Behaviour-based physical agents as information display devices

Xing, Shusong

2005

Xing, S. (2005). Behaviour-based physical agents as information display devices. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/5421>

<https://doi.org/10.32657/10356/5421>

Nanyang Technological University

Downloaded on 26 Apr 2023 17:36:23 SGT

Behavior-based Physical Agents as Information Display Devices

Xing Shusong

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree
of Doctor of Philosophy

2005

Abstract

This thesis studies the structural and behavioral design of the physical agents that can convey information to human through life-like body movements. These physical agents are natural and intuitive human-machine interfaces in social environment. The issues pertinent to behavior generation and expression on them are investigated. Based on the requirements and system components essential to the design and construction of the physical agents whose body structures are highly diversified due to their intended communication tasks, we have built three robotic puppets as human's avatars that make their bodies the instruments of expression. They take the structures of underconstrained wire-suspension mechanisms to solve the degrees of freedom problem. In order to make the motion generation process effective and flexible on the physical agents with different body structures, an agent-centered design approach that behavior generation and expression are founded on the motor capability is applied to the robotic puppets. The basic behavior modules are acquired from the mechanical structures of the robotic puppets and used as the innate behaviors that are helpful to generate illusion of life. Inspired biologically, the temporal and spatial aspects in movement synthesis are performed separately. As an answer to the correspondence problem, the multilevel behavior representation gives flexibility to motion transform while keeping the movements human-like. More complex behaviors independent from the underlying mechanical structure are synthesized from the motor primitives sequentially and concurrently. Coordination of full body movements is achieved though dynamically constraining the body parts in spatial symmetry. The demonstration shows that the robotic puppets have a diversity of body movements and enormous encoding potential in behavioral communication. This work not only achieves expressive movements on an unprecedented mechanism combining robotics with a type of traditional art form, but also provides an interesting testing bed for studying motor control and non-verbal communication.

Acknowledgements

I would like to thank Dr. Chen I-Ming, my supervisor, for his patient guidance and constant support during this research. I am especially grateful for the substantial time and effort he has given me through the years. It is under the influence of his breadth of knowledge, ability to seize new ideas, and attitude to research, I begin to use my mind but not just move like a puppet.

I am also thankful to Dr. Francis M.J Nickols, Dr. Yeo Song Huat, Dr. Chen Weihai, Dr. Yang Guiling, Dr. Yu Qing, Dr. Bai Shaoping for many suggestions. I learn a lot from the discussions with Dr. Yang.

I would like to express my sincere gratitude to Mr. You King Sun, Mr. Dash Anjan Kumar, Mr. Soo Han Chern, Ms. Theingi, Mr. Pham Hui Hoang, Mr. Pham Cong Bang, Mr. Jin Yan, Mr. Lim Chee Kian, Mr. Yan Liang. I especially thank Mr. Raymond Tay Boo Keng whose efforts on mechanical structure of the puppet ROMS-II make it the major platform used in my research and Mr. Stefan Künzler for his work of designing ROMS-III. All of them transform the lab into a place full of atmosphere of freedom and warmness. It is a pleasure to work with them.

Most importantly, I received infinite support on both life and spirit from my family.

I grateful acknowledge the support to my research from Robotics Research Center, Nanyang Technological University and Singapore government.

Table of Contents

Abstract	i
Acknowledgements	ii
Table of Contents	iii
List of Tables	vi
List of Figures	vii
1 Introduction	1
1.1 Motivation	2
1.2 Design Issues	4
1.2.1 Diversity of Body Structures	4
1.2.2 Degrees-of-Freedom Problem	5
1.2.3 The Correspondence Problem	6
1.3 Proposed Solutions	6
1.3.1 Niche Space vs State Space	6
1.3.2 Illusion of Life	7
1.3.3 The Robotic Puppet Systems	8
1.3.4 An Agent-Centered Method	10
1.3.5 Multilevel Behavior Representation	11
1.4 Literature Review	12
1.4.1 Interfaces of Natural Communications	12
1.4.2 Wire-Suspension Mechanisms	15
1.4.3 Motion Generation Methods	16
1.5 Summary	18
2 Design Considerations and Appearance Selection	21
2.1 Design Considerations	21
2.1.1 Niche Space	23
2.1.2 Design Space	24
2.2 Appearances	25
2.2.1 Degree of Embodiment	25
2.2.2 Anthropomorphism	27
2.2.3 Uncanny Valley Phenomenon	28

	iv
2.2.4 Balances between Appearances and Behaviors	29
2.2.5 Focus on Movement	32
2.2.6 Characteristics of Human Movements	32
2.2.7 Spatial Relationships	33
2.2.8 Two Kinds of Avatars	34
2.3 Summary	36
3 The Mechanics of The Robotic Puppet	37
3.1 Structure Selection	37
3.2 Puppeteer's Method of Structural Design	42
3.3 Mechanical Structure of the Robotic Puppet	43
3.4 Kinematics	45
3.4.1 Degrees of Freedom	45
3.4.2 Reference Frames	47
3.4.3 General Formulation of Kinematics	49
3.5 Simplification of Kinematics Calculation	54
3.6 The Configuration of One Arm	57
3.7 Workspace and Tension	62
3.8 Summary	65
4 Behavior Modules	67
4.1 Introduction	67
4.1.1 Requirements of Behavioral Design	68
4.1.2 Behavior Generation Methods	69
4.1.3 Complexity in Behavior Generation	69
4.1.4 Behavior Generation in Animals	70
4.1.5 Inferences from Ethology	71
4.1.6 Inferences from Force Field Experiments	72
4.1.7 Benefits of Using Primitives	73
4.2 Primitives Created by Human	76
4.3 Motor Capability Discovery	77
4.4 Movement Synthesis	78
4.4.1 Multilevel Movement Representation	80
4.4.2 Types of Primitives	81
4.4.3 Posture Primitives	82
4.4.4 Discrete Movements	83
4.4.5 Rhythmical Movements	83
4.4.6 Coordination	86
4.4.7 Full Body Movement	88
4.5 Summary	91
5 Expressive Behaviors	93
5.1 The Correspondence Problem	93
5.2 Mimicry	95
5.3 Motion Transform through Imitative Learning	97

5.3.1	Motion Transform on the Robotic Puppet	99
5.4	Body Movements in Communication	102
5.4.1	Emblems	103
5.4.2	Propositional Gestures	104
5.4.3	Spontaneous Gestures	104
5.4.4	Clothes	105
5.5	Summary	105
6	System Implementation	106
6.1	System Overview	107
6.2	Puppeteer Mechanism	107
6.2.1	Basic Design Concepts and Considerations	108
6.2.2	Use of Pulleys and Motors	108
6.2.3	Structure of Motor Platform	110
6.3	Hardware	111
6.4	Puppeteer Software	113
6.4.1	Motor Control Module	114
6.4.2	Internal Model of Body Structure	117
6.4.3	User Interface	120
6.4.4	Motion Synthesis Module	122
6.5	Puppet Show	124
6.6	Summary	125
7	Conclusion	126
7.1	Summary of Contributions	128
7.1.1	Complex Wire-Suspension Mechanisms	128
7.1.2	Agent-Centered Behavioral Design	129
7.1.3	Multilevel Behavior Representation	130
7.2	Assumptions and Limitations	131
7.3	Future Directions	132
7.3.1	Design Methodology	133
7.3.2	Behavior Generation through Self-Organization	134
	Bibliography	135
	A Figures of Puppet Dimensions	147
	B Data Structures	150
	C Function Definitions	153

List of Tables

1.1	Specifications of the robotic puppet systems	9
5.1	Types of imitative learning	94

List of Figures

1.1	Information display through expressive behaviors	3
1.2	The humanoid robots have highly diversified body structures	5
1.3	The robotic puppet systems: ROMS-I, ROMS-II, and ROMS-III	8
1.4	A general architecture of embodied conversational agents	13
2.1	Physical agents compared in one design space	22
2.2	The Uncanny Valley phenomenon	29
2.3	Different appearances cause different expectations to the behaviors	30
2.4	The robotic puppet in one-way communication to humans	34
2.5	The social agents interact with humans in two-way communication	35
3.1	Two fully constrained wire-suspension mechanisms	38
3.2	A three-wire underconstrained robot	39
3.3	The structure of a marionette	41
3.4	The stringing method of the robotic puppet ROMS-I	44
3.5	The reference frames of a robotic puppet	47
3.6	General structure of the body parts	50
3.7	Small displacement on the torso caused by lateral forces	54
3.8	One arm pulled by a string moving in a vertical plane	56
3.9	Freebody diagram of the individual links	58
3.10	Movable suspending point pulls the links in 3D	60
3.11	One suspended arm viewed from the top	60
3.12	The configurations of one arm for different string lengths	62
3.13	The workspace of one arm	63
3.14	The tension magnitude on the string in different lengths	63
4.1	The requirements and effects of the behavior generation methods	68
4.2	Three special postures of the left arm	77

	viii
4.3 Multilevel motion synthesis	80
4.4 The arm moves periodically	84
4.5 Rhythmic movement of one arm	84
4.6 The periodical movement of one arm	86
4.7 The limbs move towards the center of the body	89
4.8 One arm proceeds the other	90
4.9 The independent movements of the upper and lower body parts	90
4.10 The arm and the leg on the same side move together	90
4.11 The arm and the leg on the opposite side move together	90
5.1 Imitative learning through composing primitives	96
5.2 Puppet ROMS-II follows the simulated human model	100
5.3 Walking movement transformed from the simulated character	101
5.4 Expressive gestures	102
6.1 The system overview of the robotic puppet ROMS-II	107
6.2 The motor platform	109
6.3 Sideview of the motor platform	110
6.4 I2C bus network	111
6.5 Functional structure of movement generation and expression	113
6.6 Motor control module executes motor commands	115
6.7 Configuration for serial communication	115
6.8 I2C bus communication among the microcontrollers	116
6.9 Five basic postures	119
6.10 The user interface of the behavior editing program	121
6.11 The teaching panel	121
6.12 The robotic puppet ROMS-II can take a lot of postures	123
6.13 A short puppet show	124
A.1 The size of ROMS-I's puppet body	147
A.2 The size of ROMS-II's puppet body	148
A.3 The size of ROMS-III's puppet body	149

Chapter 1

Introduction

The trend of physical agents entering people's daily life in the forms of service robots, pedagogical robots, and entertainment robots makes social interaction important. In these situations where robots become the partners of human beings, communication with human is the focus, either as the required skills for achieving given tasks or as the primary functions of the robots whose task is to interact with people. In order to allow people to interact with the artifacts the way they would with other people, the robots are augmented with many humane features, such as speech, gestures, and emotion. Body movements and gestures are indispensable components in interpersonal communication, and fortunately the physical bodies of the physical agents provide natural resources for this form of communication.

The objective of this research is to study the structural and behavioral design of the physical agents that have rich behaviors and propose design methodology for exploiting the expressiveness of their body movements in realizing human-machine interaction. The methods and components essential to design and construction of the physical agents are studied and verified through the making of three robotic puppets without the aid of other communication channels or modalities. The issues of interest include those pertinent to movement generation and expression on the physical agents, whose body structures are highly diversified.

1.1 Motivation

Our human beings have a trend to treat computers and other forms of media in the same way of communicating with other people [1]. In order to make interaction with robots and other agents like graphic avatars easy and comfortable [2], people add multimodal communication capability, such as speech, intonation, facial expressions, and gestures to them. Thus, the communications between humans and machines can be performed in a natural and intuitive manner, and humans are no longer constrained to use windows, icons, monitors, and pointers as the interfaces. In the situations where human-machine interaction is the center of the tasks, the social capability may have more importance for the physical agents than their computational capability. By applying natural communication modalities to interaction with humans, the robots embed into both the physical space and the social space where the humans live. They bring computation into the physical environment and allow people to treat them as human partners to certain extent. The consequences of using robots, from teleoperated robots to autonomous robots, as physical and social interfaces, are not only their ubiquitous computation, but also social engagement. It is the use of socially acceptable communication in the robots that helps breaking down the barrier between the digital world and the human society, and probably makes the robots become the universal interfaces [3]. Hence, to interact with the robots or other types of agents enhanced with human-like characteristics is easy and appealing to the users from different backgrounds.

Within the most frequently used channels of verbal and nonverbal behaviors in human communication, body movements and gestures in robot design have not received enough attention proportional to their important roles. The behavior refers to any action or reaction performed by an agent. As the essential parts of non-verbal behaviors, body movements and gestures are not only the means by which people influence the environment and other people, but also a source of expression of their internal states. Body movements have powerful impact to interpersonal communication because they give cues of omnipresence, form a kind of universal language, provide trusted information, enhance understanding,

and sometime cause misunderstanding [4]. People rely heavily on nonverbal cues to express their meanings and to interpret the communicative activities of others. They have remarkable capability to distinguish body movements and are masters of exploiting the large encoding potential of all possible body movements. This type of medium has salient performance that other modalities lack in natural communication. They can be used either independently in expression, or synchronized with other modalities to complement and enhance the expression. Due to their indispensable role, body movements should be applied to human-machine interface design. So applying body movement to human-robot interaction is of consequence to the applications of the physical agents. This drives us to incorporating body movement for social interaction in physical agent design and construction.

An agent is a computational system that can automatically determine its actions according to the environment in order to satisfy its goal (Figure 1.1). It may be an animated character or a robot that acts on its own intention or motivation. A physical agent or a physically embodied agent refers to the agent that has physical body in the real world, such as a robot or a robotic puppet. The emphasis on the physical existence of the embodied agents differs them from those agents present in the form of computer programs. Both the appearance and the body structure of a physical agent may be involved in information display process. Information is encoded with the behaviors of a physical agent into expressive behaviors, which can be decoded by human observers. The information may come from the physical agents and the environment, especially humans and networks.

Figure 1.1: Information display through expressive behaviors

The scope of this study includes identifying the essential problems in design, proposing design methods for appearance, structure and behavior of the physical agents, and implementing system demonstration.

We believe how to create a physical agent that express information with body movements is a problem of design, that is, the selection of body structure, motor control architecture, behavior representation and generation methods according to the intended tasks. There are a list of decisions need to be made in designing a physical agent for expressive tasks. For example, what should the physical agent look like? Is it human-like or not? How to produce movements on it? What kind of messages can be sent with its body movement? And so on.

1.2 Design Issues

Three design issues are identified as essential to the structural and behavioral design of expressive physical agents.

1.2.1 Diversity of Body Structures

The physical agents designed for interaction applications have no standard structures. People have developed many robots that can be used in socially interactive tasks [5] [6]. Some humanoid robots have complicated body structures while other robots only have several degrees of freedom (DOFs). Though they are all designed to have human features in their appearances, one obvious characteristic shared by them is the diversified body structures. The size, appearances, number of DOFs, and materials of their bodies differ remarkably. A collection of humanoid robots are shown in Figure 1.2 to illustrate the structural diversity of physical agents. From left to right, the top row is Cero [7], the middle row are Wendy developed in Waseda University [8], ISAC [9] developed in Vanderbilt University, Felix [10]; the bottom row are Sarcos [11], Honda's ASIMO [12], Sony's SDR-4X [13], and PINO [14].

The diversity of body structures complicates behavioral design and motion control of the physical agents. The expressive behaviors designed for the specific agents may lack of

Figure 1.2: The humanoid robots have highly diversified body structures

portability. Thus, it is desirable that the behaviors designed for one physical agent can be transferred to another agent without too much modification. A method or standard should be proposed to guide the whole process from appearance selection to behavior generation in order to make structures and behaviors of the physical agents fit for the expression tasks.

1.2.2 Degrees-of-Freedom Problem

The degrees-of-freedom problem [15] [16] [17] is about how to compress the large number of DOFs in detailed representation to the smaller number of DOFs at the high-level representation. The physical agents may have many DOFs that need to be controlled in order to generate coordinated movements. The movements covering multiple body parts can be analyzed at different levels of abstraction. A comprehensive description of their motion

states in low-level details would require enormous parameters while a much smaller set of parameters would be required if the motion is represented in abstraction.

A large number of DOFs makes it possible for the designers to setup a rich behavioral repertoire on the physical agents for expression tasks. However, the enormous possibilities in behavioral design cause an exponential explosion in the number of distinctive actions that can be taken in every state.

1.2.3 The Correspondence Problem

The behaviors of one physical agent may be acquired through some types of motion transfer, such as imitative learning, from other physical agents, humans, or animated figures. There may be more or less differences on their appearances, kinematics, dynamics, and motion control methods. A measurement to evaluate behavior similarity among the source models and the target models is required to ensure the transformed behaviors on the targets would satisfy the expectation of the users. It may also be used to predict if a body movement performed by a physical agent could be regarded by a human being to be the same as human movement. Hence, there is the correspondence problem [18] in motion transfer. This problem becomes particularly obvious for expressive physical agents on which humane features are incorporated into their diversified body structures and behaviors.

1.3 Proposed Solutions

Efforts are made from several aspects to resolve the problems encountered in behavioral and structural design of the physical agents for human-machine communication.

1.3.1 Niche Space vs State Space

There are enormous possibilities in selecting the structures and behaviors for the intended physical agents. Some communication tasks may require complex behaviors while others not. Commonly, complex behaviors may not be expressed on the physical agents having

simple structures of several DOFs. To keep the balance between expressive behaviors and communication tasks, and the balance between structures and behaviors, the concepts of the design space and the niche space [19] is recruited as the basic principle in physical agent design. The notion of the niche space was originated in describing how an animal adapts to the habitat. It is used here as a criterion to evaluate the design of the physical agents. A good design is defined as a perfect match between the niche space and the design space. The set of all possible requirements to the performance of a physical agent constitutes the niche space. The design space is the range of all possible combinations in terms of various design parameters.

1.3.2 Illusion of Life

Puppeteers and film makers have recognized that the expression effects of artifacts could be reached by producing illusion of life on the audiences. *"An old dustcloth, a feather boa, or a pair of bare hands operated by a skillful puppeteer can be just as alive and real to an audience as any human actor."* [20]. Each physical agent has a collection of movements that it has the capability to perform. Though these body movements exclusively belong to the physical agent, they can be associated with certain human movements by the audiences. This concept suggests that it might not be necessary for the physical agents to strictly emulate human in structure design, and incorporating humane features in behaviors might facilitate communication. Dissimilarity on the structures between physical agents and human can not prevent them from performing behavior expression.

The body movements of the physical agents that are rational in the observer's eyes have better communication effects than fine imitation to human behaviors. This effects of illusion of life can be implemented through endowing the physical agents with emotion, personality and motivation. To make the physical agents become engaging human-machine interface, agent design should cover all three levels of communication: appearance, behavior, and emotion. Movement has more weight than appearance in generating believable performance. Even if the trajectories of the body movements of the physical agents are not

Figure 1.3: The robotic puppet systems: ROMS-I, ROMS-II, and ROMS-III

seriously close to those of human movements, information can still be delivered.

1.3.3 The Robotic Puppet Systems

The puppets are a type of primitive avatars that have been using body movements in expression for a long time. An avatar is an agent that represents a real person. There are several types of puppets, such as rod puppets, hand puppets, and marionettes. Marionette is one kind of puppets that are operated through many strings. A marionette is able to show much more postures and body movements than a rod puppet or a hand puppet through its complicated body structure. A skillful puppeteer can manipulate a marionette to produce complex human-like behaviors. Vast amount visual information enough to describe a complete story can be delivered to the audiences in the form of nonverbal communication.

To investigate the issues about behavior generation and expression, three robotic puppet systems (Figure 1.3) that realize motion control of the marionettes through mechatronic means have been built as the platforms of studying body movements. They are the physical agents that have enough mechanical complexity and behavior diversity for expression tasks. The robotic puppet ROMS-I (RObotics Marionette System) shown in Figure 1.3a was

Table 1.1: Specifications of the robotic puppet systems

Specifications	ROMS-I	ROMS-II	ROMS-III
DOF of puppet	30	35	23
String number	10	14	16
Controllable DOF	8	16	16
DOF of system	22	19	7
Motor number	8	16	16
Link number	9	11	10
Joint number	8	10	9
Joint type	8 spherical	9 spherical 1 universal	4 spherical 5 hinge
Height (m)	0.312	0.305	0.295
Weight (kg)	0.314	0.214	0.259

the first prototype mainly used to explore the process and methods of structural design and modelling [21]. The puppet body of ROMS-I is consisted of nine articulated rigid bodies. The motors of the puppet system are fixed on the frame. Two strings having fixed lengths are connected to the shoulders of the puppet body to support its weight. Based on the experience of designing ROMS-I, we further developed an improved version of the puppet system, ROMS-II (Figure 1.3b). It has a life-like figure body covered with clothes for showing the expressive behaviors. The motors on the platform are carefully arranged so that the puppet body of ROMS-II can be controlled in the manner closer to the way of a puppeteer manipulating a marionette. Six motors of its 16 motors in total are mounted on two small platforms that can rotate on the horizontal plane. It employs more strings and motors than ROMS-I (Table 1.1), and can show more behavior patterns. The third system, ROMS-III (Figure 1.3c), was developed simultaneously with ROMS-II to demonstrate the realization of the puppeteer mechanisms with various structures. The puppet figure of ROMS-III is similar to that of ROMS-II (Appendix A). The shoulder strings of both ROMS-II and ROMS-III may have variable lengths. Hence, the methods of structural design of expressive physical agents will be illustrated using the puppet ROMS-I and behavior design using ROMS-II. Puppetry is a performing art includes the entire process of designing, making, and choreography of the puppets. This is similar to the

process of design, construction, and control of a physical agent. Through building the robotic puppets, we can compare our structural design methods of the physical agents with those of the puppeteers who have accumulated abundant experiences.

