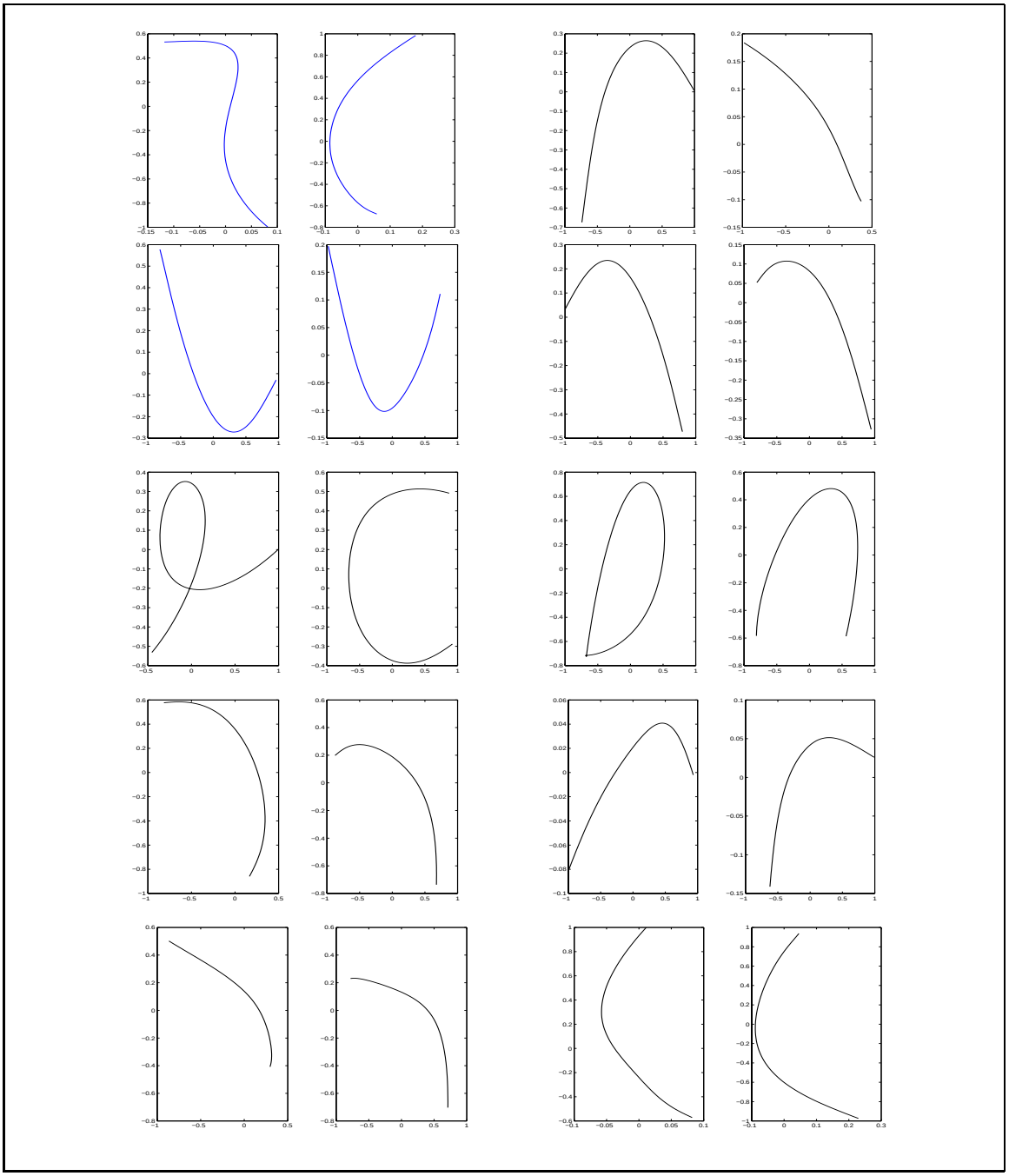


Figure 5-13: The error between the simulated primitive and its physical realization. Implementation on the humanoid introduces trajectory errors due to timing latencies and physical dynamics. For each pair shown: the left plot is the simulated 2D trajectory of the robot hand in response to a primitive; the right plot shows the actual path taken by the humanoid hand.

the facilitator's movement, then the gestural response of the robot does not appear to match well with the original motion. In Figure 5-14 we can see the original motion trajectory, the gestural response trajectory, and the trajectory as executed. Because we are using a normalized end point trajectory for feature comparisons, the mimicry can only occur to a rough approximation. Mimicry based on perception of the joint trajectories of the caregiver would certainly yield better results, though the perception of this feature is very difficult. Additionally matching the scale of the trajectories is important. While the robot may mimic a large circular arm motion with a small circular hand motion, the mismatch of scale appears erroneous. One solution under investigation is to build in assumptions about the scale of the perceived trajectories.