Another advantage of using the robotic puppets to study information display through body movement is that the puppets need not to explicitly control the balance which is demanded by the humanoid robots in order to keep them upright when standing and walking. Expressive body movements shown on humanoid robots have to be superimposed upon the fundamental behaviors of carefully-controlled gaits and postures that prevent the robots from falling down, while the puppets suspended by strings can stand in balance when it has the lowest potential energy. In addition, to give the users enough control over the expressive behaviors, the robotic puppets are not made to be autonomous and manoeuvrable in the environment. They have limited interactions with the environment, so they do not require complex perception and cognition capabilities that are necessary for autonomous robots. We can concentrate on designing the physical agents that convey information through body movement. If the robotic puppets are made to be autonomous or movable, they will have strong interactions with the environment. Their behaviors will not only be controlled by the users, but also take reactions to the surrounding situations. The displayed behaviors will not be the sole reflection of the users' intention. All these factors make the robotic puppets well-suited for our aim of studying motor control and behavioral expressiveness.

We address the behavior generation problem for the expressive agents with two methods: *agent-centered behavioral design* and *multilevel behavior representation*.

1.3.4 An Agent-Centered Method

The user-centered method is prominent in designing the robots and other types of agents engaging in human-machine interaction. The agents, either physically embodied or not, are built to have human-like conversations, human-like postures, facial expressions, human's

emotional and intentional models. Imposing human communication skills to agents facilitates human-machine interaction through grounding the shared meanings that are easily understandable to human.

Contrast to the user-centered method, we take an agent-centered perspective in design and construction of the physical agents. The behavioral design and acquisition are achieved based on the mechanical structures and motor capabilities of the physical agents. This sets up tightly associations between the body structures and behaviors of the physical agents. The motor capability uniquely determining the expressive behaviors the agents may have can be described with the workspace, extreme configurations and manipulability of their mechanical structures. The process of motor capability discovery and the agent-centered method do not conflict with the notion that the effects of social interaction fundamentally depend on the observers. Rather, they explore the mechanisms underlying anthropomorphism from another route. This design method is helpful in solving the problems of behavior generation, description, and expression on the physical agents with a wide range of body structures and morphologies.

1.3.5 Multilevel Behavior Representation

We represent and design the behaviors of the physical agents in a distributed multilevel structure which greatly reduces the complexity of movement control through combining a collection of modularized behaviors. The high level behaviors are designed separately from the low level behaviors. The behavior modules distributed at the lower levels provide services to those at the higher levels. This separation makes part of the behavior repertoire reusable on the physical agents with different body structures. The separation of the function of each behavior module from its internal implementation method provides the designer with the flexibility in selecting appropriate methods according to the mechanical structure and intended tasks. The separation of temporal and spatial aspects in each behavior module makes timing and rhythm control explicit. This multilevel behavior representation is effective in achieving motor control and coordination of the physical agents

with human-like body structures that have complex body structures and a large number of DOFs.

1.4 Literature Review

This research of the physical agents integrates the knowledge from human-computer interaction, robotics, and nonverbal communication. Though a significant body of work can be found in the supporting areas, there is little literature that can be considered directly related to it.

1.4.1 Interfaces of Natural Communications

The pursuit of the universal human-machine interface has provided more than enough impetus for people to develop various types of agents supporting natural communication. The progresses in speech recognition and synthesis, computer vision, and agent theories prompt the appearance of several types of interfaces comfortable to the human. They are *embodied conversational agents*, *intelligent environments*, and *social robots*.

Embodied Conversational Agents

Embodied conversational agents in the form of animated characters can converse with humans face-to-face through natural communication channels. In their implementations, gestures, facial expressions, body movements, intonation, and gaze are integrated with the spoken languages. Most embodied conversational agents that are developed under the general framework constituted by the perception, cognition and action modules, maintain an integration of computational models of conversation, multimodal interaction, emotion, and personality (Figure 1.4). There are plentiful studies on embodied conversational agents [22] [23] [24] for various applications. Cassell et al. [25] created a 3-D graphic character that is capable of face-to-face talking with the human in real-time using both verbal and nonverbal modalities. Emotion and personality models of humans are also incorporated into the architectures of many conversational agents as important components [26].

Figure 1.4: A general architecture of embodied conversational agents

Ball [27] applied a two-dimensional emotion model to an animated personal assistant in order to make the interaction engaging. The multimodal perception of the conversational agents employ an integration of speech recognition [28], lip reading, facial expression detecting [29], gaze detecting, spatial position tracking, gesture recognition, and intonation recognition.

Intelligent Environments

Intelligent environment is another kind of human-machine interface that employs natural communication channels to create interactive spaces. There are rather spacious places in intelligent environments to install various sensors for detecting verbal and nonverbal behaviors of the users and display devices to convey system information to the users. Coen [30] built an Intelligent Room equipped with computer vision, speech and gesture recognition systems to capture what the inhabitants are doing and saying. A similar implementation is the Smart Room [31] using computer vision, speech and other sensory modalities to identify people and interpret people's actions in the room. The Robotic Room [32] is a robot

system in the configuration of a room that offers basic support to a person living in that room. The sensors installed in the room monitor the activities of the person and determine what help the person would need. A seven-DOF mobile robot with a head, two arms and a torso can express simple emotions through the movements of its arms and the head.

Social Robots

Social robots are the robots mainly designed for the intention of social interaction with human. Their physical bodies and mobility provide them movable perception and action in the physical world and release human from the limited interactive spaces, a common restraint of the embodied conversational agents and intelligent environments. The humanoid robots resembling human in appearance and behavior have shown compelling demonstrations, such as Honda's child sized ASIMO and Sony's entertainment robot SDR-4X [13]. The humanoid robots used in behavior research may be the full-body systems [33] [34] [35], the upper torsos [36] [37] [38], the expressive faces [3] [39], and the robotic dolls [40].

Social robots have applications in service [41], entertainment [6], and education [42]. They are also used as tools to study the behaviors and mind models of humans [36]. Another way is to use the robots as social partners of the human. The robots change the behaviors and feelings of humans through social interaction, such as in psychological therapy [43].

Remarkable variations of the human-machine interaction present in the roles played by the social robots. According to Breazeal [44], the social robots are divided into four subclasses: socially evocative robots [7], social interface robots [45] [46], socially receptive robots [47] [48] [49], and sociable robots [40] [50]. These robots have increased design complexity to support the social models that are ascribed to them by human.

Like the embodied conversational agents, social robots taking part in communication with humans show some or all of the following characteristics of human-human interaction:

- They can communicate with humans in natural languages.
- Their physical bodies are appropriate for massively making use of nonverbal behavior in conversation.

- They have perception to human emotions and can express their emotions through verbal and nonverbal behavior.
- They establish and maintain both social and physical relationships with humans.
- They exhibit distinctive personality and character through the appearances and behaviors.
- They can learn from human during interaction.
- They may have varied autonomy for different applications.

1.4.2 Wire-Suspension Mechanisms

The wire-suspension mechanisms are robotic systems that are not directly driven by motors but by multiple parallel wires. By coordinately changing the lengths and the suspending points of the wires, the suspended object or the manipulator platform can move in space.

The driving method of a wire-suspension mechanism is different from that of serial or parallel robots consisting of multiple rigid bodies directly driven by motors at joints. The directions of the forces acting on the wire-suspension mechanisms are determined by the spatial positions and orientations of the wires. The suspended object or the manipulator platform is driven by the tensions on the wires. The wire-driven method has advantages in workspace, payload capability, and speed [51] [52] over common serial robots on which the motors directly drive the links. Hence, wire-suspension structures are also applied to robot pose measuring devices [53], hepatic devices [54] [55], and wire-driven manipulators with various configurations [56] [57] [58] [59] [60] [61] [62]. The structures of the parallelly wired suspension mechanisms can be categorized into either fully constrained or underconstrained ones according to the configurations of the wires [63]. If both the positions and orientations of the suspended object can not be changed in any direction when the lengths and suspending positions of the wires are fixed, the mechanism has a fully constrained structure. Otherwise, it is an underconstrained mechanism in which the suspended object may move under the external forces even though there are no changes on the lengths

and suspending positions of the wires. A formal method for distinguishing the completely constrained mechanisms from the underconstrained mechanisms is proposed in [64].

1.4.3 Motion Generation Methods

Body movement as an observable function of the physical agents can be produced with various underlying methods of motion generation. From long before, many different methods and skills have been explored by people to generate movement effects on inanimate artificial mechanisms like the puppets, animated characters, and humanoid robots. They range from traditional animation techniques of keyframe to sophisticated learning algorithms integrating kinematic and dynamic properties of the humanoid robots. People attempt to provide the robots and other artifacts with the capability to behaving and acting like their counterparts originated in nature. Several successful examples representing recent advances on this aspect include canine behaviors presented on Sony's AIBO [65], human-like walking movement on Honda's humanoid robot P2 [35], and facial expression on the face robot Kismet [50]. The major differences among them are presented on several aspects: direct intervention from humans, closeness to natural behavior, flexibility for manipulation, and easiness of representation and implementation.

Keyframing and choreography [66] method is one of the fundamental techniques for character animation. Physical or geometrical models of characters and environment commonly are not required. Directly editing motion sequences is a way to incorporate human features into the behaviors of the physical agents.

Inverse kinematics is a modelling method that determines the joint variables from the desired trajectories. Solutions to joint angles obtained through inverse kinematics calculation are precise [37]. The movements generated from inverse kinematics may look mechanical because human movements in reality are also influenced by the physical law and the biomechanical properties of the human body.

Dynamics constructs the physical models of robots or animated characters and compute

their motions according to dynamics principles [67]. Applying the dynamic model to physical agents can improve control performance in terms of force and acceleration or adapt robots to the environment [68] [69] [70]. However, the motion equations become complex when the number of interactive body parts increase and the modelling process requires precise measurement to the physical properties of the physical agents.

Sensor-based control methods designed for the manipulators such as the force field [71], impedance control [72] and the proportion-integration-differentiation (PID) control [73] may be applied to low-level control of local controllers as they are not able to represent high-dimensional movements.

Optimal control defines an objective function and make motion control as an optimization problem [74]. As it is often not clear what optimization rules should be used, motion generated with optimal control on the physical agents may look differently from that of humans.

Motion capture [75] [76] and digital puppetry [77] are motion input, acquisition and modelling techniques that are widely used in producing realistic motions on both animated characters and humanoid robots [78] through motion transformation. The common steps to transform human movements to the humanoid robots consist of selecting the type of motion capture, processing sampled motion data with noise, building a kinematic representation of the demonstrator, extracting motion parameters from the human motion, adjusting the parameters for the humanoid robots, updating the movements of the humanoid robots through optimization algorithms, and synthesizing short motion pieces into large motion sequences [76]. It is a way to reuse existing motion data as a source of producing human-like behaviors on the physical agents [79] [80] [81]. Yamane [82] implemented human-like movements on a marionette using captured human motion data.

Life-like behaviors can be produced by simulating deep level mechanisms of behavior generation [83] [84], emotional model of characters [85], and biomechanical model of body parts [86] [87]. The principles of human motor control [17] [86] [88] provide inspirations to humanoid robot control with complex body structures. Complex behaviors may be built on simpler behavior modules and acquired through imitative learning. Recent findings suggest

the presence of functional modules in the planning and in the execution of actions [89]. Williamson [90] defined four postural primitives for a six-DOF robot arm in terms of its six joint variables. Mataric [91] implemented behavior-based control over a 20-DOF humanoid simulation through a behavior collection including three primitives which are implemented using different control methods. Movement primitive is a compact representation of action sequences for generalized movements that accomplish a specific goal or task.

A variety of explorations to primitive behaviors on the robots have been performed, some of them follow the motor control theory that behaviors should be treated as dynamic systems. The pattern generator scheme for locomotion [92] can generate rhythmic movements. Basis field approaches for limb movements [73] and more recently autonomous nonlinear differential equations for rhythmic movements regard the primitives as equilibrium points or attractors.

Imitation is the process of learning new movement patterns and skills by observation [93]. Imitative learning has been applied to humanoid robot control [78] [89] though many challenging problems are still waiting to be solved [94] [95]. Atkeson [96] and Schaal [48] demonstrated in the experiment of balancing an inverted pendulum on the hand of a robotic arm that learning by imitation is most effective when the demonstrated movements are used to tune an internal model of the robot.

1.5 Summary

Nonverbal communication, especially body movements, takes an indispensable role in human communication and should be applied to human-machine interfaces as an intuitive and comfortable medium. The work presented in this thesis is an attempt to construct a physically embodied agent capable of expressing information through its gestures and body movements. The goal of the research is to study behavior generation and expression of the expressive physical agents, especially with different body structures. Design issues relevant to body structure and movement generation of the physical agents that convey information to human observers in a manner of nature and intuition are studied.

We will develop a means of motion control for the physical agents to produce a limited collection of communicative behaviors. The method should be generic to be applied to many physical agents having different body structures and physical properties. The communicative behaviors of the physical agents are presented in human-like movements with believable effects. Information display is performed only through body movements without aids from other communication modalities like speech. The physical agent we made is an avatar in the form of a robot puppet performing one way communication with human. To give the user enough control over their behaviors, they are not intended to be made autonomous and manoeuvrable in the environment.

Though the body movements of the robotic puppet are studied individually as one of the major codes in nonverbal communication, the body movements of humans are commonly operated in concert with other modalities. To treat it as an instinct entity is a way of focusing its role in expression, as well as a matter of convenience. We expect our robotic puppet can deliver information through human-like behaviors, as well as the methods developed to control the robotic puppet can provide some theoretical guidance to the design of other physical agents for nonverbal communication applications in education, service and entertainment.

Several problems that have been explored in the making of the robotic puppets have widespread significance in robotics and computer graphics. The degree-of-freedom problem is a classical topic in behavior representation, generation, and control for human, robots, and graphic characters. The correspondence problem is essential to behavior portability and transformation among embodied agents with different body structures. The robotic puppets may be used as a computer-aided tool for choreographing and recording puppet show in teaching and entertainment, or a general 3D display. The outline of the remainder of this document is as follows.

In Chapter 2, The design specifications to the structure and behaviors of the physical agents are made according to the design principle of niche space matching state space. The method of appearance selection is analyzed.

In Chapter 3, an analysis to the mechanical structure of the robotic puppet is performed

under the framework of underconstrained mechanisms. Its design is illustrated and compared with the method of puppeteer making puppets. Computation of the workspace and inverse kinematics are proposed according to the quasi-static conditions.

In Chapter 4, an agent-centered approach of movement generation and acquisition begins from the discovery to the motor capability of the physical agent. Motion synthesis of the robotic puppet is performed in a multilevel structure with temporal spatial properties separated. Coordination of the multiple parts of the puppet figure is achieved by constraining the symmetrical limbs.

To transfer existing motion sequences to the physical agents with different body structures, the correspondence problem in movement acquisition is discussed in Chapter 5. Metrics to measure the closeness between the movements are defined from both spatial and temporal aspects, and behavior acquisition is achieved by composing given motion sequences with the motor primitives of the physical agent.

The hardware and software structures of the robotic puppet are implemented to control its movements. Demonstration of a motion sequence is given in Chapter 6.

The final chapter summarizes the work and contributions to design and construction of the expressive physical agents through body movements. It also discusses possible extensions to this study for different directions.

Chapter 2

Design Considerations and Appearance Selection

The design criteria of the physical agents intended for behavior expression are determined according to the basic design principle that the state space should match the niche space. The factors influencing appearance selection for the physical agents are discussed from the perspective of human's psychological responses.

2.1 Design Considerations

The design process begins from clarifying the task requirements and structural constraints of the physical agents to behavioral expression because the possible designs should fulfill the intended tasks. That is, the design space should match the niche space. To match the design space for the niche space means that the structure and behaviors of a physical agent should be well-qualified to the intended tasks but should not overpay too many efforts and resources.

There may be no single optimal solution to the design process that looks for a match between the niche space and the design space. Evaluation to a design can be done by comparing several dimensions important to the physical agents. In practice, the niche space and the design space are composed of the factors that can exert dominant and direct influence on the performance of a physical agent. The parameters describing the design space should be selected as those factors that are closely associated with the intended applications. For the physical agents, these include a series of options conducting to usable and

Figure 2.1: Physical agents compared in one design space

comfortable human-machine interaction, such as the appearance, the body structure, the control method, and the behaviors. To illustrate the possible compositions of design criteria, different types of robots that can perform social interaction with humans are compared in the three-dimensional design space represented with the three design parameters of behavior diversity, appearance, and interaction (Figure 2.1). It can be seen from this figure that the robotic puppet exceeds the mobile robot or the face robot in behavior diversity because the robotic puppet has a larger number of degrees of freedom (DOFs). This means the robotic puppet has more potential of presenting many different movement patterns to encode messages for expressive tasks than a mobile robot. The appearance of the robotic puppet is more similar to the human than a mobile robot. A face robot or a humanoid robot has better interaction with humans than the robotic puppet as they can communicate with humans through multiple channels and set up closer interactive relationships with the environment. A robotic puppet for behavior expression task has poor perception to the environment because it has no sensors.

2.1.1 Niche Space

The basic requirements to a physical agent designed for social interaction are usability and comfortability because the performance depends on not only its behaviors but also people's perception.

The usability means that the agents could be used as a tool to deliver information to the observers through their body movements according to the intention of the users. The users can represent and input the intended behaviors into the physical agents. The agents can make body movements in the real world according to the users' commands. So the agents must have a physical body and enough encoding capability through many different behaviors to express the information. The usability of the behavior expressive agents requires:

- The expressive agents have physical embodiment in the real world.
- The agents have a behavior repertoire large enough to encode the information.
- The behaviors of the agents are controllable.
- There is an user interface for the users to represent and designate the intended behaviors.

The comfortability of the expressive agents requires the information to be delivered to the observers in the manner of natural communication. There would be no efforts to be made for the observers to accept the information and understand the meanings. The observers may feel easy, intuitive, and delightful in the communication process. To make the behavior expressive process comfortable, the agents should have physical embodiment in the real world as the media of information expression. Like human bodies, the physical bodies of the agents produce spatial effects. Both the structure and the behaviors of the agents should have humane features which may produce illusion of life on the observers. Communicating with human in natural manner is likely to make human comfortable. The comfortability of the agents consists of following aspects:

- Physically embodiment in real world. This point consists with the requirements of the usability.
- Humane features in the body and behaviors of the agents in order to produce illusion of life on the observers.
- Personal features of the agents in the behaviors in order to produce illusion of life.
- Friendly user interface for the users to represent and input the behaviors.
- Limited interaction with the observers.

2.1.2 Design Space

The making of the express agents includes a series of decisions to the design parameters of the appearance, the mechanical structure, the behavior generation methods, and the electronics. The design space of the expressive agents includes:

- Whether the expressive agents have physical bodies or not?
- Is the appearance human-like or not?
- The number of DOFs.
- Is perception to the environment necessary to behavior expression?
- The degree of autonomy in behavior generation.
- The ways of behavior representation.
- The methods of behavior generation.
- The types of expressive behaviors.
- User interface for behavior input.
- Hardware structure.
- Software composition.

2.2 Appearances

The appearances of the physical agents should be selected to facilitate the human-machine communication and fit for the behaviors the agents are able to perform.

2.2.1 Degree of Embodiment

There are a number of agents that are designed to interact with human in different application domains such as web agents, synthetic characters, and social robots. They can be categorized into either embodied or disembodied agents. The robots that have physical bodies and the animated avatars that have simulated or animated bodies [97] are two types of embodied agents. Disembodied agents exist as computer programs interacting with humans through speech or text entered at the keyboards and displayed on the screens. The embodied agents have the advantages of interacting with people through multimodal communication, such as gesture, facial expression, gaze, and body movements. The non-verbal cues can carry the messages on their own during interaction with people without using spoken language, or can be used by avatars to complement or enhance their verbal messages. The comfortable interface of the embodied agents suggests they have stronger capacity in social interaction with humans than the disembodied agents. So the agents for social interaction tasks are designed to have physical or simulated bodies, which also introduce structural and computational complexity. For example, detecting and understanding human social cues through multimodal sensing is one major issue to be addressed for the embodied agents involved in two-way human-machine interaction. However, this problem does not exist for the disembodied agents.

Embodiment means that the social capability manifested from an agent's behaviors can not independently exist in the form of abstract algorithms but requires a body to associate the meaning of the behaviors with the context. Through embodiment, an agent sets up a repeatedly interactive relationship with the environment, which specifically refers to the human for social interaction tasks, so that both sides can influence each other. For conversational agents, the animated figure body is enough for their tasks of discoursing

with people mainly in speech. Other agents that are required to have mobility and non-verbal communications are designed to have physical bodies. One example of the use of embodiment is the entertainment robot - Sony's AIBO [65]. As the pet robot has a physical body, it can move around to express its intimacy to its owner using spatial distance and accept the owner's touch through the sensors on its body.

Embodiment also means the mutual influence between an agent and the environment, which mainly refers to human in social situations. The nature and degree of embodiment have significant impact to the performance of the physical agents in social interaction. Their body structures are designed to have different complexity to cope with a wide range of applications that require different levels of interaction with human. For example, humanoid robots often have tens of DOFs in order to implement human-like behaviors while several DOFs are enough for service robots moving in the office or hospital environment. The humanoid robots have more capacity to interact with human than the service robots.

Embodiment is not solely related with body structures and behaviors of the agents. Another factor in determining the degree of embodiment is the surroundings of the physical agents. For example, the face robot Kismet [47] that can display many facial expressions has 15 DOFs. But it may not be as useful as a mobile robot for assistant task. Embodiment requests that the body structure of the physical agents should match the intended tasks.

Our robotic puppet designed for the task of presenting information with body movements has limited embodiment. Its body structure (Figure 1.3) is similar to that of humans on the limb level. It has four limbs with each of them consisting of a connection between two linkages. The figure body has nine parts connected through threads. Each joint has three DOFs and is treated as a spherical joint. The puppet figure has enough structural complexity to generate plentiful gestures and body movements to encode the messages for expression, but its structure does not have too many details. Hence, the complexity of its body structure is appropriate for our aim of studying behavior expression.

2.2.2 Anthropomorphism

We inherently like to interpret the actions of others from our perspective. It is taken for granted that the meanings of movements would depend on the observers. This is the phenomenon of anthropomorphism, the tendency to attribute human characteristics to inanimate objects, animals, and artifacts with the view of helping people rationalize their actions [5]. In physical agent design, anthropomorphism is utilized to enrich the behavioral characteristics of the agents in order to make our understanding to their movements easier. To make human-machine interaction comfortable is the major power of applying anthropomorphism to physical agent design.

The physical agents designed to have human-like or animal-like morphologies help initiate communication and interaction through providing people intuitive and comprehensible social cues. People will treat a physical agent with human-like appearance differently from a pet robot. Anthropomorphism offers the physical agents life features that are significant for a machine to be socially engaged. The resemblance on the appearances and behaviors between the physical agents and their natural counterparts causes illusion of life. Hence, it is advantageous to add life characteristics in physical agent design and implement human-like behaviors on the anthropomorphic, i.e., human-like, body structures. The robotic puppet designed for conveying information to human through body movements obviously belongs to this category. Another benefit to mimic the human body in physical agent design is that the agents with body structures similar to that of humans have more potential in acquiring and expressing human-like movements.

There is no doubt that the employment of some anthropomorphic qualities, either in appearance or behavior, is one of the major considerations in designing the physical agents whose tasks are to communicate meaningfully with people. The problem is that at what degree the resemblance between the physical agents and humans is enough for the intended tasks. The intuitive impression is that, in a successful design, the physical agents should be made as similar as possible to humans [43]. Many efforts on both structural and behavioral design for humanoid robots have been made toward this target [47] [78] [98]. Sometimes

the unspoken thought behind this quest for perfect matching to humans in physical agent design may be a belief that mimicking human is a way to control the humanoid robots [68] but not an intention to elicit anthropomorphism in human-machine interaction. It is also possible that people use this method to either control the humanoid robots or facilitate social communication. The intention of applying human features in the physical agents depends on the applications. If the anthropomorphic agents are used as the replacement of real man, they must accurately mimic human on the required aspects. The applications requiring realistic human models include using the robots as tools to study humans [36] or as substitutes for humans [99], artificial life [87] for animation, and animatronics [100] for film-making.

If the tasks of the physical agents are mainly interaction with humans, there are several reasons for us to give negative response to the opinion of making the physical agents perfectly match humans. The intention of applying anthropomorphism to structural and behavioral design for the physical agents is to facilitate the interaction between humans and machines rather than to duplicate humans and build an artificial one.

2.2.3 Uncanny Valley Phenomenon

The anthropomorphism phenomenon suggests that incorporating human-like characteristics into physical agent design might be of benefit to the interaction with humans. It is meaningful for the anthropomorphic agents to keep some degree of resemblance to human. The second factor that must be taken into account in physical agent design is the "Uncanny Valley" (Figure 2.2) phenomenon discovered by Mori [101] when studying psychological responses of human observers to humanoid robots. The major point of the Uncanny Valley phenomenon is that human's response to the physical agents from a non-realistic portrayal to reality is nonlinear. When the resemblance between the anthropomorphic agents and humans is approaching a perfect matching but has not reached it, the believability [98] of the physical agents falls down and forms a steep valley. At the Uncanny Valley, subtle imperfections on the structures or behaviors of the physical agents will reveal the fact that the physical agents are not real ones and will cause unpleasant or even highly repulsive

Figure 2.2: The Uncanny Valley phenomenon [101]. Human’s responses to anthropomorphic artifacts may experience a sudden change.

feeling in humans. The lifelike physical agents suddenly exposing the unpleasant nature destroy people’s expectations.

The Uncanny Valley phenomenon in human’s response to the physical agents involved in communication leads to the conclusion that it was not desirable to make the anthropomorphic agents as similar as possible to humans. This accounts for the observations that the physical agents with caricatured or iconic appearances are often more effective in communication than those with lifelike appearances. The appearance of humans is not the best and only selection to the appearance of the physical agents for expressive tasks. We should avoid uncanny valley through making the appearances of the physical agents be easily recognized as artifacts and should make them approximate the highest point before the uncanny valley by offering them engaging appearances. The robotic puppet that does not try to accurately imitate the appearance and behaviors of humans obviously satisfies this requirement.

2.2.4 Balances between Appearances and Behaviors

The physical agents that try to closely mimic human appearances and behaviors may unnecessarily restrict the structural design and actual functionality. There exist many options to appearances and body structures of the expressive physical agents. Structural design constrained by the only direction toward the complexity of human body ignores the fact that other selections, such as iconic and caricatural appearances, may be also suitable

Figure 2.3: Different appearances cause different expectations to the behaviors

for the intended expressive tasks. In addition, all robots are developed by far as tools or social partners to help people with their useful aspects. Making the physical agents perfectly human-like implies that they would also process the unwanted characteristics of humans, such as the weakness in human body structure.

The appearances and body structures of the physical agents are important because they constrain the motor capability and bias the context of communication. Motor capability is the potential of all body movements through which an agent does work or expresses its internal states. Major metrics of motor capability are workspace, extreme positions, ranges of velocity and acceleration. The choice of appearances and body structures also constrain the human's ability to interact with the robot. For example, the face robot Kismet has highly emotional expressions, but it is unable to interact with people through touch, manipulation, and displacement because it is designed to be a talking head [47].

The structural and behavioral design of the physical agents need a balance between the appearances and the performance. The appearance of a physical agent provides cues for the context of human-machine interaction and evokes people's expectations to the performance of the physical agent. A complex body structure excites people's high expectations to its performance that is expressed through complex behaviors. An appearance similar with

human easily elicits anthropomorphism. But strict resemblance to human tends to increase people's expectations to the performance of the physical agents. Under this condition, people will use more natural and unconstrained communication to interact with the physical agents. It will take the designers a lot of efforts to improve the performance of the agents to the level that can fulfill people's expectations. For instance, a humanoid robot will elicit more expectations to the robot's social interactive capability than a robotic puppet. When communicating with a humanoid robot, people will feel less constrained to use the manner of conversing with another person. However, if the performance of the physical agents fails to satisfy people's expectations during interaction, people will lose most of their trust to the physical agents and are likely to experience confusion and disappointment. Employing excessive natural modalities in communication tends to disclose the nonliving substance under the life-like exterior. Such a design of pursuing realistic humans should be avoided because it unnecessarily increases the requirements to behavioral design. To guarantee an engaging interaction, the physical agents should regulate the people's expectations by providing a less realistic but appealing appearance. In Figure 2.3, the puppet figure having a young man's appearance may evoke more concrete expectations to its behaviors among people than the other puppet figure without costumes. Its movements should be agile and sturdy, just like an energetic young person. Otherwise, it will not come up to people's expectations. For the other figure abstractly representing the human, whatever human-like movements are acceptable. Hence, the abstraction in appearance provides more space for behavioral design.

So what we need is not a synthetic human but a simpler one that has a suitable mapping between the appearance and the behavior. The unnatural features revealed by the appearance of a physical agent help people setting up the impression that the physical agent has limited capacities of social interaction. People have to selectively use natural communication when the interaction begins.

Eklundh [7] noticed that the robot moved around in a non-predictable way may cause people to interpret it as acting intentionally, and as if it had personality. Scheeff [45] also confirmed that people regarded the behaviors of the robot rational even when the robot

was malfunctioning and moving erratically. This suggests that behavior repertoires of the physical agents did not require precise likeness to human behaviors. Behavioral design can take advantage of this point for expressive tasks.

Our quest for expressive agents has no intention to duplicate either the appearances or behaviors of humans onto the physical agents, but to facilitate behavioral expression by adding human traits on the physical agents.

2.2.5 Focus on Movement

Like the process of puppet making [20], physical agent design can be roughly divided into two stages: body structural design and behavioral design. The appearances, morphologies and mechanical structures of the physical agents are determined in the first stage while the behavioral design stage selects the methods to input and express body movements. Though it is obvious that the two stages in physical agent development are closely inter-related, the question is which one has more influence in eliciting anthropomorphism. Both Mori's experiment [43] and psychological studies [102] on anthropomorphism demonstrate that movement is more effective in producing illusion of life than appearance. So we put the emphasis of physical agent design on the behaviors, and sufficiently exploiting motor capability is the major component of expressiveness.

Different from the common way to express emotion and personality through the body movements biased by the states of internal models, our method for expressing emotional states is closely associated with the motor capability of the physical agent. We use extreme postures to express highly emotional states, easy postures to reveal stable internal states.

2.2.6 Characteristics of Human Movements

As expressions of motor control, the body movements of human have four identifiable characteristics: the flexibility, uniqueness, consistency, and modifiability [16]. They distinguish human movements from the unnatural movements of robots that are programmed to repeat precise trajectories. They are important indicators to behavioral design for the physical agents because life-like effect is a desirable feature of the agents' movements.

Flexibility of movement is the capability of selectively using different body parts from many available DOFs to perform the same movement. Uniqueness is the phenomenon that two movements are never performed in exactly the same way. The consistency of movement is the relative stability of movement on temporal and spatial aspects from one performance to the next. More or less variations are involved in the movement which will not be recognized as different actions or tasks. Modifiability of movement is the capability to change the pattern of movement even when it is on the fly. These characteristics of movement constituting the foundation of complexity and diversity of human movement may be used to evaluate the motion generated on the robotic puppet. The methods applied to motion control of the robotic puppet should take into account the multiple characteristics of human body movements in order to functionally achieve all or part of them on the movements of the physical agents. Many expressive tasks require maintaining the essential features in human movements when they are re-produced on the physical agents. The characteristics of human movements remind us that only the contents of behavior expression is not enough to implement human-like movements on the physical agents, the manner in which the contents are displayed is also necessary.

2.2.7 Spatial Relationships

Another factor must be considered in physical agent design is the locale where the interaction happens. It depends on the tasks the physical agents will perform. Four basic conditions are identified:

- (1) The physical agents interact with the users face to face in the same place. One example involved in this kind of interaction is the face robot Kismet [50].
- (2) The physical agents have enough autonomy to move in the places where the users may or may not be present, such as the entertainment robot Sony's AIBO [65].
- (3) The physical agents are teleoperated by the users at some places far away from the locations where the physical agents act, such as the service robot CERO [7].
- (4) The users control the physical agents far away. Through the physical agents, the users can communicate with other people in the proximity of the physical agents. The physical

agents become the avatars of the users. The remotely controlled mobile robot Sparky [45] is such a telepresent robot [46].

Discrimination is made between the teleoperation robots and the telepresent robots based on the fact that teleoperation robots do not involve in social interaction with other people, while the users of telepresent robots communicate with other people through the physical agents acting in remote places. The robotic puppet is a telepresent physical agent that is made to deliver user's information to observers.

2.2.8 Two Kinds of Avatars

The behavioral expression of the physical agents is a form of nonverbal communication. Communication happens as a process of social interaction that transmits information through symbols encoded by the sender and interpreted by the receiver. The symbols may be in any form, such as speech, facial expressions or body movements, but they must have shared meanings between the interaction participants. The communication can not work if the cues or signs from which inferences are made are randomly selected.

Figure 2.4: The robotic puppet in one-way communication to humans

Communication can operate as either a one-way process (Figure 2.4) or a two-way process. One-way communication happens in the situations that the sender needs no feedback from the receiver or receivers. One case of one-way communication frequently seen is news broadcasting. Some avatars are made to perform one-way communication, such as the information assistants and pedagogical robots. Their tasks are delivering information to the audiences through natural modalities such as speech and non-verbal cues. Once started to make a presentation, they need not check the status of the observers and use them as feedback to modify their behaviors. Or they do not like to be interrupted by the audiences. They assume that the audiences were able to correctly receive and understand the messages they send. Our robotic puppet is such a one-way avatar that conveys information to human

Figure 2.5: The social agents interact with humans in two-way communication

observers but does not have perception to human's response. Information is encoded into its body movements and delivered to human observers through its performance.

Most interpersonal communication in daily activities takes the two-way mode, such as those in conversation and games. Participants have perception to each other and use the response of the other person involved in the interaction as feedback. The communication continues in the reciprocal process. Social robots, conversational agents, teleoperation [103], virtual reality, intelligent environments [30] and tele-embodied avatars [46] are the interfaces using two-way communication (Figure 2.5). Social robots and conversational agents emphasize the interaction with human with natural modalities. The tele-embodied avatars transmit people's response from the avatars' locations back to the users.

However, fluent information delivery through body movements in two-way communication may be blockaded if (1) the agent lacks an adequate movement vocabulary to encode the information it wishes to express, or (2) the body movements of the agent can not be understood by the observers, or (3) multiple interpretations associated with the movements tend to cause misunderstanding, or (4) the agent can not sense or understand the responsive messages sent by human. The most intractable part in two-way communication should be the last one because it is much more difficult for the agent to understand human than for human to comprehend the agent's behaviors though many efforts have been made to improve the agent. There is a remarkable unsymmetry of cognitive capabilities between the agents and human. This problem is avoided by the robotic puppet advocated to one-way communication of behavior expression.

2.3 Summary

It is clear from previous discussion that the structural design of the physical agents is the result of a series of tradeoffs between related, and often conflicting design parameters. The wide range of social interaction tasks asks for physical agents with different appearances, functions, and behaviors in order to reach a balance between body structures and the intended tasks. The notion of embodiment suggests that the physical agents could have body structures with various complexities to set up various levels of interactive relationships with humans. Due to anthropomorphism, the appearance and behaviors of a physical agent should have a little similarity with human, while the uncanny valley phenomenon refutes the opinion to precisely mimic human's appearance. To maintain people's engagement in the interaction, the appearances of the physical agents should be made appealing but not very similar with humans. To constrain the natural modalities people use in interaction with the physical agents, the appearances should be a little abstract. These design factors unavoidably result in highly diversified body structures of the physical agents. The design principle of niche space matching state space is applied to the physical agents in order to make the appearances, structures, and behaviors appropriate to behavior expression tasks.

Chapter 3

The Mechanics of The Robotic Puppet

The robotic puppets are realized as motor-controlled marionettes that combine traditional puppetry with robotic technology. The mechanical part of every robotic puppet consists of a multiple link mechanism in an underconstrained parallelly wired suspension structure. Compared to the conventional robots driven by electro-magnetic actuators, using parallel wires for robotic systems have the advantages of light weight, simple mechanical design, and effective performance with low rating motors. Major issues in designing wire-suspension mechanisms are the selection of wire number and the arrangement of the wires and the motors. The robotic puppet ROMS-I is used to illustrate the process of structural design. A list of assumptions are made to the torso part and the mechanical properties of the puppet figure. The body movements of the puppet are limited to quasi-static situation. Based on these assumptions and limitations, the inverse kinematics is calculated according to statics.

3.1 Structure Selection

The robotic puppet is designed to have an underconstrained wire-suspension structure that can use a smaller number of wires, the control variables in the system, to operate the puppet figure with a larger number of DOFs. For example, ten strings are used in the robotic puppet system ROMS-I (Figure 1.3a) to produce movements on the 30 DOFs figure body.

Figure 3.1: Two fully constrained wire-suspension mechanisms: (A) "FALCON" [51] and (b) "WARP" [56]

Taking an underconstrained wire-suspension structure in the robotic puppet is a way to address the degrees of freedom problem from the prospective of structural design. For fully constrained mechanisms, there is definite kinematic relationship between the position and orientation of the suspended objects and the wire configurations. The suspending points of wires are commonly fixed and the spatial states of suspended object totally depend on the lengths of wires, such as the "ROBOCRANE" [56] and "FALCON" [51] wire-suspension robots. Similar to the parallel robot manipulators, their forward kinematics is more difficult to obtain than the inverse kinematics. However, fully constrained mechanisms demand a large number of wires and carefully selected wire arraignment. There is limitation to the least number of wires for different tasks in terms of DOFs in fully constrained mechanisms. To make one suspended object to move along all the six-DOF in space, at least seven wires are needed [63] [64]. The "FALCON" robot uses seven wires and the "WARP" [56] robot uses eight wires to achieve six-DOF motion. The suspending points of wires must come from different directions surrounding the suspended object and thus restrict the workspace. To enlarge the workspace, it is possible to add redundant wires to the system. The wires of a completely constrained mechanism actively exert tensions on the suspended objects. In underconstrained mechanisms, the tensions are passively generated on the wires under the gravitational force of the suspended object. The situation of wire-suspension mechanisms

Figure 3.2: A three-wire underconstrained robot [63].

is similar to that of force-closure grasps [104] which was originally proposed for the multi-fingered robotic hand to grasp polyhedron objects as both of them constrain the motion of the object by exerting forces and moments in order to keep the controlled object in equilibrium. The major difference between them is the directions of the exerted forces that point to the object in a grasp or away from the object in suspension. In fully constrained mechanisms with only one suspended object, the force-closure method can give answers to how many wires are enough and where to suspend the wires in order to make the system have the desired number of DOFs while preventing negative tensions on the wires. According to the force-closure theorem [104] [105], the sufficient and necessary conditions to fully constrain one rigid body with n -DOF motion in space request n independent wires in at least $n + 1$ wires that are suitably arranged.

The force-closure theorem was proposed for the condition of constraining one object. In order to determine how many wires are required for a motorized marionette system taking a fully constrained structure, we need to resort to the Kutzbach constraints for spatial mechanisms discussed later.

Underconstrained mechanisms use less number of wires whose suspending points can be arranged on the same side, commonly on the top of the frameworks. It is possible for an underconstrained mechanism to have six-DOF motion using less than six wires (Figure 3.2). The minimum number of the wires to make the suspended object move in six-DOF is three if the suspending points can move in space [63]. The moving suspending points increase the number of inputs to the systems. To control the suspended object moving in six DOFs, at least six inputs of wire lengths and movable suspending points are necessary. The

wire suspending points of the three-wire underconstrained mechanism in [106] are movable in three-dimensional space. While a completely constrained mechanism can control the position and orientation of the suspended object by changing only wire lengths, three-wire underconstrained mechanisms can not perform both translation and orientation in space by only changing the wire lengths. When only the lengths of the wires change, the suspended object in the underconstrained mechanism has uncontrollable states [61].

For underconstrained mechanisms which have low stiffness, dynamics has salient effect on inverse kinematics computation and control. Motion on other DOFs is governed by the dynamics of the suspended object and wires. The inverse kinematics problem needs to involve dynamics and geometrical constraints. The positions of movable suspending points can be obtained through inverse kinematics calculation in which the tensions on the wires are the major variables. To realize the wire tension vectors, the wire suspending points must be along the lines of the calculated forces from inverse kinematics. When the force directions change, the suspending points must be able to move to new positions on the extended line of the force vectors in order to provide the tensions passively. The tensions on the wires are generated passively due to the gravitational force on each suspended object.

A calculation method for determine how many wires are appropriate for an underconstrained mechanisms, especially when the suspended objects are linked rigid bodies, has not been proposed. The force-closure method that can give appropriate wire number according to the required DOFs is not valid for underconstrained mechanisms because the total wrenches on the suspended object can not form a closure though they are symmetry. If the robotic puppet uses fully constrained structure, there will be a large number of wires that may wrap with each other during the movements. Hence, the structure of the robotic puppet follows the driving method of the conventional marionettes that must take the underconstrained structure coming from puppeteers' accumulated experience long passed down from history.

Figure 3.3: The structure of a marionette. It consists of an articulated body, a control bar and appropriately arranged strings. The shoulder strings are first attached and used as the reference in connecting the knee strings and hand strings.

3.2 Puppeteer's Method of Structural Design

Puppetry belongs to the few art forms in which the performer is also responsible for design, construction, plot, and choreography of puppet [20]. The entire procedure the puppeteer experience is similar to our situation in designing and controlling the expressive physical agents. Corresponding to structural design and behavioral design of the physical agents respectively, puppet construction and dramatic presentation are two linked but divisible stages in puppetry.