5.5 Discussion

Moving from the theory of motor primitives, garnered from neurophysiological data, to the application of the theory on a humanoid robot, we uncovered a number of unexpected issues and found numerous alternate paths to explore.

First, the overarching assumption is that if we encode motor actions and embed them in a high dimensional space, then a distance metric will be sufficient to discern the similarity of two points in the space. However, it can be the case that we would want two motor actions to be judged as similar even though they appear very far apart given the encoding. In addition, we may want to exploit invariance under time or invariance under joint position depending on the task. This would require separate encodings. Thus we cannot expect to find a static encoding that suffices for all situations.

We found in practice that the search heuristics play a large role in the success of the system. These heuristics guide the search towards an acceptable solution. However, in doing so, they reduce the breadth of the search and thus the system tends to find the same solution for multiple problems. In some situations, this can be seen as desirable, yet often it is the diversity of responses to a complex environment that gives the best results.

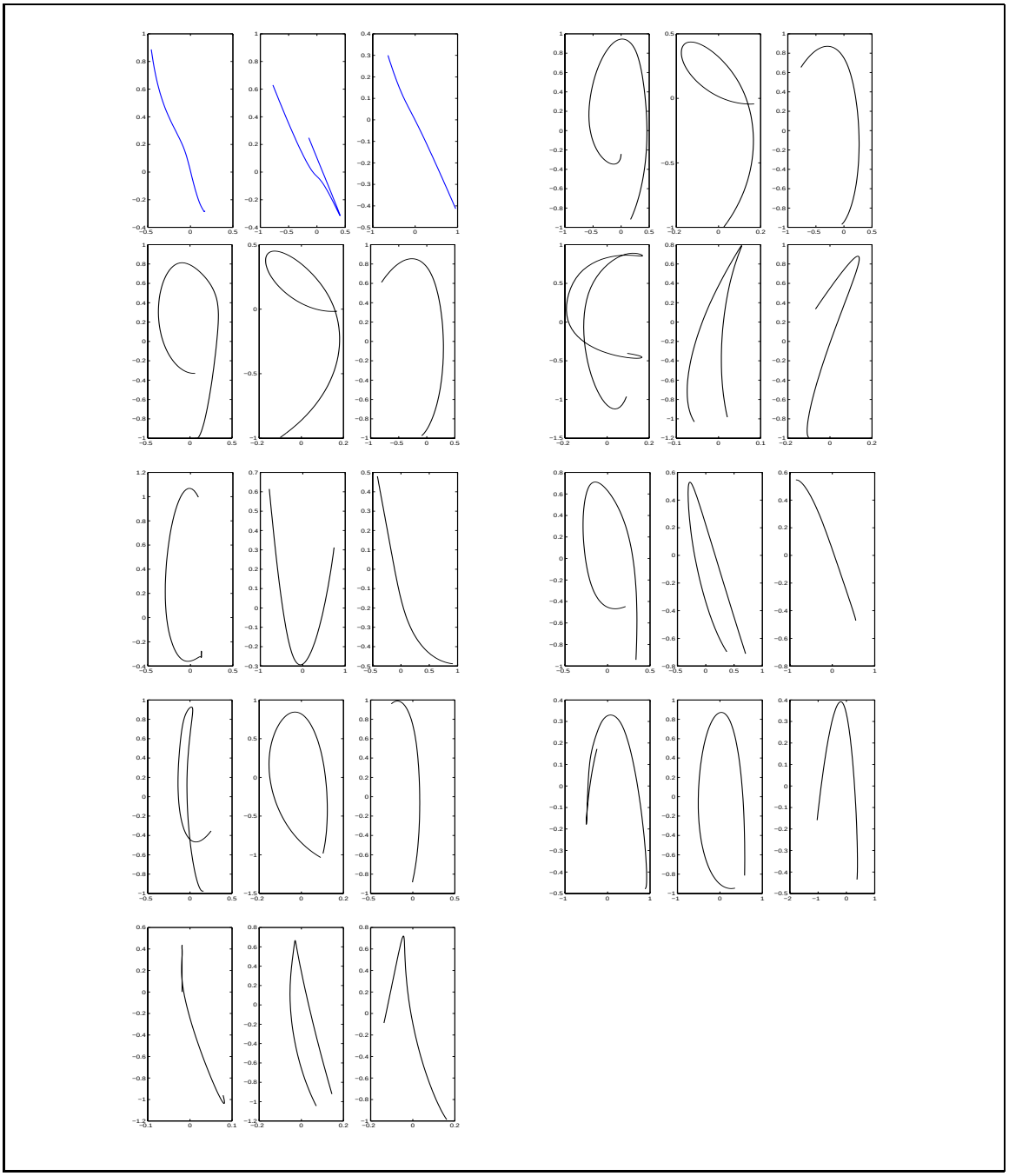


Figure 5-14: The response of the gestural language to the motor mimicry task. The response and perceptual input are displayed as the 2D projection of the end point trajectory. For each triplet: (left) the perceptual input as animate trajectory; (middle) the selected gestural response based upon search through the gestural language; (right) the actual response generated by the robot.

The clustering threshold, ϵ, plays a critical role. Its value is hand tuned. If ϵ is too small, then the number of gestural primitives is large. This results in large primitive activation sets which are computationally expensive to evaluate. Increasing ϵ reduces the size of the activation sets drastically. It also reduces the number of canonical gestures in its repertoire. A small set of canonical gestures for the gestural language is desirable if the gestures can be combined to form more complex gestures.

Much of the combinatorial aspect of the gestural language is derived from the transition graphs. A number of issues were found in using transition graphs with the language. These issues proved to be prohibitive in allowing the language to combine primitives in a useful manner. The issues are:

- While the work of (Kositsky 1998) in this area allowed velocity based transitions within a trajectory, the gestural language uses position based transitions between trajectories. This requires the consecutive execution of discrete primitives with an interpolation mechanism between them. A more flexible combinatorial method such as Kositsky's may be necessary.

- The transition graphs need a high level of connectivity to be effective. High connectivity allows for a diverse set of motions to be formed through the variety of paths through the graph. To obtain a highly connected graph, the data set needs to be large in comparison to the number of primitives in the graph. In this work the data set was prohibitively small.