In construction stage, some experiences are required to make and connect the four major components of a traditional marionette: the body parts, joints, the control bar, and strings into a workable puppet (Figure 3.3). The materials of a puppet figure may be papier-mache, cloth, and wood. The proportions between the body parts of a puppet portraying the complete human figure is roughly the same as the human but not excluding exaggeration and distortion added to the puppet character in certain instances. The mass and mass distribution of the puppet are two factors that should be considered in structural design and motion control. The body parts may be connected with either hinges or wires. Each joint may have one to three DOFs. Commonly, the one-DOF joints are in the elbows, knees and ankles, while the two-DOF or three-DOF joints are at the neck, shoulders, and wrists. The control bar, the so called "perch" in the West or "Gou Pai" in the Chinese puppetry, is a frame made into a cross or an airplane shape for binding the strings and suspending the puppet figure. A puppeteer uses the control bar to operate the strings that drive the puppet figure. The motion of the puppet figure can be produced by either altering the position of the control bar in full body movements or operating a part of the strings to control the movements on individual body parts. The tilt or balance of the entire control bar may generate the actions of bowing, head turning, and kneeling on one knee. The arm waving, lifting, and leg movements can be achieved by manipulating individual strings.

The range of various movements depends on the design and arrangement of the joints, the control bar, and the strings. The length of the strings depends on several factors: the height of the marionette, the size and type of the control bar, and the movements to

perform. For example, if lying down action will be shown, longer strings would be necessary.

The stringing method has obvious impact on the movement range and operation of a puppet. In wooden marionettes, the strings need to be attached to small rings and screw-eyes fixed in appropriate positions in the control bar and the puppet figure. The strings tied in place of the shoulders take the main weight of the figure body in all movements (Figure 3.3). Two strings are tied evenly so that the shoulders hang level when the marionette's feet are just raised from the ground. These two strings are often suspended to the same position on the control bar. Thus, sideways or forward tilting of the control bar will not damage the stability of the puppet figure. After the shoulders strings have been connected, all other strings are measured and adjusted in relation to the shoulder strings and must be tied carefully so as to keep the shoulder strings always tight. They are connected to the farthestmost tip of each link from the torso. For example, the knee strings run from the control bar to the screw-eyes fitted just above the front of each knee joint, and the hand strings are attached to the palms.

3.3 Mechanical Structure of the Robotic Puppet

The structural design of the robotic puppet cares more about the stringing method than the puppet figure. The robotic puppet replaces puppeteer's manual operations with the motors.

The robotic puppet ROMS-I in Figure 3.4 consists of nine moving parts (Appendix A). The head and the torso form one part. They are made out of wood and connected with threads. There is no more constraint added to the joints to prevent them from freely rotating around three directions in space. Each of the eight joints at the shoulders, hip, knees, and elbows has three rotary DOFs. They can be treated as spherical joints. Hence, the puppet figure of ROMS-I totally has 30 DOFs: the torso part in space has six DOFs and the joints have 24 DOFs. Within the three-dimensional workspace of the robotic puppet, the positions and orientations of the body parts of the puppet figure can be controlled through changing the lengths and the positions of suspending points of the strings.

Figure 3.4: The stringing method of the robotic puppet ROMS-I. Two strings have the same lengths and support the weight of the puppet body. Each string is attached to the end of one link on the limbs.

Though some experiences from puppetry are adopted in current implementation of the robotic puppet (Figure 3.4), the design of the robotic puppet is, to a certain extent, different from a traditional marionette that is manipulated by translating and rotating the control bar and through directly pulling individual strings with the fingers.

The movement of the puppet figure is caused by the strings whose lengths and suspending positions are changed remittently by motorized pulleys mounted on a stationary frame that does not roll and pitch like the control bar used by a puppeteer. Two parallel strings attaching to the shoulders of the puppet figure support its weight and stabilize the suspended torso part. Its tendency of rotation in the sagittal plane or around the vertical axis caused by the movements of the limbs in 3D space is restrained by the shoulder strings.

The origin of local coordinate is set at the mass center of the torso part. The strings are attached to the ends of the links away from the torso and driven by the overhead motors. Details of the implementation will be introduced in Chapter 6.

3.4 Kinematics

The inverse kinematic problem for the robotic puppet is defined as the determination of the lengths and the suspending positions of the strings in order to make the puppet figure take the desired postures. The inverse kinematics problem of the underconstrained mechanisms is more difficult than fully constrained mechanisms because the positions and orientations of the suspended objects are defined not only by the geometric relationships between the links and the strings, but also by the gravitational force and dynamical effects of the suspended objects. Because of computational complexity in the dynamic properties of the robotic puppet, the inverse kinematics problem is solved according to static equilibrium. Equations involving variables representing system configuration are given by the geometrical constraints and the conditions that total forces and moments on the suspended objects are zero when in equilibrium. The solutions are valid for practical control of the robotic puppet when it moves in low speed.

The tips of the limbs have salient visual effects in behavioral expression. In many situations we mainly care about the positions of the farthest ends of the limbs from the torso. The structures of the arms and legs of the puppet figure are not redundant. Once the end points of the limbs are determined, the positions of the elbows and the knees can be easily obtained.

3.4.1 Degrees of Freedom

The robotic puppet has many joints and strings that provide it the abundance of body movements. It requires many variables to specify or represent the myriads of possible configurations.

Assume that the figure body consists of n rigid bodies and j joints, and the DOF of each

joint is f_i , ($i = 1, \dots, j$). Let F be the total number of DOFs of the figure body. According to Kutzbach criteria, the total DOF of the puppet figure before being suspended with wires is:

$$F = 6n - \sum_{i=1}^j (6 - f_i) = 6(n - j) + \sum_{i=1}^j f_i \quad (3.4.1)$$

The suspended figure body (Figure 1.3) of the robotic puppet ROMS-I has nine links connected with eight joints that are treated as three-DOF spherical joints. According to Equation (3.4.1), the figure body has totally 30 DOFs.

When the puppet body is tied down to the fixed frame using the strings, each string will introduce one constraint which is the distance (or length of the string) between the attaching positions on the frame and the puppet body.

Assume that there are k strings attached to the figure body from a fixed frame where the motors are mounted, then the number of the DOFs of the wire-suspended robotic puppet denoted as F_s equals to:

$$F_s = F - k = 6(n - j) - k + \sum_{i=1}^j f_i \quad (3.4.2)$$

The DOF of a suspended puppet F_s represents the freedom of movement that the puppet may have after it is tied down with the strings. In other words, these DOFs are not constrained by the strings. Without other form of input, these DOFs can only be determined through static or dynamic equilibrium under the gravity influence. The controllable DOFs of the puppet are, in fact, the k strings. The lengths of the strings are the control variables to the system.

In current implementation of ROMS-I, two of the total ten strings have fixed lengths and are used to support the weight of the puppet body. Another eight strings whose lengths can be changed independently are connected to the figure body. Thus, the robotic puppet system ROMS-I has 22 DOFs which are not constrained by the strings (Table 1.1).

At any moment of body movements, only part of the totally eight strings may be involved in the motion control of the robotic puppet. The other strings are loose and do not exert tensions on the suspended figure body.

Figure 3.5: The reference frames of a robotic puppet. Three reference frames are used in posture representation.

3.4.2 Reference Frames

The position of a rigid body is defined by the location of a point on that body and the body's orientation. The puppet figure and the human body that have branching structures can be treated as a system composed of multiple articulated rigid bodies [107]. The configuration of the puppet figure in space is defined by its (1) location; (2) orientation; and (3) joint configuration or the posture. When the puppet is driven by the strings to perform an action, the body changes its position in terms of the location, the orientation, and the posture.

The locations and body postures of the robotic puppet can be represented in three reference frames (Figure 3.5) according to its suspended structure:

1. *The Cartesian space* is the extrinsic frame $\{O\}$ representing the positions of the figure body relative to its immediate surrounding space along the X_o -, Y_o -, and Z_o -axes. The origin of the frame is set at a particular location on the frame supporting the puppet figure.
2. *The joint space* is the intrinsic frame $\{\theta_i\}$ of the figure body representing relative

positions and orientations of the articulated body parts in joint angles $(\theta_1, \theta_2, \dots, \theta_F)$, where F is the total number of DOFs in the joints of the puppet figure. The center of the mass of the torso part at (x_V, y_V, z_V) with respect to the frame $\{O\}$ is selected as the origin of frame $\{V\}$ fixed on the puppet body. All the joint angles on the puppet are defined with respect to frame $\{V\}$. The 4×4 coordinate transformation matrix ${}^O T_V$ represent the position and orientation of the puppet reference frame $\{V\}$ in the Cartesian space frame $\{O\}$:

$${}^O T_V = \begin{bmatrix} {}^O R_V & {}^O P_V \\ 0 & 1 \end{bmatrix} \quad (3.4.3)$$

where ${}^O R_V \in \mathbb{R}^{3 \times 3}$ is a 3×3 matrix of the orientation of frame $\{V\}$ with respect to frame $\{O\}$. ${}^O P_V = [x_V, y_V, z_V]$ is the position vector of the origin of frame $\{V\}$ with respect to frame $\{O\}$. A full representation of the configuration of a puppet can be written as $(\theta_1, \theta_2, \dots, \theta_F, {}^O T_V)$ because it includes the posture of all body parts in $(\theta_1, \theta_2, \dots, \theta_F)$ with respect to frame $\{V\}$ and ${}^O T_V$, the location and the orientation of the puppet body with respect to the Cartesian space $\{O\}$. For simplifying the puppet control, some measures have been taken to keep the torso part stationary in body movements. Hence, the transformation ${}^O T_V$ remains constant.

3. *The string space $\{s_i\}$* is represented by the lengths and suspending positions of the strings $s_1, s_2, \dots, s_k$, where k is the total number of the strings exerting tensions on the figure body. The lengths of the strings have linear relationships with the rotary angles of the motors. They are the control variables of the system that directly drive the puppet and produce body movements. The configuration of the figure body described in its positions and postures in space can be uniquely determined by the strings that are tight when the robotic puppet is under gravity balance.

The coordinate transformations between the three reference frames can not be performed only through kinematics because the configuration of the robotic puppet in an underconstrained structure is also determined by the gravitational force besides the geometrical relationships between the strings, the body parts and the supporting frame. Any

transformation between two reference frames must incorporate the equilibrium conditions.

The location of the figure body with respect to the supporting frame and the human is commonly represented with its coordinates along the X_0 –, Y_0 –, Z_0 –axes in the frame $\{O\}$. It is convenient and intuitive to represent the postures with joint angles $(\theta_1, \theta_2, \dots, \theta_F)$ in the joint space $\{\theta_i\}$ of the puppet figure for behavior expression tasks. The position and orientation of a body part Q obtained from forward kinematics with joint angles $\{\theta_i\}$ can be described as a function of the joint agents in frame $\{V\}$: ${}^V T_Q = g(\theta_1, \theta_2, \dots, \theta_j)$. Hence, the coordinate of the body part Q relative to the Cartesian frame $\{O\}$ can be obtained through the transformation:

$${}^O T_Q = {}^O T_V^V T_Q = {}^O T_V \cdot g(\theta_1, \theta_2, \dots, \theta_j) \quad (3.4.4)$$

Given a set of string lengths $(s_1, s_2, \dots, s_K)$, we can obtain joint angles $(\theta_1, \theta_2, \dots, \theta_j)$ through Equation (3.4.5). Then, the position of the body part with respect to the frame $\{O\}$ can be determined. Given a set of joint angles $(\theta_1, \theta_2, \dots, \theta_j)$, however, may not be able obtain the string lengths $(s_1, s_2, \dots, s_K)$ unless the constraint conditions in Equation (3.4.5) are satisfied.

3.4.3 General Formulation of Kinematics

The configuration of the puppet figure with n DOFs can be specified with n joint variables $(\theta_1, \theta_2, \dots, \theta_n)$. The control variables to the motion of the figure body are the lengths of k strings $(s_1, s_2, \dots, s_s)$. As the robotic puppet is an underconstrained mechanism, the number of strings is commonly less than the total DOFs of the suspended figure body, that is, $k < n$. To determine the configuration of the figure body represented with n variables

Figure 3.6: General structure of the body parts

$(\theta_1, \theta_2, \dots, \theta_n)$ based on the lengths of k strings $(s_1, s_2, \dots, s_k)$, we need n equations. That is,

$$\begin{aligned}
 f_1(\theta_1, \theta_2, \dots, \theta_n) &= s_1 \\
 f_2(\theta_1, \theta_2, \dots, \theta_n) &= s_2 \\
 &\vdots \\
 f_k(\theta_1, \theta_2, \dots, \theta_n) &= s_k \\
 g_{k+1}(\theta_1, \theta_2, \dots, \theta_n) &= 0 \\
 g_{k+2}(\theta_1, \theta_2, \dots, \theta_n) &= 0 \\
 &\vdots \\
 g_n(\theta_1, \theta_2, \dots, \theta_n) &= 0
 \end{aligned} \tag{3.4.5}$$

Following assumptions are made in kinematics calculation:

- The changes of the string lengths caused by their elasticity are ignored.
- The friction on the joints and pulleys can be ignored.
- The mass distributions of the linkages are equal.
- The shapes of the limbs are ignored. So they can be treated as lines with masses.

- The displacements of the string suspending points caused by the shape of the pulleys are ignored.
- The strings are regraded to be tied to the tips of the links.
- The movements of robotic puppets are restricted to low speed and quasi-statics motion.

The first k equations are determined by the geometrical constraints of the mechanical structure of the robotic puppets. Each equation can be obtained from a vector loop formed by the links and the respective strings.

$$\sum \vec{L}_i = 0 \quad (3.4.6)$$

where $\vec{L}_i$ is the vector of a string or link in the loop including the string s_i . In each vector loop, there is at least one string that has never been included in other loops.

The orientation of the vector of the link l_i is described with two angles θ_{i1} and θ'_i around Y and Z axes respectively:

$$\begin{aligned} \vec{l}_i &= Rot(Y, \theta'_i) \cdot Rot(Z, \theta'_i) \cdot l_i \\ &= \begin{bmatrix} l_i \cos \theta_i \cos \theta'_i \\ l_i \sin \theta'_i \\ -l_i \sin \theta_i \cos \theta'_i \end{bmatrix} \end{aligned}$$

The vector of the string s_i uses two temporary joint variables α_i and α'_i around Y and Z axes respectively:

$$\begin{aligned} \vec{s}_i &= Rot(Y, \alpha_i) \cdot Rot(Z, \alpha'_i) \cdot s_i \\ &= \begin{bmatrix} s_i \cos \alpha_i \cos \alpha'_i \\ s_i \sin \alpha'_i \\ -s_i \sin \alpha_i \cos \alpha'_i \end{bmatrix} \end{aligned}$$

And vector of the string s_{i+1} is:

$$\begin{aligned} \vec{s}_{i+1} &= Rot(Y, \alpha_{i+1}) \cdot Rot(Z, \alpha'_{i+1}) \cdot s_{i+1} \\ &= \begin{bmatrix} s_i \cos \alpha_{i+1} \cos \alpha'_{i+1} \\ s_i \sin \alpha'_{i+1} \\ -s_i \sin \alpha_{i+1} \cos \alpha'_{i+1} \end{bmatrix} \end{aligned}$$

The vector $\overrightarrow{C_{i+1}C_i}$ of the distance between two suspending points C_i, C_{i+1} whose positions are known constants is:

$$\overrightarrow{C_{i+1}C_i} = \begin{bmatrix} x_{ci} - x_{ci+1} \\ y_{ci} - y_{ci+1} \\ z_{ci} - z_{ci+1} \end{bmatrix}$$

The vector summation of the loop including the string s_i and the link l_i is:

$$\sum \vec{L}_i = \vec{l}_i - \vec{s}_i + \vec{s}_{i+1} + \overrightarrow{C_{i+1}C_i} \quad (3.4.7)$$

By replacing the vectors with their components, we get the a 3×1 matrix:

$$\begin{pmatrix} l_i \cos \theta_i \cos \theta'_i - s_i \cos \alpha_i \cos \alpha'_i + s_i \cos \alpha_{i+1} \cos \alpha'_{i+1} + x_{ci} - x_{ci+1} \\ l_i \sin \theta'_i - s_i \cos \alpha_i \cos \alpha'_i + s_i \cos \alpha_{i+1} \cos \alpha'_{i+1} + y_{ci} - y_{ci+1} \\ -l_i \sin \theta_i \cos \theta'_i + s_i \sin \alpha_i \cos \alpha'_i - s_i \sin \alpha_{i+1} \cos \alpha'_{i+1} + z_{ci} - z_{ci+1} \end{pmatrix} = 0 \quad (3.4.8)$$

The other $(n - k)$ equations can be obtained from static equilibrium or quasi-static conditions. The k strings are the control variables of the puppet who has definite configuration if it stays in equilibrium. When the figure body is in balance states of being static or moving slowly, the total forces $\vec{F}_i$ and total moments $\vec{M}_i$ on link l_i can be treated as zero:

$$\sum \vec{F}_i = 0 \quad (3.4.9)$$

$$\sum \vec{M}_i = 0 \quad (3.4.10)$$

These equations are valid in the quasi-static condition under which influences of rather small acceleration can be ignored.

The total forces on the link l_i is:

$$\sum \vec{F}_i = \vec{F}_{i1} + \vec{F}_{i2} + \vec{T}_i + \overrightarrow{T_{i+1}} \quad (3.4.11)$$

While

$$\vec{T}_i = \begin{bmatrix} -T_i \cos \alpha_i \cos \alpha'_i \\ -T_i \sin \alpha'_i \\ T_i \sin \alpha_i \cos \alpha'_i \end{bmatrix}$$

and

$$\vec{T}_{i+1} = \begin{bmatrix} -T_{i+1} \cos \alpha_{i+1} \cos \alpha'_{i+1} \\ -T_{i+1} \sin \alpha'_{i+1} \\ T_{i+1} \sin \alpha_{i+1} \cos \alpha'_{i+1} \end{bmatrix}$$

$$\vec{F}_{i1} = \begin{bmatrix} F_{xi1} \\ F_{yi1} \\ F_{zi1} \end{bmatrix}$$

$$\vec{F}_{i2} = \begin{bmatrix} F_{xi2} \\ F_{yi2} \\ F_{zi2} \end{bmatrix}$$

Following matrix is obtained:

$$\begin{pmatrix} F_{xi1} + F_{xi2} - T_{i+1} \cos \alpha_{i+1} \cos \alpha'_{i+1} - T_i \cos \alpha_i \cos \alpha'_i \\ F_{yi1} + F_{yi2} - T_{i+1} \sin \alpha'_{i+1} - T_i \sin \alpha'_i \\ F_{zi1} + F_{zi2} + T_{i+1} \sin \alpha_{i+1} \cos \alpha'_{i+1} + T_i \sin \alpha_i \cos \alpha'_i \end{pmatrix} = 0 \quad (3.4.12)$$

The summation of the moments $\vec{M}$ on the link $\vec{l}_i$:

$$\sum \vec{M}_i = \vec{l}_i \times (\vec{F}_{i2} + T_{i+1} \vec{T}_i + \vec{G}_i) \quad (3.4.13)$$

That is,

$$\begin{pmatrix} \sin \theta'_i (F_{zi2} T_{i+1} \sin \alpha_{i+1} \cos \alpha'_{i+1}) - \sin \theta_i \cos \theta'_i (F_{yi2} - T_{i+1} \sin \alpha'_{i+1} - m_i g) \\ \sin \theta_i \cos \theta'_i (F_{xi2} - T_{i+1} \cos \alpha_{i+1} \cos \alpha'_{i+1}) + \cos \theta_i \cos \theta'_i (F_{zi2} T_{i+1} \sin \alpha_{i+1} \cos \alpha'_{i+1}) \\ \cos \theta_i \cos \theta'_i (F_{xi2} - T_{i+1} \cos \alpha_{i+1} \cos \alpha'_{i+1}) - \sin \theta'_i (F_{yi2} - T_{i+1} \sin \alpha'_{i+1} - m_i g) \end{pmatrix} = 0 \quad (3.4.14)$$

From Equations (3.4.5), we can obtain the configuration of the puppet figure if the lengths of the strings are given. The direct kinematics of the robotic puppet is to determine the configuration of the figure body in terms of $(\theta_1, \theta_2, \dots, \theta_n)$ under equilibrium conditions when the lengths of the strings $(s_1, s_2, \dots, s_k)$ are given, while the inverse kinematics is to determine the lengths of the strings $(s_1, s_2, \dots, s_k)$ from the configuration of the figure body in space represented with $(\theta_1, \theta_2, \dots, \theta_n)$.

The results of solving the equations should be checked by the tensions T_i , $(i = 1, \dots, k)$ on the k strings because the robotic puppet is a wire-suspension mechanism, in which the strings can only exert tensions on the suspended object along the direction of retraction. This is a condition that must be satisfied in order to ensure the obtained results of the equations are valid in reality.

Figure 3.7: Small displacement on the torso caused by lateral forces

From the motion equations of one single arm, it is clear that kinematics properties including configuration, workspace and manipulability are dependent on the string length, position of string suspending point, position of string attaching points on links, masses of links, mass distribution on links, and joint types.

3.5 Simplification of Kinematics Calculation

The computation requires to solve Equations (3.4.5) for the kinematics of the robotic puppet is enormous because the constructed puppet figure has 30 DOFs which need 30 equations. In addition, it is difficult to assign the initial values to the large number of variables in the equations for numerical solutions. Thus, the kinematics calculation should be simplified while keeping the induced errors within an acceptable range. An intuitive method is to divide the whole body kinematics into multiple parts. This can be achieved by making the torso part of the figure body stationary when the limbs move.