The complete system performed well on the motor mimicry task given the correct circumstances. The relatively small size of the data set limited the diversity of primitives and the ability to combine them effectively. Thus, the types of motor actions that could be mimicked are fairly stereotypical. Although the perceptual system is relatively robust, the human facilitator is forced to maintain a set distance from the robot and execute deliberate motions to guarantee a fair perceptual feature.

Chapter 6

Conclusion and Future Work

6.1 Review

This thesis describes research done on the humanoid robot platform Cog. We proposed a data-driven approach to learning naturalistic gestures for humanoid motor control. The proposed model develops an organizing principle for the representation of motor actions. The representation was designed to allow for generalization of a small repertoire of canonical motor actions into a broad set of complex motions.

The work was motivated by neurophysiological findings that suggest a similar type of motor organization occurs in natural systems. We reviewed this research and the impact the research has had in humanoid motor control. Additionally, we gave a brief survey of unsupervised learning techniques that are applicable to the work done. We also reviewed related approaches to motor control.

The work in this thesis was done in the context of a human-robot imitation framework. As such, we discussed the imitation framework and how a representational system such as the gestural langauge is a critical component of imitation.

The gestural language was built from a large data set of joint trajectories taken from the robot as a facilitator guided the robot through a range of natural motions. This approach was described and compared to other techniques of motion capture.

The gestural language itself is based on clustering the data set into sets of binary trees. The data set is also decomposed hierarchically into kinematic spaces, so that

a complex motion can be described in terms of the concatenation of motor actions of lower kinematic spaces. These two structures are the key to the gestural language. They allow the decomposition into kinematic spaces to be reversed to reconstruct the complex motions. However, the reconstruction is done in terms of the gestural primitives. This provides a means for stitching the primitive binary trees together so that a generalized and novel set of gestures are created.

We also looked at dimensionality analysis and transition graphs as means to extend the gestural language. Finally we described the application of the gestural language to a real task: motor mimicry. We demonstrated how the proposed system could be used in a real world application on a physical robot. The motor-mimicry task was decomposed into a problem of a feature search through the gestural language. The feature for this task was a two dimensional animate trajectory. We evaluated the effectiveness of the system on this task, as well as the model as a whole, in Chapter 5.