The position of the torso part often leaves its balance due to the forces of the connected arms and legs (Figure 3.7). The force of an arm or a leg on the suspended torso part may cause the shoulder strings to deviate from the vertical direction a small angle β which is a

function of the force $\beta = f(\vec{F})$ exerted by the limb. The lengths of the shoulder strings s_1 and s_2 are assumed to be the same as s . The width and height of the torso are w and h respectively. From the equation of force balance on the torso $\vec{T} + \vec{F} + \vec{G} = 0$, where $\vec{T}$ is the total tensions on the strings, $\vec{F}$ is the total forces of the connected limbs, and $\vec{G}$ is the weight of the torso part, the deviation angle β can be obtained by:

$$\beta = \arctan\left(\frac{F_x}{G - F_y}\right) \quad (3.5.1)$$

where F_x is the total forces of the limbs along the x -axis direction and F_y along the y -axis. Though the relative motion of the arm and leg is a combination of rotation and translation, the motion of the arm and leg can be regarded as pure rotation with respect to the torso if the displacement of the torso can be ignored when the arm moves. By increasing the mass of the torso to a large value, the torso can maintain stationary relatively. Then the control of the robotic puppet can be simplified by independent control of each limb.

The mass of the torso part can be calculated using the maximum tension on the strings driving limbs in the workspace. The transitional displacement on the torso caused by stretching the arms or legs is measured with the relative displacement r defined as the square root of its two components along x - and y -axes:

$$r = \sqrt{r_x^2 + r_y^2} \quad (3.5.2)$$

where r_x and r_y are the ratios between the displacements and the lengths of the torso along the x - and y -axis direction.

$$r_x = \frac{s}{w} \sin\beta = \frac{sF_x}{w\sqrt{(mg - F_y)^2 + F_x^2}} \quad (3.5.3)$$

$$r_y = \frac{s}{h} (1 - \cos\beta) = \frac{s}{h} \left(1 - \frac{mg - F_y}{\sqrt{(mg - F_y)^2 + F_x^2}}\right) \quad (3.5.4)$$

To minimize the effect of the lateral forces on the torso, it is desirable to make the relative displacements along all directions as small as possible by shortening the string length and increasing the mass and size of the torso. For the case of one arm moving in the vertical plane, the relative displacement caused by the lateral forces can be restricted

Figure 3.8: One arm pulled by a string moving in a vertical plane

to a given range described with the ratio r by increasing the mass of the torso part to a value calculated with Equations (3.5.3) and (3.5.4).

Despite the benefit that motion control becomes easier by considering the torso a static object, the robotic puppet loses some expressive capability of body movements due to this simplification. The human has perception to the flow of movement that is exhibited by the order of body parts joining in motion. An unhampered free flow can be distinguished from a hampered bound flow [108]. The movements starting from the torso and then flowing out towards the extremities of the limbs are perceived to be freer when the limbs first begin to move than those in which the torso remains motionless. For example, a slashing movement originates in the torso and proceeds towards the shoulder, the upper arm, and the forearm. While pressing movement presents a bound flow that starts from the hand, progresses inwards to the arm, shoulder, and finally to the torso. The movement of the robotic puppet always starting from the ends of limbs and flowing to the torso can not produce a feel of free flow. In addition, the torso attends many movements in various amplitudes. When a person walks, accompanying the reciprocating movements of the legs, the center of the body moves up and down.

3.6 The Configuration of One Arm

Motion control of the robotic puppet necessitates the inverse kinematics relationship from the desired body postures to the required string lengths. This can be illustrated by a simple model involving only one arm pulled by one string (Figure 3.8). An arm or a leg of the figure body has two linked rigid bodies whose lengths are l_1 and l_2 respectively. The two links connected with the two one-DOF hinge joints J_1 and J_2 can move in the vertical plane when the length s of the string attached to the end of the second link changes. Their masses m_1 and m_2 are assumed to be equally distributed on the links. The influences of the shape of the links to the configuration are ignored. The positions of the string suspending point C and the fixed joint J_1 in the Cartesian space are given.

In this system, there are two DOFs on the arm and one active input, i.e., the length of the string. The configuration of the arm that is represented with the rotary angles of the two joints can not be determined with only kinematics because both the arm and the string form an underconstrained mechanism. There is a determined mapping between the configuration of the arm and the lengths of the string when the arm is in static equilibrium. Hence, the configurations of the arm with respect to different lengths of the string can be calculated according to statics condition.

The joint variables of the joint J_1 and joint J_2 are denoted as θ_1 and θ_2 respectively. Let J_3 be the position where the string attaches to the end of the second link, and the angle between the string and the direction of the x-axis is α . Using the standard vector loop method on the two links and the string, we can get a geometric relation:

$$\overrightarrow{J_1 J_2} + \overrightarrow{J_2 J_3} + \overrightarrow{J_3 C} + \overrightarrow{C J_1} = 0 \quad (3.6.1)$$

with $\overrightarrow{J_1 J_2} = [l_1 \cos \theta_1, l_1 \sin \theta_1]^T$, $\overrightarrow{J_2 J_3} = [l_2 \cos \theta_2, l_2 \sin \theta_2]^T$, $\overrightarrow{J_3 C} = [s \cos \alpha, s \sin \alpha]^T$, $\overrightarrow{C J_1} = [x_{J_1} - x_C, y_{J_1} - y_C]^T$. The position of the string suspending point and the joints are $C = (x_C, y_C)$ and $J_i = (x_{J_i}, y_{J_i})$, $i = 1, \dots, 3$.

From the freebody diagram of individual links of the arm (Figure 3.9), we have the

Figure 3.9: Freebody diagram of the individual links

force balance equations:

$$\vec{F}_1 + \vec{F}_{12} + \vec{G}_1 = 0 \quad (3.6.2)$$

$$\vec{F}_{21} + \vec{T} + \vec{G}_2 = 0$$

$$\vec{F}_{12} + \vec{F}_{21} = 0$$

and moment balance equations:

$$\overrightarrow{J_1 J_2} \times \vec{F}_{12} + 0.5 \overrightarrow{J_1 J_2} \times \vec{G}_1 = 0 \quad (3.6.3)$$

$$\overrightarrow{J_2 J_3} \times \vec{T} + 0.5 \overrightarrow{J_2 J_3} \times \vec{G}_2 = 0$$

where $\vec{T} = [-T \sin \alpha, T \cos \alpha]^T$ is the tension on the string. $\vec{G}_1 = [0, -m_1 g]^T$ and $\vec{G}_2 = [0, -m_2 g]^T$ are the gravitational forces on the links l_1 and l_2 .

By replacing the vectors in the motion Equations (3.6.1), (3.6.2) and (3.6.3) of the arm with their components, a 6×1 matrix of equations is obtained:

$$\begin{pmatrix} l_1 \cos \theta_1 + l_2 \cos \theta_2 + s \cos \alpha + x_{J1} - x_C \\ l_1 \sin \theta_1 + l_2 \sin \theta_2 + s \sin \alpha + y_{J1} - y_C \\ f_{1x} - T \sin \alpha \\ f_{1y} + T \cos \alpha - (m_1 + m_2)g \\ l_1 \sin \theta_1 f_{1x} - l_1 \cos \theta_1 f_{1y} + 0.5 l_1 \cos \theta_1 m_1 g \\ l_2 \sin \theta_2 f_{1x} - l_2 \cos \theta_2 f_{1y} + l_2 \cos \theta_2 (m_1 - m_2)g \end{pmatrix} = 0 \quad (3.6.4)$$

where $g = 9.81 \text{ m/s}^2$. And f_{1x} and f_{1y} are the magnitudes of the two components of the

force vector $\vec{F}_1 = [f_{1x}, f_{1y}]^T$ at the joint J_1 along the directions of the x-axis and y-axis. There are totally 6 unknowns $\{\theta_1, \theta_2, \alpha, T, f_{1x}, f_{1y}\}$ in the 6 equations within the matrix.

Configuration determined by string length

Combining the equations in the matrix (3.6.4), and eliminating the temporary variables $\{\alpha, T, f_{1x}, f_{1y}\}$, the configuration of the arm in terms of the joint variables (θ_1, θ_2) with respect to the string length s is implicitly described by following two equations that can be solved numerically with Newton-Ramphson method [109]:

$$(l_1 \cos \theta_1 + l_2 \cos \theta_2 + x_{J1} - x_C)^2 + (l_1 \sin \theta_1 + l_2 \sin \theta_2 + y_{J1} - y_C)^2 - s^2 = 0 \quad (3.6.5)$$

$$\begin{aligned} & (\cos^2 \theta_1 \sin \theta_2 m_1 l_1 + \cos \theta_1 \sin \theta_2 m_1 (x_{J1} - x_C) + 2 \cos^2 \theta_1 \sin \theta_2 m_2 l_1 \\ & + \cos \theta_1 \sin \theta_2 \cos \theta_2 m_2 l_2 + 2 \cos \theta_1 \sin \theta_2 m_2 (x_{J1} - x_C) \\ & - 2 \cos \theta_2 \sin \theta_1 \cos \theta_1 m_2 l_1 - \cos^2 \theta_2 \sin \theta_1 m_2 l_2 - \cos \theta_2 \sin \theta_1 m_2 (x_{J1} - x_C) \\ & - \sin \theta_1 \cos \theta_1 \cos \theta_2 m_1 l_1 - \cos \theta_1 \cos \theta_2 m_1 (y_{J1} - y_C) - \cos \theta_1 \cos \theta_2 m_2 (y_{J1} - y_C)) \\ & / (\cos \theta_1 \cos \theta_2 (m_1 + m_2) (l_1 \cos \theta_1 + l_2 \cos \theta_2 + (x_{J1} - x_C))) = 0 \end{aligned} \quad (3.6.6)$$

Once the configuration of the arm is determined by solving Equations (3.6.5) and (3.6.6), an equation for calculating the magnitude of the tension $\vec{T}$ can be obtained:

$$T = \frac{0.5 \cos \theta_2}{\sin(\alpha - \theta_2)} m_2 g \quad (3.6.7)$$

The same configuration of the single arm structure having two DOFs can be represented in either two joint angles θ_1 and θ_2 or the length of one string. To specify the configuration in the string space requires less information.

Example:

Using the puppet ROMS-I as an illustration (Appendix A), the length and mass of the upper arm of link l_1 are 0.069m and 0.005kg; the length and mass of the forearm of link l_2 are 0.07m and 0.006kg. The positions of the joint J_1 is (0.215, 0.304) in the global reference frame whose origin is at the bottom of the supporting frame of the robotic puppet. The

Figure 3.10: Movable suspending point pulls the links in 3D

Figure 3.11: One suspended arm viewed from the top

position of the suspending point C of the string is $(0.335, 0.46)$. When given the length of the string $s = 0.15\text{m}$, the configuration of the arm becomes $(\theta_1 = -38.75^\circ, \theta_2 = 76.67^\circ)$ which is obtained by solving Equations (3.6.5) and (3.6.6). The tension of the string is 0.0665N and the endpoint of the arm J_3 is at $(0.294, 0.316)$.

Configuration in 3D Space

By changing the string length s , the arm consisting of the links l_1 and l_2 can be moved up and down in a vertical plane. To give the puppet a larger collection of behaviors, the structure should be designed to allow the arm move in the 3D space. One of the solutions is to make the string suspending point movable on the horizontal plane (Figure 3.10). This design adds two translational DOFs into the mechanism. Suspended in the space (Figure 3.10), the arm moves when the length or the position of the suspending points of

the string changes. The position of the string suspending point in three dimensional space is $C = (x'_C, y'_C, z'_C)$, and the positions of the joints are $J_i = (x'_{J_i}, y'_{J_i}, z'_{J_i}), i = 1, \dots, 3$.

When in equilibrium, the links l_1, l_2 , the string s , and the joints J_1, J_2, J_3 stay in the same vertical plane **II** where the system has the minimum potential energy. The angle β between the vertical plane **II** and the direction of x-axis can be calculated by (Figure 3.11):

$$\beta = \arcsin\left(\frac{x'_C - x'_{J_1}}{z'_C - z'_{J_1}}\right) \quad (3.6.8)$$

The calculation to the positions of the joints J_2, J_3 in 3D space can solved by the Equations (3.6.5) and (3.6.6) according to the planar motion condition. The angles θ_1, θ_2 between the links l_1, l_2 and the horizontal plane are calculated by replacing the distances $x_C - x_{J_1}$ with:

$$x_C - x_{J_1} = \sqrt{(x'_{J_1} - x'_C)^2 + (y'_{J_1} - y'_C)^2} \quad (3.6.9)$$

and $y_{J_1} - y_C$ with:

$$y_{J_1} - y_C = y'_{J_1} - y'_C \quad (3.6.10)$$

in Equations (3.6.5) and (3.6.6) which become:

$$\begin{aligned} & (l_1 \cos \theta_1 + l_2 \cos \theta_2 - \sqrt{(x'_{J_1} - x'_C)^2 + (y'_{J_1} - y'_C)^2})^2 \\ & + (l_1 \sin \theta_1 + l_2 \sin \theta_2 + y'_{J_1} - y'_C)^2 - s^2 = 0 \end{aligned} \quad (3.6.11)$$

$$\begin{aligned} & (\cos^2 \theta_1 \sin \theta_2 m_1 l_1 - \cos \theta_1 \sin \theta_2 m_1 \sqrt{(x'_{J_1} - x'_C)^2 + (y'_{J_1} - y'_C)^2} \\ & + 2 \cos^2 \theta_1 \sin \theta_2 m_2 l_1 + \cos \theta_1 \sin \theta_2 \cos \theta_2 m_2 l_2 \\ & - 2 \cos \theta_1 \sin \theta_2 m_2 \sqrt{(x'_{J_1} - x'_C)^2 + (y'_{J_1} - y'_C)^2} - 2 \cos \theta_2 \sin \theta_1 \cos \theta_1 m_2 l_1 \\ & - \cos^2 \theta_2 \sin \theta_1 m_2 l_2 + \cos \theta_2 \sin \theta_1 m_2 \sqrt{(x'_{J_1} - x'_C)^2 + (y'_{J_1} - y'_C)^2} \\ & - \sin \theta_1 \cos \theta_1 \cos \theta_2 m_1 l_1 - \cos \theta_1 \cos \theta_2 m_1 (y'_{J_1} - y'_C) - \cos \theta_1 \cos \theta_2 m_2 (y'_{J_1} - y'_C)) \\ & / (\cos \theta_1 \cos \theta_2 (m_1 + m_2) (l_1 \cos \theta_1 + l_2 \cos \theta_2 + (x_{J_1} - x_C))) = 0 \end{aligned} \quad (3.6.12)$$

The position of the joint $J_2 = (x'_{J_2}, y'_{J_2}, z'_{J_2})$ in 3D space is:

$$\begin{aligned} x'_{J_2} &= x'_{J_1} + l_1 \cos \theta_1 \cos \beta \\ y'_{J_2} &= y'_{J_1} + l_1 \sin \theta_1 \\ z'_{J_2} &= z'_{J_1} + l_1 \cos \theta_1 \sin \beta \end{aligned} \quad (3.6.13)$$

And the position of the joint $J_3 = (x'_{J_3}, y'_{J_3}, z'_{J_3})$ is:

$$\begin{aligned} x'_{J_3} &= x'_{J_2} + l_2 \cos \theta_2 \cos \beta \\ y'_{J_3} &= y'_{J_2} + l_2 \sin \theta_2 \\ z'_{J_3} &= z'_{J_2} + l_2 \cos \theta_2 \sin \beta \end{aligned} \quad (3.6.14)$$

3.7 Workspace and Tension

Figure 3.12: The configurations of one arm for different string lengths. ($l_1 = 0.069m$, $m_1 = 0.005kg$, $l_2 = 0.07m$, $m_2 = 0.006kg$, $J_1 = (0.215, 0.304)$, $C = (0.335, 0.46)$, $s = 0.07m, 0.08m, \dots, 0.31m$)

The intended task of the robotic puppet is to convey information with its body movements. The puppet figure presents various postures in space when the lengths of the strings are changing. The workspace and tensions on the strings have influence to its expressive capability. We use the single arm example given in Section 3.4 to show the workspace and tension.

Workspace of One Arm

Figure 3.12 shows the different configurations that one arm of the robotic puppet ROMS-I may take when the length of the string varies within the valid range, which synergistically

Figure 3.13: The workspace of one arm. The workspace is represented in (a) the Cartesian space and (b) the joint space. ($l_1 = 0.069\text{m}$, $m_1 = 0.005\text{kg}$, $l_2 = 0.07\text{m}$, $m_2 = 0.006\text{kg}$, $J_1 = (0.215, 0.304)$, $C = (0.335, 0.46)$, $0.07\text{m} < s < 0.31\text{m}$)

Figure 3.14: The tension magnitude on the string in different lengths. ($l_1 = 0.069\text{m}$, $m_1 = 0.005\text{kg}$, $l_2 = 0.07\text{m}$, $m_2 = 0.006\text{kg}$, $J_1 = (0.215, 0.304)$, $C = (0.335, 0.46)$, $0.07\text{m} < s < 0.31\text{m}$)

determines the workspace of the body part with its structure. A large workspace which possibly offers a rich and diversified behavior repertoire is preferred. The workspace of the arm moving in the vertical plane is the set of all positions that the end of the arm can reach. Its workspace is a curve shown in Figure 3.13(a). Within the limited range of the string length, the workspace is continuous. The workspace of the robotic puppet is formed by the collection of the workspace of all body parts.

Body postures are intuitively to be described with joint angles of the puppet figure. However, as the joint angles are not the control variables of the system, the configurations specified by randomly assigned joint angle values may not valid. For examples, the arm in Figure 3.8 consisting of two links has two DOFs planar motion with the string length as the only system input. From the workspace described by two joint angles in Figure 3.13(b), we can see that the ranges of joint angles are highly restricted. The workspace described in either the Cartesian space or the joint space is a curve or 1-D manifold as there is only one input. If the assigned joint angles of the two links fail to be on the curve, the expected posture will never be reached.

Tension on String

Figure 3.14 shows the relation between the tension and the string length. A shorter string length on the arm corresponds to a larger tension, that is, the tension on the string increases when the arm is lifted up. At two places in the workplace, the tension on the string increases remarkably. One place is at the configuration when the end point of the arm is on the horizontal of the shoulders. Below this value, pulling the string increases the tension slowly though the configuration of the arm changes a lot. The other is at the configuration that the string is nearly parallel to the arm. The magnitude of the tension increase sharply when the arm is near this extreme configuration. The magnitude of tension must be restrained to avoid pulling the torso part away from its balance place. The configuration of the arm corresponding to the maximum magnitude of tension allowed forms the boundary of the workspace.

3.8 Summary

Our physical agent for expressive tasks is a motorized marionette designed according to underconstrained wire-suspension structure. The stringing method and the arrangement of the motors are designed to give the robotic puppet behavior diversity and ease of control. The kinematics that is processed as statics is practical for the motion control of the robotic puppet. The string lengths, masses of links, string attaching positions and other design parameters synergistically determine the configuration of the suspended puppet figure.

The estimation of the physical parameters of body parts may not be precise because the shapes of the body parts are irregular. The mass center of the torso is measured with the balance method. To make the torso stable when the arms and legs are engaged in movements, the weight of the torso is augmented by adding additional weight.

A list of assumptions and restrictions are made to simplify the kinematics calculation. Besides the simplifications on measuring the physical properties of the body parts, two major assumptions are the fixed torso and quasi-static state.

The elastic strings pulled by the gravitational forces on the body parts all the time can store some amount of potential energy. However, the elasticity of the strings made of cotton and nylon is very limited. The change of potential energy stored in the elongated strings cannot produce obvious differences on the string lengths during the motion of the puppets. To simplify the kinematics computation, the lengths of the strings are assumed to be fixed. Potential energy stored in the strings does not have remarkable influence to the motion of the puppets. The effects of the elasticity of the strings to the movements are then ignored.

The motion of the wire-suspended puppets is heavily affected by the dynamics of the mechanisms. Dynamic properties have more impact to the movement of under-constrained wire-suspension mechanisms than the fully constrained mechanisms. Except under static condition, its motion states at any moment have relations with its previous states. To determine its motion states, both the kinematics and dynamics are required. Without dynamics, the implementation of motion generation method on the puppets according to

statics highly restricts the variations of body movements and thus the expressiveness of behaviors. When in equilibrium, the configurations of the puppet figures having minimal potential can be determined according to statics.

No dynamics model was developed for the puppet mechanisms, because (1) The robotic puppets are used only as a platform to illustrate the design process which can also be done by other expressive physical agents, such as human-like or animal-like robots. (2) The size of the puppet is kept small. Due to the miniature size of the body parts, it is extremely difficult to mount sensors which are essential to measure the motion states [82] and perform the dynamics analysis. (3) Whether or not to calculate dynamics is also depends on the requirements of the expressive tasks, or the niche space. Expressive tasks of a particular category like sudden and jerky motion, fast movements of limbs require dynamic analysis. Since in this research such kind of motions are not demonstrated, the study of dynamic analysis was felt not very essential.

Chapter 4

Behavior Modules

This chapter resolves the problem of bringing movements to the robotic puppets, which require a rich collection of behaviors for the expression tasks. Our behavioral design method is based on a set of behavior modules and a multilevel behavior representation that organizes the behavior modules into both diversified and complex behaviors. The behavior modules developed from an agent-centered perspective form the building blocks of the behavior repertoire of the robotic puppets. The multilevel structure of behavior representation provides portability and flexibility to behavioral design for the physical agents with a wide range of mechanical structures. To produce coordinated movements on the many body parts, a biologically inspired method is used to constrain them. The large number of degrees of freedom causes two problems: computational complexity in specifying the configurations, the indeterminacy of possible configurations. The second aspect need not be discussed for the robotic puppets due to the assumption of fixed torso and rather simple structure of single limbs. For physical agents with other body structure, the indeterminacy may become unavoidable. The issue of indeterminacy is closely related with motor equivalence, one of the four major features of human movements.

4.1 Introduction

Producing movements on human-like physical agents is different from that of industrial robots or mobile robots. Their tasks of interacting with human give several requirements to behavioral design: abundant behavior patterns, human-like behaviors, personality in

Figure 4.1: The requirements and effects of the behavior generation methods

behaviors, and less workload in behavior input (Figure 4.1).

4.1.1 Requirements of Behavioral Design

To have large behavior encoding capability, the physical agents must have many DOFs which need to be constrained to produce desired movements. The physical agents have complex kinematics and dynamics in real world due to the large number of DOFs. When specifying the body movements, the user wishes the physical agents to have less DOFs. To be engaging, the body movements should show some general features of human behavior and unique features or personality of the physical agents. To reduce the time spent on designing

body movements from scratch, the behaviors should be made into modules and existing motion sequences should be utilized. The diversified appearances and body structures of the physical agents make behavior reuse important.

4.1.2 Behavior Generation Methods

We explore three behavior generation methods to produce life-like and abundant body movements according to the requirements of behavior design: primitives, multilevel motion synthesis, and motion transform. The primitives bring the unique features of the physical agents, but they are not human-like. The primitives are used as basic behavior modules in motion synthesis and motion transform. The method of motion transform utilizes existing motion sequences to produce human-like movements on the physical agents. Human features are incorporated into the motion sequences generated in the process of multilevel motion synthesis through the involvement of the user. The many DOFs are constrained to generate coordinated full-body movements.

4.1.3 Complexity in Behavior Generation

The large number of DOFs of human-like physical agents gives the designers tremendous freedom for behavioral design. People have control over the movements of the agents. However, the enormous possibilities in behavioral design make the control over the movements of the physical agents a difficult task requiring rich skills and labor. There should be a tradeoff between the complexity of behaviors and the control method.

An important work of behavioral design is to explore and find new movement patterns so that a behavioral repertoire can be set up for expression tasks. The number of DOFs of a human-like physical agent causes an exponential explosion in the number of possible actions that can be taken in every state. For example, the limbs of ROMS-I has eight spherical joints and in total 24 DOFs before it is suspended in space with the strings (Figure 1.3a). If each of the DOF was explicitly controlled at every instant in the movements, given each DOF had only three possible values, e.g., the maximum, the minimum, and the median,

there is a combination of $3^{24} > 10^{10}$ different postures that can be taken by the puppet. As it is impossible to set up a huge behavior repertoire including so many postures, it is necessary to use more compact behavior representation, or to acquire and maintain only those behaviors closely relevant to the intended tasks. Both of the methods are used in behavioral design for the robotic puppet. The first method that represents the behaviors through a reduced number of DOFs has been approached in the structural design. By taking the underconstrained structure, only nine independent strings are required to control the 30-DOF puppet ROMS-I. The number of the DOFs of the robotic puppet will be further reduced in the forms of motor primitives and the multilevel behavior structure. The second method that only keeps most useful behaviors is implemented through mimicking humans or human-like movements.

Several factors influence the determination of behavior generation methods for controlling the robotic puppet, such as the nature of prospective tasks, sensory information, and mechanical constraints.

4.1.4 Behavior Generation in Animals

Before starting to make the robotic puppet move, we first examine behavior generation mechanisms in the natural biological systems that may provide suggestions to movement control of artifacts. Two types of behavioral acquisition, selection and coordination processes happening on animals are discriminated by ethological study [110].

More or less encoded in heredity materials, innate behaviors happen broadly under the nature condition. Animals and human are born with certain specific behaviors which are hard-coded in their motor control mechanisms. Some innate behaviors are as simple as reflective behaviors, some are not so complex that they are composed of many behavior modules and use memory to store knowledge about the environment. An example of the innate behavior happening in sequence is the mating cycle of wasps in which each action is triggered by changes of internal states and events from the environment. The complete action sequence implements one important goal though the animal may not realize its

consequence for species survival. Some innate behaviors need customization based on the situation an animal lives in during its exploration in the environment. Their adaptability gets enhancement through setting up and keeping environmental model in memory for future use. Internal states and motivation stemming from survival conditions or more abstract goals of an agent may play a role as environmental stimulus in activating a behavior. With the aids of innate behaviors, animals can associate environmental stimuli with their motor commands.

The other way to acquiring behaviors is through learning. The learning ability is complementary to innate behaviors that alone can not cover all the possibilities of the complex environment. In mammals and especially primates, the babies must spend a great deal of time learning from experience. Sometimes even common behaviors necessary for survival, such as the hunting skills of lions, are learned in practice.

4.1.5 Inferences from Ethology

The insights from ethology suggest that there will be two methods for robot makers to generate behaviors: through preprogrammed behaviors, learning from others. And simple behavior modules can be sequenced to create complex behaviors. One problem is how the preprogrammed or innate behaviors on the robots are produced. The common methods to provide the robots with preprogrammed behaviors are either directly assigned by human or abstracted from human movements. Contrast to these methods, our method of producing innate behaviors on the robots is to discover them from the specific mechanical structures. All the special configurations such as those corresponding to the singularity, boundary, and manipulability can be used as posture primitives. They correspond to the discontinuous parts in the entire workspace. Commonly the kinematics of the robots is nonlinear. Therefore, the postures or movements can not be superimposed directly. Through this method, the workspace, i.e., the range of all valid postures is divided into some smaller regions. Within each small region, the kinematics is roughly continuous. This method works well on the robotic puppet whose workspace is rather continuously. It may not be effective for

the robots with very complex structures because there are a large number of singularities in their workspace.

4.1.6 Inferences from Force Field Experiments

In the experiments with spinal frogs [89], a certain region of the spinal cord was stimulated with microelectrodes and the forces at the ipsilateral ankle were registered. The position of the limb was systematically changed while the same location was stimulated. The registered forces were directed to a single point within the limb reach. All of the forces formed a field that converged at the equilibrium point. When the limb was at the equilibrium point, the electrostimulation did not elicit any force. Stimulation of various locations generated different fields with different zero-force points. Two simultaneous stimulations to two spinal regions elicited a vector field that was proportional to the vector sum of the fields produced by the independent stimulation of each region. These experiments suggest that the limb postures are represented in the form of convergent force fields acting on the limb's end point. The described finding suggests that the central nervous system represents movements and postures in the vector form which is different from the methods that have been used in robot kinematics.

The force field experiments on frogs and rats revealed the origin, representation, and organization of the primitives.

1. Primitives can be represented internally by animals as a force field with single equilibrium point.
2. Their organization principles are linear supposition and inhibition.
3. It can be inferred that the force fields must be caused by the interaction of kinematics of body parts and biomechanical properties of muscles whose viscoelastic characters are tuned to specific states by neural circuit units. The force field is the representing format of postures.

We postulate that the formation of the primitives would make the workspace which

the animal can reach as large as possible through combining as small a number as possible of primitives. The force fields must be the result of optimization on the biomechanics and kinematics of body parts through evolution. So when primitives are created for robot control, they should root in physical properties of the robot. This is in contrast to the methods of deriving primitives from human motion data [111].

It must be the result of long term natural evolution, the formation of primitives, with only four primitives to cover the entire task space, the neural circuits adapted to viscoelastic property of muscles and kinematics of leg structure. They found the best postures and most efficient computation.

Though the formation of the primitives should be attributed to the interaction between neural circuits and the musculoskeletal structure of the body parts, it is reasonable to believe that more adaptation is made by the former to fit the later but not vice versa because it is much easier for neural circuits to make changes. This means the musculoskeletal structure is the major constraint to motion performance. Hence, the design of primitives for robots should be founded on the decisiveness of their specific mechanical structure.

4.1.7 Benefits of Using Primitives

Motor primitives are biological structures that organize the underlying mechanisms of complete movements, including spinal fields and central pattern generators. Encoded in neural circuits activating on parts of the musculoskeletal structure, they can generate the stereotypical behaviors. In a computational sense, primitives can be viewed as a basic set of motor programs that are sufficient, through combination operators, for generating entire movement repertoires. The mechanics of motor apparatus covered by the primitives is relevant not only to control, but also to computation and representation of motor behaviors. A motor primitive has the following characteristics:

- Modularity: a primitive encapsulates kinematics, dynamics of body parts and control method applied on them. Information is hidden from upper layers. Complex behaviors may be formed by combining primitives.

- Objectivity: a primitive achieves a specific task.
- Plasticity: a primitive may be tuned to fit for a specific task through learning and experience.
- Extensibility: the set of primitives may be extended to include new primitives learned from experiences.

The essential feature of primitives is their modularity with what complex movements may be synthesized. The modularity of primitives simplifies the behavior organization in the physical agents that have to perform a great deal of computation to determine the trajectories, velocities, torques, etc. Incorporating primitives into a hierarchical structure eliminates some computation and delay among the large number of movement units. The primitive concept brings out several benefits for behavior generation and control:

- The modular feature encompassing kinematics and control makes the control architecture simple.
- Constructing large behavior repertoire from a small set of reusable modules and avoiding inverse kinematics calculation for each configuration.
- Dividing the full body into parts.
- Simple interactions to high-level motor command center.
- Using less sensory information, implementing rapid response of reflex behaviors.
- A robot controller designed this way might offer stability during interactions with different environments [112].

The inverse kinematics of primitives, corresponding to the specific postures, are calculated beforehand and kept in the memory as a mapping for the robot puppet from the joint space to the string space. Once an expected posture is assigned as a vector of rotary joint angles in the joint space, the robot tries to get the assigned angle vector through combining

primitives linearly or nonlinearly, i.e., to interpolate in the workspace and directly obtain solution in the string space without solving inverse kinematics.

The common method of motion synthesis is editing the movements directly, such as in the keyframe and choreography [113] methods that specify body movements in the joint spaces of the animated characters. They are not effective to the robotic puppet with an underconstrained suspension structure. The postures specified in the joint spaces of the robotic puppet may not be reached no matter how to pull the strings. The primitives encapsulate the inverse kinematics and provide the user a set of behavior modules to construct complex behaviors.

An idealized set of primitives should have a minimal number of primitives but they are enough to form the whole behavior repertoire. The primitives should satisfy:

1. They are independent to each other; each primitive can not be represented with others.
2. The number of primitives is equal to the number of DOFs of the system under investigation.

However, most physical agents are nonlinear systems, and it is impossible to find such a set of primitives. Thus, we have to use a larger number of primitives to produce more behaviors.

For a system having n DOFs, a primitive is defined in the joint space as a vector $P = (\theta_{10}, \theta_{20} \dots \theta_{n0})^T$, while θ_{n0} is the equilibrium position or the extreme position of the n^{th} DOF. A primitive is stable at the vicinity of its equilibrium position. It is convenient to use the extreme positions to represent the primitives which correspond to the boundary or singularities of the workspace. The areas between extreme positions are commonly regarded to be continuous and stable.

4.2 Primitives Created by Human

There are two ways proposed by previous researches to provide robots with primitives: 1) hand-coded intuitively, 2) abstracted and learned from demonstration. The primitives manually assigned by the human reflect the prospective of people about what kind of primitives the robot should have. The quality of primitive selection depends on the experience of the person. The primitives designated by human may not be suitable for the body structures of the robots. This method is not efficient for the complex physical agents whose behaviors are depicted with many control variables. Imitative learning is a major method to transfer captured human motion data to humanoid robots. The assumption implicitly taken by the learning method is that a primitive works well on the human will also work on robots. Less attention has been paid to the preliminary condition that transferring primitives abstracted from human motion data to humanoid robots requires enough similarity on kinematics and dynamics between them. However, there are often significant differences on the numbers of DOFs, body sizes, materials and so forth between the human and humanoid robots, and among different robots. Primitives that work well on the human may not be feasible or appropriate for the robots whose kinematics and dynamics properties have obvious differences from those of humans. Most robots, whether anthropomorphic or not, are composed of motors and linkages do not have the springy properties of muscles.

Inspired by the natural innate behaviors, which are the result of co-evolution of both behaviors and musculoskeletal structures, we believe primitives should be tightly coupled with the underlying structure of the robots. So the decisive effects of the mechanical structure on primitive design for robots should not be ignored. As the first step in motion generation, we should discover primitives from robot's mechanical properties though it is often impossible to find out all primitives along with this method due to the complexity and inherent nonlinearity of the mechanical structure. However, the primitives obtained by this way correspond to the robot's body structure and may be regarded as its innate behaviors.

This discovery process looking for primitives representing innate behaviors makes a

Figure 4.2: Three special postures of the left arm

division to the behavior space which is formed by the collection of behaviors available in the workspace of the robotic puppet. The reduced behavior space can save the work of other behavioral design methods such as the method of learning from demonstration.

4.3 Motor Capability Discovery

Different from the hand-code and imitation methods, our primitive design method takes an agent-centered perspective. It depends on the discovery of motor capability of the physical agent. The mechanical structure of the robotic puppet provides information about the puppet's capability to generate movements for expressive tasks. The motor capability determined by its body structure constrains the range of behaviors and tasks the physical agents can perform. Though the other factor relevant to motor capability is the control method, which has impact on the performance of body movements, we only consider motor capability provided by the mechanical structure which is the fundamental element. The primitives having these characteristics are most likely at the extreme positions within the boundary of the workspace. This kind of positions that reflect the unique features of a mechanical structure can be obtained by checking the special regions in the workspace, such as the singularity and manipulability of the limbs. A direct method to determine the primitives for a single limb of the robotic puppet is to use the extreme postures of lifting up and falling down the limb. A transition posture between the two can be also selected as a primitive. Three primitives working on the right arm of the puppet ROMS-II are

shown in Figure 4.2. Though this is the simplest condition of motor capability discovery, a large amount of movements can still be found due to the structural complexity of the puppet body. For a puppet figure with each limb having only three primitives or innate postures, there are $3^4 = 81$ postures can be formed in total. If the postures mirroring about the sagittal plane are regarded as the same postures, there are still 36 distinct postures. Through exploring its motor capability, the robotic puppet can automatically create a rather big number of primitives enough for many expressive tasks.

These primitives calculated from the mechanical structure of the limb are closely associated with characteristics of the body part. These behaviors have no relationships with human behaviors. They can be regarded as the "innate" behaviors of the robot puppet. In behavioral expression they can deliver messages reflecting the unique features of the robotic puppet, such as its personality and identity. Whatever the robot puppet acquires and expresses human-like behaviors, the "innate" behaviors will show some consistency in its behaviors.

4.4 Movement Synthesis

Puppeteers spend many years to study and explore puppet's behaviors. They discover the movements that a puppet is able to perform through extensive trial-and-error guided by their experiences and intuition. This process of looking for available behaviors is equal to the use of heuristics and rule-based methods to search the behavior space of the puppets. Nevertheless, the puppeteer's method can not be applied to the robotic puppet because of our lack of the experiences of the puppeteers. And it may be difficult to transform puppeteer's experiences into the operational format to control robotic mechanisms. The overwhelming computation of using brute force search to build a behavior repertoire is formidable.

Inspired by the fact that a distributed hierarchical structure in human behavior organization would reduce control complexity, we use a multi-layer behavior structure to address the computation complexity problem.

Human Behavior Organization

The behaviors of the human are generated in a distributive multilevel structure. Human's motor control systems are formed by a close integration of the peripheral motor system and the central motor system [114]. The peripheral part consists of muscles and nerve fibers while the central part includes several motion-related subsystems. The production of movement from the action plan to the detailed motor behaviors is the result of the interaction among cerebral cortex, basal ganglia, cerebellum, brain stem, and spinal cord [16].

Movement synthesis on the physical agent with complex body structure takes a multilevel approach that processes the temporal and spatial aspects in motion representation and control separately.

Separation of the Spatial and Temporal Aspects

As the movement possesses both spatial and temporal characteristics, a thorough representation of any movement must include its instantaneous configuration in space and the evolving process in time. The spatial aspect refers to the space occupied by the body parts involved in a movement. The temporal aspect includes the timing of movements, velocity, acceleration, and rhythms of motion sequences.

Viewing from the process of movements, the spatial and temporal aspects are indivisibly blended and mutually constrained. For example, in the task that let a robot arm draw a square, the temporal characteristics that exhibit as speed and acceleration of the robot arm have apparent influence to the spatial aspect of the movement. If the speed and acceleration are not precisely controlled, the robot arm will bias the expected path at the corners. However, biological evidences show the temporal and spatial aspects of behavior are separable [115].

The spatial and temporal aspects of a movement can be separated and controlled as two individual processes. Time-independent properties and parameters representing spatial aspects of the movement can be determined first and then combined with temporal constraints at the time when the movement is executed.

Figure 4.3: Multilevel motion synthesis

4.4.1 Multilevel Movement Representation

The spatial characters in the behavior of the robotic puppet lead to full body movement in space while the movement is proceeding along the time line. When independently considering the spatial aspect, the whole body movements of the robotic puppet can be decomposed into behavior modules acting on the kinematically coupled body parts. The synthesis process of spatial motion implemented on the robotic puppet has five layers (Figure 4.3). The movements are synthesized bottom-up, and performed top-down on the robotic puppet.

The behaviors of the robotic puppet are presented in this layered structure with each layer corresponding to a specific level of abstraction. The motion synthesis uses a bottom-up method. Lower level behavior units provide services to higher level movements as their building blocks when movement instructions descending from higher levels are executed. The first layer implements the interface between the underlying mechanical structure and the supported high level movements. In human movement control, basic motor units possess an internal model of the biomechanical structure of body parts. The feedforward effect of the internal model reduces the delay of feedback loop and amount of communication between layers of the hierarchical control structure [74]. Similarly, the behavior modules in the first layer keep the kinematics models of the associated body parts of the puppet figure. They execute inverse kinematics calculation and make coordinate transformation

between the three reference frames.

The second layer is represented with the discrete and static postures that the body parts take. The postures obtained from examining the motor capability of corresponding body parts are the "innate" postures reflecting the mechanical properties of the robotic puppet. More postures for expression may be assigned by the human with any method. The third layer implements point-to-point discrete movements and continuous oscillation movements. The fourth layer coordinates movement modules on different body parts into harmonious whole body movement. The fifth layer generates the movement of emotional expression. There is no limitation to the methods of creating behavior modules in each layer.

In addition to simplifying the control architecture of the robotic puppet, the multilevel structure of the behavior representation provides flexibility for designers to take different engineering methods of motor control. At certain degree, the behavioral design process is independent from the underlying mechanical structure. This gives behavioral design considerable portability which is a salient problem for expressive physical agents made into a highly diversified body structure.

4.4.2 Types of Primitives

Complex movements are the outcomes of flexible combination of motor primitives when they are instantiated for task execution. A primitive macroscopically showing modular feature must include an internal model of the associated mechanical structure in order to implement local non-feedback control, control variables of motion, parametric representation of movement states, and sometimes transformation between relevant coordinates.

There is no standard way to create primitives on the physical agents. Any control method acting on body parts that is compatible to the intention of producing complex behaviors through combinational operation may be used in representing and constituting motor primitives. Schema [71] have been used to form primitives on mobile robots, impedance control [73] [90], force fields [73], and central pattern generators [92] [116] are

used to control humanoid robots and physically simulated characters. Our layered primitive structure is similar to schema in concept that complex behaviors include simple behaviors but it is more complex and flexible than previous implementations.

4.4.3 Posture Primitives

The second level in behavior structure is formed by a list of posture primitives. The postures recruited as primitives are extreme and comfortable postures for manipulation, manually set postures, or learned postures. It is convenient to directly represent postures with joint angles in the joint space of the robot puppet. Given a posture in a set of joint angles, the required string lengths with which the motors can drive the puppet to the desired posture can be calculated by Equations (3.6.5) and (3.6.6) discussed in the previous chapter. The postures must satisfy the prerequisites of the equations that the torso part is assumed to be stationary so that the inverse kinematics problem can be solved by dividing the whole body into smaller parts. The posture of a puppet figure having n DOFs can be represented with n joint variables $(\theta_1, \theta_2, \dots, \theta_n)$. The corresponding string lengths that make the robotic puppet take the posture are $(s_1, s_2, \dots, s_k)$ where k is the number of the strings. The posture primitive of the robotic puppet with n DOFs suspended with k strings is defined as a vector consisting of n joint angles and the corresponding lengths of k strings:

$$P = (\theta_1, \theta_2, \dots, \theta_n, s_1, s_2, \dots, s_k)^T \quad (4.4.1)$$

Commonly, only part of the entire body is involved in a movement or the movements on some body parts are of interest to the observer. Thus, the representation of a posture primitive can include only the relevant movement on body parts and ignore the irrelevant ones. A posture primitive can also be described with a subset of the total n' DOFs and k' strings:

$$P = (\theta_1, \theta_2, \dots, \theta_{n'}, s_1, s_2, \dots, s_{k'})^T \quad (4.4.2)$$

where $n' \leq n$ and $k' \leq k$.