6.2 Recommendations For Future Work

As is usually the case, a number of issues related to this work became apparent only after the system had been built and tested.

First and foremost is the quality of the motion data set. As we have noted, the size of the data set and the types of motor actions contained in it were problematic. The data set is a critical component in any unsupervised learning approach and exploring alternative means of motion capture should be the first course of action. Two solutions under consideration are to build a motion capture suit tailored to the robot and to investigate pre-existing motion capture data sets.

In this work we used a purely joint position based encoding. The disadvantages of this encoding become evident when a small data set is used. The data set cannot span the full range of the motor action space, requiring a heavy dependence on the ability to generalize and combine the gestural primitives. A velocity or gradient based encoding as used by (Kositsky 1998) and (Fod et al. 2000) may be a desirable avenue to explore. In essence we need to consider methods to allow the continuous adaptation

and combination of primitives.

A shift in direction that we hope to explore is the application of LLE to a new, larger data set. If we can learn the inverse mapping to the higher dimensional space, then we may be able to integrate this tool into the larger framework. This may prove advantageous if we can use LLE to find a set of low dimensional orthogonal axes which represent the space of gestures. If we can build the gestural language in this space, then we can easily parameterize gestures in terms of their global characteristics.

A final direction of further work would be to explore alternative applications of the gestural language. This would allow us to better assess its viability as an organizational principle. Pointing and social gesturing are two domains that may be well suited for exploration.

While there are many directions in which to extend and reevaluate this work, it has been instructive in broaching the larger question of: How do we build representational motor systems for robots? This work proposes a step towards answering that question, and in doing so, opens up many new paths for exploration.

Bibliography

Allot, R. (1995), Motor Theory of Language Origin, *in* J. W. et al., ed., 'Studies in Language Origins', Vol. 3, Amsterdam: John Benjamins, pp. 125–160.

Arkin, R. (1998), *Behavior Based Robotics*, The MIT Press, Cambridge, MA.

Berniker, M. (2000), A Biologically Motivated Paradigm for Heuristic Motor Control in Multiple Contexts, Master's thesis, MIT AI Lab, Cambridge, MA.

Bindiganavale, R. & Badler, N. (1998), Motion Abstraction and Mapping with Spatial Constraints, *in* 'Proceedings of the Workshop on Motion Capture Technology', Geneva, Switzerland.

Bizzi, E., Accornero, N., Chapple, W. & Hogan, N. (1984), 'Posture Control and Trajectory Formation During Arm Movement', *The Journal of Neuroscience* **4**, 2738–2745.

Bizzi, E., Mussa-Ivaldi, F. & Gistzer, S. (1991), 'Computations Underlying the Execution of Movement: A Biological Perspective', *Science* **253**, 287–291.

Bodenheimer, B. & Rose, C. (1997), The Process of Motion Capture: Dealing with the Data, *in* 'Proceedings of Eurographics Workshop on Computer Animation and Simulation', Vol. 1, pp. 3–18.

Breazeal, C. & Scassellati, B. (1998), Imitation and Joint Attention: A Developmental Structure for Building Social Skills on a Humanoid Robot, *in* 'Computation for Metaphors, Analogy and Agents', Springer-Verlag, pp. 125–160.

Brezeal, C. (2000), Sociable Machines: Expressive Social Exchange Between Humans and Robots, PhD thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA.

Brooks, R., Brezeal, C. et al. (1998), Alternative Essences of Intelligence: Lessons from Embodied AI, *in* 'Proceedings of the American Association of Artificial Intelligence Conference (AAAI-98)', AAAI Press, pp. 961–998.

Cole, J., Gallagher, S. & McNeill, D. (1998), Gestures after total deafferentation of the bodily and spatial senses, *in* S. et al., ed., 'Oralite et gestualite: Communication multi-modale, interaction', Harmattan.

Craig, J. J. (1989), *Introduction to Robotics, Mechanics and Control*, second edn, Addison-Wesley, Reading,MA.

Flannery, B., Teukolsky, S. & Vetterling, W. (1988), *Numerical Recipes in C*, W.H. Press.

Flash, T. & Hogan, N. (1985), 'The coordination of arm movements: an experimentally confirmed mathematial model', *Journal of Neuroscience* **5**(7), 1688–1703.