The links and strings controlled by the primitives vary according to the tasks. The most fundamental primitives include one string and two links of one arm or leg. A posture

primitive covering part of body that kinematically couples several connected links and joints forms a basic behavior module.

Though it is convenient to represent body expression tasks as postures in the joint space, the joint space of the robotic puppet is not continuous. Many postures that can be described in terms of two or more joint variables can not be reached due to specific statics of the wire-suspended structure.

4.4.4 Discrete Movements

The movement of humans can be categorized into discrete movements and continuous movements [117]. Arm reaching movement and taking postures are examples of the discrete movements. The discrete movements are set up on the posture primitives. A discrete movement is represented with a transition $DM = (P_s, P_d, t)$ from the starting posture P_s to the destination posture P_d within a duration t . Via points obtained through interpolation between the starting and destination postures are used to smooth the motion.

4.4.5 Rhythmical Movements

One common type of continuous movements is the rhythmical movements or oscillations, such as walking, swimming, flying, and waving hands (Figure 4.4). As one of the basic types of human and animals' activities, rhythmical movements in the natural environment are assumed to be generated by central pattern generators, which is a kind of neural circuits continuously producing signals for rhythmical movement patterns [16].

Rhythmical movements on robots can be implemented by simulating the central pattern generator with artificial neural networks or differential equations that can generate cyclic but not convergent movements. The movement generated in this way is similar to human rhythmic movements that oscillate around an equilibrium point. This method is coincident to the essential characteristics of human movement understood by dynamic theory of motor control. Some postures obtained from the motor capability discovery can be used as the

Figure 4.4: The arm moves periodically

Figure 4.5: Rhythmic movement of one arm. The movement is measured by the angle between the arm endpoint and horizontal position.

equilibrium position of the rhythmical movement. It is also possible to use wavelet augmented with noise to simulate the human rhythmic movement. To simplify the calculation, we use a periodic function represented in the joint space of the puppet to generate rhythmic movements. As the extremity of arm is of interest, the rhythmic movement is generated at the end of the arm (Figure 4.5). The endpoint of the arm moves along the curve which is the workspace of the arm. The angle between the position of the arm extremity and the x -axis direction around the fixed joint J_1 is denoted as θ_3 . Then the trajectory of periodical movement can be generated with:

$$\theta_3 = a \sin(\omega t + \phi) + \theta_0 \quad (4.4.3)$$

where θ_3 is the angular displacement of the end point of the arm with respect to a reference position θ_0 , a is the magnitude of oscillating movement, ϕ is the starting phase of the oscillation in a period of ω .

Along the time line of the periodic movement, each transient posture of the arm is treated as a low level posture primitive and the required string length is calculated through inverse kinematics. From the geometric relationship between θ_3 and θ_1, θ_2 ,

$$\tan \theta_3 = \frac{l_1 \sin \theta_1 + l_2 \sin \theta_2}{l_1 \cos \theta_1 + l_2 \cos \theta_2} \quad (4.4.4)$$

we get

$$l_1 \sin \theta_3 \cos \theta_1 - l_1 \cos \theta_3 \sin \theta_1 + l_2 \sin \theta_3 \cos \theta_2 - l_2 \cos \theta_3 \sin \theta_2 = 0 \quad (4.4.5)$$

Given θ_3 , the string length can be obtained by solving the Equations (3.6.5) (3.6.6), and (4.4.5) numerically.

Figure 4.5 shows that the arm is making a periodic movement around the horizontal direction in a range of 30° by periodically changing the length of the string. It can be seen from Figure 4.6(a) that the angular displacement θ_3 selected for describing the periodic movement of the end point of the arm has approximately linear relationship with the string length. So the inverse kinematic calculation for determining the string length in order to produce the periodic movement on the arm needs not be performed on each small step.

Figure 4.6: The periodical movement of one arm. The endpoint of the arm oscillates when the string length changes periodically. (a) The range of the string length in a periodic movement. (b) The string length changes periodical in order to produce periodic movement on the arm. (c),(d) The trajectories of the arm in the Cartesian space. ($l_1 = 0.069\text{m}$, $m_1 = 0.005\text{kg}$, $l_2 = 0.07\text{m}$, $m_2 = 0.006\text{kg}$, $J_1 = (0.215, 0.304)$, $C = (0.335, 0.46)$, $a = 30^\circ$, $\omega = 2\pi$, $\theta_0 = 0$, $t = 3$)

Figure 4.6(B) shows the change of the string length in three periodic arm movements. The trajectories of the arm represented in the Cartesian space of the three periodic movements are shown in Figure 4.6(c) and Figure 4.6(d).

4.4.6 Coordination

Both the human and the robotic puppet have many DOFs that need to be controlled in order to generate coordinated movements. The degrees-of-freedom problem requests a method to organize the individual DOFs and represents body movements in abstraction. The coordination is to master the many DOFs in a movement by ways of reducing the number

of independent variables to be controlled. Humans have magic capability to coordinate a large number of DOFs in the form of joints and muscles. The large number of potential DOFs precludes the possibility that each DOF is controlled individually at every point in time. Bernstein [16] hypothesized that the human construct a single coordinative structure in order to produce an entire class of movements. Coordinative structures are assembled temporarily and flexibly so that a single component at the micro-level may participate in many different coordinative structures for different tasks or conditions. Conversely, a single coordinative structure may require the use of different micro-components at different times. Following the assembly of the coordinative structure, a further constraint process must come into play to produce a particular movement.

To generate harmonious full-body movements on the robotic puppet, some constraints must be applied to its DOFs in order to combine individual DOF into larger behavior modules. In one behavior module, the DOFs are not controlled individually but are treated as a whole. They are constrained in a relatively fixed pattern during the movement. To put the joints and links together into a collective unit allows the whole body to be controlled as if it had fewer DOFs. For example, the symmetry in body structure can reduce the information required to specify the movements.

There are three general ways working on different levels of the systematic organization to implement movement coordination among multiple DOFs. Coordination of body parts in a system can be achieved through structure design, kinematic calculation, and sensory feedback. The use of all the three methods for coordinating body parts can be found in human movements [16].

Through carefully organized and configured mechanical structure, some DOFs in a mechanism can be constrained to form individual behavior modules. A mechanism with many couplings has the weakness of lacking flexibility and can only be used for very limited tasks.

Sensory information provides constraints between the body parts. Depending on the properties of sensory information, constraints may take the forms of position, force, etc. Swarm behavior of the robot team and object operation with multiple manipulators are

examples of realizing coordination by taking advantage of the sensory information.

Coordination can be accomplished by kinematic calculation when the kinematic model of the mechanism is known. Constraints between different parts form one more layer of kinematic relationships in the space. During a movement, the individual DOFs are temporarily constrained to form fixed units. The kinematic relations between body parts are task-specific. This method is practical to the robotic puppet as the global posture can provide some constraint information for coordinating the primitives.

4.4.7 Full Body Movement

Each primitive acting on one part of the puppet body constrains a small set of links and strings. For example, the arm waving movement covers one string and two links of a single arm. These behavior modules setting up on a decomposed figure body can not generate coordinated full body movements unless the distributed structure is organized systematically. Coordination requires adding constraints on the body structure in order to associate different movements and body parts.

There are several theories about coordination of human behavior, such as hierarchical control [118] and self-organization [119]. Our coordination method takes the functional perspective to Cohen's theory [113] [120] about movements involving multiple DOFs spreading the full body. By comparing and analyzing animals with different evolutionary histories, it points out that full body movements performable for a human or the animal are constrained by the body morphology. For example, a starfish with radial symmetry body structure can move by contracting and extending its limbs. Frogs control their bodies by dividing the control to body into the upper and lower parts. Horses are more highly evolved, so they can control their bodies like frogs when they run very fast. The associated body parts are constrained as a specific behavior module. Besides this behavior, horses can control the left and right body respectively in order to move the left forelimb and the left hind leg simultaneously. When a horse trots, the left part and right part are crossly controlled. In this movement, the right forelimb and the left hind leg can move together, the left forelimb

Figure 4.7: The limbs move towards the center of the body

and the right hind leg then move together. Highly evolved animals have more whole body behavior modules available to constrain the multiple DOFs than low-level animals. So their full body movements have more diversity and flexibility.

Full body movements on the robotic puppet are synthesized by coordinating the body movements through constraining the primitives. Several primitives applied to different body parts can be grouped into a behavior module. A synergist is a set of strings or motors that work together to produce a given movement. Synergists are dynamically formed in motion sequences to implement coordination among body parts. This forms the forth layer of the behavior structure of the robotic puppet that synchronizes the full body movements. The behavior modules of the full body movements are represented in groups of the constrained primitives.

There are five behavior modules that constrain the full body movements in concert. These behavior modules are dynamically formed through the connections across body parts and can be combined to form complex body movements. The details required to describe body movements are reduced when they are represented on the higher level.

1. The *radial* module makes all limbs to contract towards or to extend away from the center of the body (Figure 4.7). The four limbs accept the same movement command specifying one limb's motion at the same time. Each limb is composed of two links pulled by only one string. Hence, in the radial movement, the eight links are constrained into one synergist.
2. The *waving* module generates swaying movement on the body through driving each

Figure 4.8: One arm proceeds the other

Figure 4.9: The independent movements of the upper and lower body parts

Figure 4.10: The arm and the leg on the same side move together

Figure 4.11: The arm and the leg on the opposite side move together

limb rhythmically (Figure 4.8). The motion instruction for a single limb is repeatedly applied to each limb under synchronization of a periodic function.

3. The *segment* module divides the body into the upper part and the lower part. The two arms of the upper body part are constrained into one single movement unit and the two legs of the lower part are treated identically (Figure 4.9). Movement instructions specifying the motion of the arm and the leg are required to achieve the full body movement.
4. The *ipsilateral* module constrains the left arm and the left leg as one movement unit, and the right arm and the right leg as the other unit (Figure 4.10). Two movement instructions respectively specifying the motion of a left limb and a right limb are required.
5. The *cross* module constrains the left arm and the right leg as one movement unit, and the right arm and the left leg as the other unit (Figure 4.11). One movement instruction specifying the movement of a limb is sufficient to generate the full body movement.

The symmetry in the body structure and movement reduces the amount of information required to specify the motion states of the DOFs spreading the entire body.

4.5 Summary

One way to provide the robotic puppet with expressive behaviors is through motion synthesis. Inspired by the modular features of human motor primitives, our motion synthesis is based on a collection of movement units of postural primitives. Previously the primitives designed for humanoid robots or other physical agents are given by the human. We take an agent-centered method in behavioral design that discovers the extreme and special configurations from the mechanical structure of the robotic puppet and use them as the initial set of primitives in the behavior repertoire. The method of motor capability discovery uses

the specific body structure as the source in behavior generation. The process of behavioral design and representation is preformed in a multilevel structure in order to give the behaviors the properties of portability and flexibility, which are important for expressive physical agents whose body structures are highly diversified.

The primitives are conceptually generated based on the information exhibited by the underlying mechanical structures. In implementing the primitives on the robotic puppets, the configurations having special features, such as the boundary of the workspace, are selected as the primitives. The primitives generated this way are posture primitives. The workspace of one single arm of the robotic puppets which has only two degrees of freedom is approximately continuous. When the number of degrees of freedom on the body parts increases, the workspace quickly becomes so complex that the method of looking for special configurations in the workspace may not be able to work.

Chapter 5

Expressive Behaviors

The human-like appearance of the robotic puppet evokes people's expectations to human-like behaviors. Body movements with human traits are necessary for the expression task of the robotic puppet not only because they produce illusion of life, but also because many fundamental movements are exclusively used by the human, such as walking. The body movements of the human have distinctive patterns due to many factors, such as the physical properties of body structures, psychological states, and cultural background. The behaviors made up from motor primitives of the robotic puppet exhibit specific properties of the mechanical structure. They do not have the features of human behaviors, and they are not human-like. There are a lot of human-like movements that have been produced in human motion capture, simulation and animation. They can be used as plentiful resources to implement human-like movements on the robotic puppet. Our method to give human-like movements to the robotic puppet is to adapt existing human motion sequences through imitative learning.

5.1 The Correspondence Problem

The major problem that must be solved in imitative learning is the correspondence problem that the behaviors of the target model must map those of the source model. This problem becomes particularly obvious for expressive physical agents that have various appearances and body structures. There may be remarkable differences on the number of DOFs, sizes of body parts, and materials among the physical agents. Even for the robotic puppet,

Table 5.1: Types of imitative learning as the combinations of goals, actions, and states

Goal	Action	Result	Type of social learning
Yes	Yes	Yes	Imitation
Yes	Yes	No	Failed imitation
Yes	No	Yes	Goal emulation
Yes	No	No	Goal Emulation
No	Yes	Yes	Mimicry
No	Yes	No	Mimicry
No	No	Yes	Emulation
No	No	No	No social learning

changes may be made to the number of strings, stringing positions, and body structure of the puppet figure. To have a similar embodiment with the human is a special case within all possible situations. The imitative performance of the robotic puppet may have good result only if the correspondence problem is considered. Nehaniv [18] defines the correspondence problem as relational mapping in studying imitation learning of autonomous agents:

Given an observed behavior of the model, parse this behavior into action sequences $t_1, \dots, t_k$, starting in state y_0 with states $y_1, \dots, y_k$, the states that successively result from carrying out the t_i in order. Find a correspondence Φ on states and Ψ on action sequences, relating one's own actions and states to those of the observed model, and from a corresponding state c_0 , execute corresponding behaviors $b_1, \dots, b_k$ such that the resulting behavior is as similar as possible, where the degree of dissimilarity is measured by summing an error measure $d((y_i, t_i), (c_i, b_i))$ over i .

Three components in the correspondence problem can be identified: the possible states of the two models, sequences of actions that result in changes from the starting states to the end states, and measures of dissimilarity in matching behaviors. The prerequisite of exact behavioral mapping, the identical body structures, the identical starting states, and the identical motion sequences will never be satisfied in human movement imitation. A useful correspondence could be a partial mapping. People must propose a criterion to evaluate the behavioral mapping. The degree of dissimilarity can be studied using various metrics and measures of correspondence.

Categorization of Imitative Learning

When studying the behaviors of the physical agents with some degree of autonomy, the states of the models involved in the imitative learning can be further distinguished into the common state and the goal state. The goal is defined as the configuration of the states that meet an external or internal criterion given by the designer. The state is an instantaneous representation to the system and the environment. An action and a sequence of actions are behaviors that transform the states of the physical agents. All the combinations of these three factors define the different conditions of imitative learning (Table 5.1). Our intended motion transform from human motion data to the robotic puppet belongs to mimicry, only transferring the movement patterns of the human to the robotic puppet.

Correspondence of Expression Behaviors

The correspondence of expression behaviors happens in three levels. The first level is the correspondence of body parts. The left and right arms of the robotic puppet maps to the left and right arms of a human. The second level is the correspondence of actions. A pointing movement on the left arm of the robotic puppet maps to the pointing movement on the left arm of the human. The spatial and temporal aspects of the movements of the robotic puppet maps to the spatial and temporal aspects of the movements of the human. The third level is the correspondence of the meanings of the body movements which depend on the interpretation of the human observers. This level of correspondence is left to the user's discretion.

5.2 Mimicry

A human can control complex movements through flexible combination of motor primitives with some learning ability. From the fact that the imitative behaviors occur in the babies at only one hour of age [121], it is reasonable to postulate that their imitative behaviors could not be acquired by experience but were a result of an inborn ability, due to the sensorimotor connections in the brain. The existence of the mirror neurons that combine the perceptual and motor primitives also suggests that there are prepared behavior modules to facilitate

Figure 5.1: Imitative learning through composing primitives

imitative learning. This means that the imitation of movement patterns is not achieved by repeated training, but through combining existing motor primitives. Each primitive is an element of computation in the behavior repertoire that transforms desired limb trajectories into motor commands. Theoretical studies [122] have shown that a system's ability to learn depends on the formation of its primitives.

Following the imitative learning mechanisms of humans and animals, we implement motion transform from simulated human motion data into motion sequences on the robotic puppet through imitative learning. The behavior modules in the behavior repertoire will compete with each other in composing the given motion data with the largest correspondence. The motion of the source model is mapped onto a set of existing primitives of the target model. Subsequently, the most appropriate primitives are adjusted by learning to improve the performance. If there is no existing primitive matching the given movement, a new primitive is created and inserted into the behavior repertoire. This method of using primitives in imitative learning is closely related to the notion of mirror neurons.

Sometimes the motion sequences on the robotic poppet are not continuous because of several reasons, such as the user skips the intermediate movements, original motion sequences before transformation have sudden motion changes. To make the motion proceeds

smoothly, intermediate movements can be inserted in the motion sequences through interpolation. Another way similar to the interpolation is to use existing primitives in the behavior repertoire.

5.3 Motion Transform through Imitative Learning

The movement transfer includes two parts of work: one is to acquire new primitives from the displayed motion sequences, the other is to combine existing primitives to form a motion sequence as close as possible to the original one according to a given measure. The method of imitative learning does not attempt to make the robotic puppet, the target model, to truly learn the behaviors of the source model, but mimicry the behaviors on surface. As the imitative learning is not sensitive to the remarkable differences on the body structures of the source model and the target model, we use it to make the motion transform. The source model may be a human, simulated human, or even animated characters with exaggerated behaviors.

Most human-like physical agents have similar branching structures. The movements represented in the joint space of the source and the target models are more similar than in the Cartesian space. Therefore, movement representations should be first transformed into the joint space before making motion transformation.

When the user inputs a motion sequence for motion transformation on the robotic puppet, it is his task to map the corresponding body parts between the source model and the robotic puppet. At the beginning of the motion transform, the behavior repertoire has an initial set of primitives obtained through motor capability discovery or direct user setting. They may correspond to the extreme positions, middle positions, or other special positions in the workspace.

During the transformation, postures that can not be represented with existing primitives will be inserted into the behavior repertoire as new primitives. As humans depend on some small areas in the complete workspace of body movements in behavior expression, this method is effective to obtain human features in the motion sequences.

The algorithm of the motion transformation through learning is:

1. Given a motion sequence of the source model described in a discrete format $M_s = \{M_1, M_2, \dots, M_i, \dots\}$, where M_i is the posture of the source model at time i
2. Given a behavior repertoire $B = \{P_1, P_2, \dots, P_j, \dots\}$ of the target model where P_j is a postural primitive
3. Given an empty motion sequence $M_t = \{S_1, S_2, \dots, S_i, \dots\}$ of the target model
4. For each M_i in M_s
 - (a) For each P_j in B
 - i. Compute spatial distance $D_{j,i} = |P_j - M_i|$
 - ii. If spatial distance $D_{j,i} < D_c$, (the given spatial distance constant), let $S_i = P_j$ and return
 - (b) Get the primitives P_a and P_b that are closest and second closest to M_i respectively
 - (c) If M_i is not in $[P_a, P_b]$, (the range between P_a and P_b), then calculate the lengths $S_a = \{s_1, s_2, \dots, s_k\}$ of the k strings corresponding to M_i through extrapolation on $[P_a, P_b]$
 - (d) Else calculate the lengths $S_a = \{s_1, s_2, \dots, s_k\}$ of the k strings corresponding to M_i through interpolation on $[P_a, P_b]$
 - (e) Calculate the lengths $S_b = \{s_1, s_2, \dots, s_k\}$ of the k strings corresponding to M_i through inverse kinematics
 - (f) If the difference $|S_b - S_a| > s_c$, the given tolerance of interpolation error, then store (M_i, S_b) into B as a new primitive
 - (g) Let $S_i = S_b$ and return

When the motion transformation begins, the given motion sequence M_s of the source model are represented with a sequence of postures $\{M_1, M_2, \dots\}$ at time $t = 1, 2, \dots$. The postures are described in the joint space of the source model.

The robotic puppet system keeps a behavior repertoire B in which a list of posture primitives $\{P_1, P_2, \dots\}$ are stored. Defined in Equations (4.4.4) and (4.4.2), each posture primitive in $\{P_1, P_2, \dots\}$ represents a posture in both the joint space and the string space.

The empty motion sequence M_t is used to store the results of motion transformation. Each item in the sequence $\{S_1, S_2, \dots\}$ is a configuration of the string space of the robotic puppet. The spatial distance measure D defined by the user has obvious effects on the motion transform.

If the spatial difference defined by the user between the two values of the string lengths calculated through interpolation and inverse kinematics for moving the robotic puppet to the required direction is smaller than the user-assigned tolerance, that posture can be represented linearly with the existing primitives. Otherwise, the posture is inserted into the behavior repertoire as a new primitive.

The kinematics of the body part of the robotic puppet ROMS-I and ROMS-II has a little linearity. The higher the nonlinearity, the more primitives are needed to be recruited into the behavior repertoire. The tolerance of string lengths set by the user has influence on the number of primitives. Increasing tolerance level may cause greater difference on movement transformation, but reduces the number of primitives.

The algorithm makes inverse kinematics calculation in each step in order to obtain the string lengths $\{s_1, s_2, \dots, s_k\}$ that drive the puppet to the posture M_i of the source model. It can then be used to compare with the values of the string lengths obtained through interpolation between the neighboring primitives P_a and P_b in the behavior repertoire. The computation spent on inverse kinematics can be saved if the relationship between the string lengths and the postures is approximately linear.