Flash, T., Hogan, N. & Richardson, M. (2000), Optimization Principles in Motor Control, *in* M. Arbib, ed., 'The Handbook of Brain Theory and Neural Networks', The MIT Press, Cambridge, MA, pp. 682–685.

Fod, A., Mataric, M. & Jenkins, O. C. (2000), Automated Derivation of Primitives for Movement Classification, *in* 'Proceedings of the First IEEE-RAS conference on Humanoid Robotics (Humanoids 2000)', Massachusetts Institute of Technology, Cambridge, MA.

Gallese, V. & Goldman, A. (1998), 'Mirror Neurons and the simulation theory of mind-reading', *Trends in Cognitive Sciences*.

Hastie, T. & Stuetzle, W. (1989), 'Principal Curves and Surfaces', *Journal of the American Statistical Association* **84**(406), 502–526.

Hinton, G. & Sejnowski, T. (1999), *Unsupervised Learning: Foundations of Neural Computation*, The MIT Press, Cambridge, MA.

Jenkins, O. C., Mataric, M. & Weber, S. (2000), Primitive-Based Movement Classification for Humanoid Imitation, *in* 'Proceedings of the First IEEE-RAS conference on Humanoid Robotics (Humanoids 2000)', Massachusetts Institute of Technology, Cambridge, MA.

Kositsky, M. (1998), A Cluster Memory Model for Learning Sequential Activities, PhD thesis, The Weizmann Institute of Science, Israel.

Mataric, M. J., Zordan, V. B. & Williamson, M. M. (1999), 'Making Complex Articulated Agents Dance', *Autonomous Agents and Multi-Agent Systems* **2**(1), 23–44.

Mussa-Ivaldi, F. (1997), Nonlinear Force Fields: A Distributed System of Control Primitives for Representing and Learning Movements, *in* 'Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation', Monterey, CA.

Oja, E. (1982), 'A simplified neuron model as a principle component analyzer', *Journal of Math and Biology* **2**(15), 267–273.

Ooyent, A. V. (2001), Theoretical aspects of pattern analysis, *in* M. S. L. Dijkshoorn, K.J. Towner, ed., 'New Approaches for the Generation and Analysis of Microbial Fingerprints', Elsevier, Amsterdam.

Pratt, G. & Williamson, M. (1995), Series Elastic Actuators, *in* 'Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-95)', pp. 399–406.

Riesenhuber, M. & Poggio, T. (1999), 'Hierarchical models of object recognition in cortex', *Nature Neuroscience* **2**(11), 1019–1025.

Rose, C., Bodenheimer, B. & Cohen, M. (1998), Verbs and Adverbs: Multidimensional Motion Interpolation Using Radial Basis Functions, *in* 'Proceedings of the IEEE Computer Graphics and Applications Conference', pp. 32–40.

Roweis, S. & Saul, L. (2000), 'Nonlinear Dimensionality Reduction by Locally Linear Embedding', *Science* **290**, 2323–2326.

Scassellati, B. (2001), Discriminating Animate from Inanimate Visual Stimuli, *in* 'Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence'. To appear.

Shankar, S. & King, B. (2000), 'The Emergence of a New Paradigm in Ape Language Research'. UB Center for Cognitive Science Fall 2000 Colloquium Series.

Stein, R. B. (1982), 'What muscle variables does the nervous system control in limb movements?', *The Behavioral and Brain Sciences* **5**, 535–577.

Thoroughman, K. & Shadmehr, R. (2000), 'Learning of action through adaptive combination of motor primitives', *Nature* **407**, 742–747.

Ude, A., Atkeson, C. & Riley, M. (2000*a*), Planning of Joint Trajectories for Humanoid Robots Using B-Spline Wavelets, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation', San Francisco, CA, pp. 2223–2228.

Ude, A., Man, C., Riley, M. & Atkeson, C. (2000*b*), Automatic Generation of Kinematic Models for the Conversion of Human Motion Capture Data into Humanoid Robot Motion, *in* 'Proceedings of the First IEEE-RAS conference on Humanoid Robotics (Humanoids 2000)', Massachusetts Institute of Technology, Cambridge, MA.

Williamson, M. (1995), Series Elastic Actuators, Master's thesis, MIT AI Lab, Cambridge, MA.

Williamson, M. (1996), Postural Primitives: Interactive Behavior for a Humanoid Robot Arm, *in* Maes & Mataric, eds, 'Fourth International Conference on Simulation of Adaptive Behavior', The MIT Press, pp. 124–131.