5.3.1 Motion Transform on the Robotic Puppet

The robotic puppet ROMS-II keeps a behavior repertoire formed by the postural primitives and movement primitives. There are twelve postural primitives acting on the four limbs, and seven movement primitives in it. The postural primitives are created through motor capability discovery. Each primitive encapsulates the angles representing the posture of the associated body part and the corresponding string length. The postural primitives are stored as a list of items in the behavior repertoire. The postures of lifting the arm to the highest position and dropping the arm down are two primitives of ROMS-II. These

Figure 5.2: Puppet ROMS-II follows the simulated human model

postures are at the boundary of the workspace of the arm consisting of two links and one string. The middle position between these two extreme postures is also a little special. The tension begins to increase when the string drives the arm passing through this position. So this posture is also recruited as a primitive into the behavior repertoire and each arm has three posture primitives working on it. There are also three posture primitives associated with each leg. One primitive is controlled by the knee string and one by the ankle string pulling the leg backwards. The standing posture is at the boundary of these two primitives and used as the third primitive on the leg. The seven motion primitives are set up through multilevel motion synthesis. They are not used in motion transform because they are based on the postural primitives, but used to constrain the many DOFs in composing and coordinating complex movements.

Based on the primitives, the imitation learning method can transform motion data from various source models to the robotic puppet. Besides the motion synthesis method, the robotic puppet can have walking movement through transforming the motion data of a physically simulated human figure modelled in ADAMS 12 (Figure 5.2). As the body structure of the simulated model is different from that of the puppet figure, representing the movements and correspondence in the joint space is appropriate. The angle γ between the legs and the vertical direction is simply used to measure the spatial distance $D = |P_j - M_i|$,

Figure 5.3: Walking movement transformed from the simulated character

where the P_j is the angle of the j^{th} primitive associated with the leg of the puppet, M_i takes the value of the angle γ of the source model at time i .

The string length s_i of the postural primitive S_i in the target motion sequence M_t can be calculated with the linear interpolation:

$$s_i = \frac{s_{j+1} - s_j}{\gamma_{j+1} - \gamma_j} \gamma - \frac{s_{j+1}\gamma_j - s_j\gamma_{j+1}}{\gamma_{j+1} - \gamma_j} \quad (5.3.1)$$

while s_j is the string length of the j^{th} posture primitive P_j in the behavior repertoire, and γ_i is the angle specified by the primitive P_j . The string length can be calculated with the angle γ of the given posture M_i of the source model falls between the j^{th} primitive and $(j+1)^{th}$ primitives of the target model. There are three posture primitives working on each leg when the motion transform begins. The result of the motion transformation through imitative learning based on the primitives is shown in Figure 5.3.

The generated movement of the puppet is not a perfect mapping to the source movement of the simulated character. The motion transformation from the simulated character to the robotic puppet relates to the correspondence problem. There are obvious differences on the body structures between the simulated character and the robotic puppet. The postures represented in joint angles must be transformed into string lengths to drive the motors. Because this transformation is not strictly linear, some differences generated between the

Figure 5.4: Expressive gestures. (a) Metaphoric gestures, (b) Emblem, (c) Iconic gesture, (d) Deictic gesture.

movements of them. Even though, the generated motion sequence of the puppet could be regarded as a walking behavior. Other factors that may have influences to the differences between their postures include the friction on the strings, the clothes, and the ground.

5.4 Body Movements in Communication

Humans are inborn masters of utilizing and perceiving body movements as a media. Human body movements have huge capability for encoding messages in non-verbal communication. Krout [4] observed 7,777 distinct gestures in schools and 5,000 hand gestures in hospitals. It is estimated that people may take as many as 1,000 steady body postures. If considering the fact that a person often displays several behaviors at any instance during a communication, and the specific meanings of the behaviors are determined by the context, the potential of expressiveness of human body movements is enormous. The robotic puppet is much simpler than a human, but its behavior diversity provide its with a large behavior encoding potential.

For the construction of physically embodied agents, there are types of gestures and body movements that can serve as key roles. In natural human conversation, both body movements and gestures add redundancy when the speech is in a noisy environment. This gives the listener cues about where in the conversation one is, and adds information that is not conveyed by the accompanying speech. For this reason, body movements and gestures can profitably work together in expressive physical agents.

A person may not have perception to ongoing communication through body movements. Like all forms of nonverbal communication, body movements have both intentional and unintentional forms which occur simultaneously and are largely independent of each other. The unintentional spontaneous communication refers to a sender's nonvoluntary display of inner emotional states and a receiver's direct and immediate sensory awareness of those states [123]. For the robotic puppet, we can implement spontaneous communication through modelling its internal states, such as emotion and intention. Whereas spontaneous communication occurs as a conversation of natural gestures, symbolic communication involves the use of arbitrary symbols, socially defined and intended to convey specific messages. While the language is the clearest example of symbolic communication, many nonverbal cues are also used symbolically. Symbolic gestures represent some widely recognized conventionalized meanings with body configurations and movements (Figure 5.4). For example, a thumbs-up gesture indicates agreement.

5.4.1 Emblems

Emblems are a class of gestures that we often use intentionally. These gestures are culturally specified in the sense that one single gesture may differ in interpretation from culture to culture. But in a few cultures these gestures can constitute more than ten percent of the gestures produced by humans [22]. Despite the paucity of emblematic gestures in everyday communication, it is appropriate to be implemented with the robotic puppet. They are widely used in interface design with the intention of creating gestural interfaces that can replace language in human-computer interaction. The emblems have a clearly defined meaning in the situations. However, the small number of different emblematic gestures is

not enough to form a gestural language. And people don't make plenty use of this type of body movements in communication.

5.4.2 Propositional Gestures

Another conscious gesture that has been the object of the study in the interface community is the propositional gesture. It indicates the position of the referred object in space. An example is the use of the hands to measure the size of a symbolic space. Another example is pointing to an object when referring to it. These gestures are not unwitting and in that sense not spontaneous. These gestures can be particularly important in certain types of task-orientated talk. Gestures such as these are found notably in communicative situations where the physical world in which the conversation is taking place. These gestures do not make up the majority of gestures found in spontaneous conversation. They receive the a lot of attention because they are conscious, witting gestures available to our self awareness.

5.4.3 Spontaneous Gestures

The vast majority of gestures are the gestural vehicles for our communicative intent with other human. They are unconscious and unwitting. These gestures are not available to conscious access by the person who produces them. We appear to lose access to the form of gestures, even though we have perception to the information that they convey. There are four types of spontaneous gestures.

1. *Iconic gesture* are some pictorial gestures intuitively representing physical entities in the world. It depicts some features of the action or the event by the form of the gesture. Iconic gesture may specify the viewpoint from which an action is taken.
2. *Metaphoric gestures* are also representational, but the concept they represent has no physical form but abstract objects or concepts. An example is the gesture that a person waves his fist to indicate the fighting.
3. *Deictic gestures* are those that point objects in the environment with arms and figures. They may also be used to indicate unseen, abstract or imaginary things.

4. *Beat gestures* are small baton-like movements that do not change in form during the expression. They have the meaning of emphasis or taking the attention of others.

5.4.4 Clothes

It has been assumed that there were no friction exerted by the joints, strings, and clothes when the inverse kinematics problem of the puppets was formulated based on the body structure of ROMS-I. The puppet bodies of ROMS-II and ROMS-III were designed to have clothes. In the experiment on ROMS-II, practical movements were roughly close to the intended postures. The accuracy of its motion was lower than that of ROMS-I due to the friction of the clothes. The clothes had slight influence to most movements of the puppet. ROMS-II commonly restricted to limited motion ranges. The major benefit of the friction is that it made the motion of the puppet less mechanical. For some special postures, such as those need to bend the limbs, the hard limits of clothing on ROMS-II are observable.

The clothes and hat are another form of nonverbal communication. They lend a little of personality of the robotic puppet and bias the context of performance. They assist in the perception to the front side of the puppet figure.

5.5 Summary

Character animation and simulation provides us plentiful resources of human-like movements which are necessary for the expressive tasks of the robotic puppet. Imitative learning that only implements surface mimicry is used to perform the transformation of body movements from the simulated character to the robotic puppet whose movements are highly constrained by its existing behavior modules. The closeness of the transformed body movements is compared with the original ones mainly from the spatial aspect because accurate timing on the movements of the physical agents may depend on the dynamic properties of them.

Chapter 6

System Implementation

The robotic puppets are avatars of humans. When a user wants to send information to others, he can specify the motion sequences of a robotic puppet. Then the information is conveyed to the human observers through its body movements. The robotic puppets act as the representations of the users in this one-way communication. ROMS-I is the first implementation of a robotic puppeteer system focusing on structural design. ROMS-II has apparent improvement on the arrangement of the strings and motors over ROMS-I. The stringing method is realized through the platform on the top of the frame for mounting the motors. The motor platform has four layers to arrange many motors and change the positions of six motors. A puppeteer software provides the user three behavior generation methods: motor capability discovery, multilevel motion synthesis, and motion transformation. ROMS-III has a similar mechanical structure and motion control software to those of ROMS-II.

This chapter uses the implementation of the robotic puppet ROMS-II to illustrate the system components of realizing expressive physical agents in the order of mechanical structure, electronic design, and software. The aim of these efforts is to make the robotic puppet have life-like body movements. The effectiveness of behavior expression will be demonstrated by ROMS-II with a short puppet show.

Figure 6.1: The system overview of the robotic puppet ROMS-II

6.1 System Overview

The robotic puppet ROMS-II consists of an articulated puppet figure, a frame on which many motors mounted, many strings suspending the puppet figure, a behavior generation and control system (Figure 6.1). The robotic puppet is controlled by a person who inputs and composes the body movements. Controlling the robotic puppet is similar to choreography of the movements for a performer on a stage. The user specifies the movements explicitly and then sets an emotional state that the robotic puppet uses to make its movements life-like. It accepts human's input and drives the puppet figure to move. The implementation of the puppet mechanism with robotic technology makes the control of the puppet simple. The user needs not to have the skills of a puppeteer. To cope with the condition that the mechanical structure of the robotic puppet may vary on the number of the strings and the body structure, the motor controller board must have extensibility in order to adapt to the change on the number of motors.

6.2 Puppeteer Mechanism

The robotic puppets ROMS-II and ROMS-III were designed on the experience of ROMS-I (Figure 1.3). The motors of ROMS-I are mounted on the beams of the frame and the

suspending points of the strings are fixed. The servo motors can only rotate within a range of 180° . There are several improvements on the structure of ROMS-II. The platform to mount the motors of ROMS-II has four layers that can accommodate more motors than ROMS-I. There is no limitation to the rotation of the motors which use pulleys to retract the strings. Thus, ROMS-II can display a wide assortment of body movements for human-machine communicative purpose. The strings are the direct source of driving the puppet figure move. The structural design of the robot puppet arrange the strings in appropriate positions while the behavioral design determines how to drive the strings.

6.2.1 Basic Design Concepts and Considerations

The aim of the structural design is to give the puppet a wide range of body movements and an easy way to control it. From the analysis to the kinematics of the robotic puppet, we can see that the stringing method constrains the configurations or postures that the puppet figure can take. The strings are the direct source driving the puppet figure. In addition, as the connection between the motors and the puppet figure, the strings and the stringing method have remarkable influence to the structural design of the supporting frame where the motors are mounted. To arrange the strings in the most appropriate and advantageous positions and orientations for puppet motion control, the supporting frame of the robotic puppet must have a well-designed platform to fix the motors, which give the power to the strings, to the right positions. Thus, the positioning of the strings and motors on the platform is the most important factor in the structural design of the robot puppet. Other considerations in the robot puppet design include motor selection, the sizes of the pulleys used on the motors, the shape of the motor platform, the making of the puppet figure that is suitable for a motorized marionette, and the restriction of movement due to the clothes worn by the puppet.

6.2.2 Use of Pulleys and Motors

The purpose of using pulleys is to provide string retracting capabilities to the servo motors. The main consideration is the pulley diameter once the motor positions have been

Figure 6.2: The motor platform

determined. In order to use less rotation on the motors to produce a wide range of motion on the puppet, it is preferable to have a larger diameter. A large diameter of the pulleys will facilitate the retracting of the strings. However, the size of the pulley is restricted by the space on the platform. The diameter of the pulleys is 50 mm.

The motors selected to drive the robotic puppet is Parallax 900 series continuous rotation RC servo motors. These motors accept speed control performed through continuous pulse-width-modulation (PWM) signals in a frequency of 50 Hz. The puppet figure of ROMS-II is suspended with 14 strings that are manipulated by 16 servo motors. Within the total 16 motors, ten motors are used to pull the strings attached to the body parts, one motor pulls the couple formed by two arms to produce clap movement, three motors control the positions of the torso and the head, and two motors are used to rotate the two small platforms in the four-layer platform. There are six motors on the two small platforms that make the suspending points of the six strings movable on the horizontal plane. So the control variables to the system are consisted of both the lengths and suspending points of the strings. Within the 14 strings, ten strings are attached to the limbs to produce

Figure 6.3: Sideview of the motor platform having four layers to accommodate the motors various movements. Two attached to the shoulders to support the puppet figure and move it up and down. The other two strings are attached to the head and two hands for special postures. The users can increase or reduce the number of strings in accordance with the requirements of behavioral expression.

6.2.3 Structure of Motor Platform

The conventional puppets are controlled manually through the strings attached on the control bar and manipulated by the puppeteers. We use the motors to drive the strings, so the structure of the platform on which the motors are fixed has a remarkable impact to the arrangement of the strings in space and the way to pull the puppet figure. Our platform to support the motors is put on the top of the robotic puppet. Due to the positional dimension of the strings and the size of the selected servo motors, a single layered platform is insufficient to provide enough space to mount the motors. Thus, it is designed into four layers in order to accommodate the many motors (Figure 6.2 and 6.3).

Figure 6.4: The network build on I2C Bus

6.3 Hardware

The electronics of the robotic puppet includes two parts: the circuit to drive the motor and the computer to calculate the body movements. An assembly of multiple PICmicro microcontrollers are responsible to produce the PWM signals to drive the servo motors.

Most computation related to behavior generation is performed on a Linux workstation. The multiple servo motors are directly controlled by a motor controller board. The behavior control hardware of the robotic puppet consists of:

1. One motor controller board with 17 PICmicro PIC16F876 microcontrollers.
2. Redhat Linux 6.1 running on the PC with a 200 MHz Pentium CPU.

Networked Motor Controller

The robotic puppet has 14 strings driven by 16 motors in current implementation, the motor controller board can be catered for future expansion with more strings and motors. The robotic puppet and other physical agents are likely to be designed into a variety of body structures. The string binding method and the number of strings may vary from puppet to puppet with different expressive performances. In order to do so, we have developed a networked motor controller that allows more motors to be accommodated easily.

The networked motor control circuit board is composed of 17 PICmicro 16F876 microcontrollers communicated through a common I2C bus (Figure 6.4). Another one connects the motor controller board with the PC. Each microcontroller is dedicated to position and velocity control of a particular servomotor. The microcontroller transforms motor control instructions coming from the PC into corresponding PWM signals to drive the servomotor. The I2C bus is an industrial standard for inter-board level communication between up to 124 microcontrollers. It has a simple structure that uses two wires to implement duplex byte communication. Each microcontroller connected to the I2C bus is given a unique address by which the messages delivered from the PC onto the I2C bus can locate the correct receiver. When the robotic puppet employs more DOFs, new motors and the corresponding microcontrollers can be added to the motor network. The networked motor controller can also include heterogeneous devices compatible with the I2C bus protocol, such as various sensors. This networked controller structure is also suitable for the implementation of concurrent and distributed low-level behaviors.

The microcontrollers are working under a frequency of 8 MHz produced by one crystal oscillator. The average speed of the communication between two microcontrollers on the I2C bus reaches about 80K bits per second though the interrupt service programs for timing and inter-chip communication in each microcontroller may cause some delay.

Serial Port Communication to the Linux Workstation

One of the 17 microcontrollers on the networked motor controller connects the board to the Linux PC workstation through the RS-232 two-wire serial communication. The default baud rate of the RS-232 serial communication is 9600 bits per second. Motor control commands descended from the embedded PC through the serial port at an actual speed of 70 bytes per second are executed by the microcontrollers.

Firmware Developmental Environment

Programmes running in the microcontrollers are written in assembly language proprietary to this class of chips. The firmware development tools include the MPLAB IDE integrated

Figure 6.5: Functional structure of movement generation and expression

development environment software and the PICSTART Plus programmer from Microchip Corporation. MPLAB IDE is a windows-style software used for programming and debugging the PICmicro microcontroller applications. It compiles application programs into a kind of HEX format files. The PICSTART Plus programmer reads these HEX files and burns the program into the 16F876 microcontrollers.

6.4 Puppeteer Software

The software developed for the robotic puppet performs the function of one-way behavioral communication with the observers. It provides an interface to accept motion control instructions from the users. The users have three ways described in the previous chapters to input and edit the movements for the robot puppet: primitives, motion synthesis, and motion transformation (Figure 6.5). These different motion generation methods give the users some options to varied workload and movement quality in order to make the motion

input process comfortable and effective. The users can manually edit the behavior modules in the behavior repertoire, combine the primitives to produce motion sequences, or drive the puppet with existing motion sequences from other sources through motion transformation. An initial set of primitives created through motor capability discovery is stored in the behavior repertoire for motion synthesis and motion transformation. The major modules of the puppeteer software include:

1. The graphic user interface for primitive setting and motion editing.
2. The internal structural model of the robotic puppet.
3. Motor capability discovery.
4. Layered representation of behaviors.
5. Motion transformation.
6. Emotional model.
7. Motion output and motor control.
8. Motor speed calibration.

Except for the motor control module, most parts of the software were developed in C language and compiled with GCC 2.2 on a PC under the Redhat 6.1 Linux operation system. The graphic user interface is implemented with X/Motif 2.1.

6.4.1 Motor Control Module

The motor control module written with the assembly language for 12 bit PICmicro microcontrollers runs in the microcontrollers in the networked motor controller. The interrupt resources of the microcontrollers used by this motor control module include the serial synchronous port, Timer 2, EEPROM, and I2C bus collision detection. It implements:

1. Transforming the motion sequences produced by the behavior generation modules into executable motor control instructions (Figure 6.6).

Figure 6.6: Motor control module executes motor commands

Figure 6.7: Configuration of USART transceiver in PIC microcontrollers for RS-232 communication.

Figure 6.8: I2C bus communication among the microcontrollers

2. Configuring and setting up RS-232 communication between the Linux workstation and the networked servo motor controller (Figure 6.7).
3. Sending the motor control instructions to the motor control board through the serial port.
4. Setting up duplex I2C bus communication between the microcontrollers on the networked motor controller (Figure 6.8).
5. Distributing motor control instructions among the motor microcontroller on the I2C bus.
6. Storing motor data and states.
7. Timing and generating PWM signals for speed control of the servo motors.
8. Error detecting and recovery.

Within the 17 PICmicro chips on the controller board, 16 microcontrollers control the 16 motors, and one microcontroller, called a common microcontroller, acts as the interface between the Linux workstation and other networked microcontrollers on the I2C bus. The program runs on this chip is different from those in the motor control microcontroller. The common microcontroller acts as a duplex transceiver. The program in the common microcontroller communicates with the PC through the RS-232 interface and with other onboard PICmicro microcontrollers through the I2C bus. The program inside the common microcontroller receives the motor control messages from the behavior generation program in the PC and delivers them to each microcontroller that uses the received messages to generate PWM motor control signals. One part of the motor control module running on the Linux workstation sends the motor control commands to the networked motor controller through the RS-232 interface.

6.4.2 Internal Model of Body Structure

An internal model of the body structure of the robotic puppet is kept as a component of the puppeteer software for kinematics calculation by the Equations (3.6.5) and (3.6.3). The

entire body of the puppet figure is divided into four limbs and treated individually. The module of motor capability discovery obtains the special postures from the body structure. In the behavior repertoire, there are several behavior modules that can act on the same body part. In a motion sequence, behavior modules are associated with the body parts to produce body movements. If a behavior module has been bound with a body part, other behavior modules can not use the same body part before that behavior finishes.

Coordinate Transformation

There are three coordinate frames in representing configuration of the robotic puppet. They are the joint space, Cartesian space and string space. Coordinate transformations between them are implemented through the Equations (3.4.3) and (3.4.4).

Data structure

The data structures defined in this program are listed in Appendix B. They include the descriptions to the physical properties and motion states of the motors, strings, joints, body parts, and coupled body structures (See Appendix B). The primitives, motion sequences, motion states of the puppet figure, and temporal information also have their data structures that store these variables during the body movements.

Representation of Body Structure

The robotic puppet is an articulated figure composed of multiple links between which the connections are loosely tied by the strings. The joints at the elbows and knees are modelled as one-DOF hinge joints and those at shoulders are three-DOF spherical joints. The body structure model of the robotic puppet represents a skeletal structure which allows proper limb movement under string tension and the gravitational force. The data structure storing the body structure in terms of its link lengths, masses, joint angles, topology, and string lengths maps its configuration into global coordinates (Appendix B).

The torso of the robotic puppet is selected as the root of the branching structure. The origin of the local frame is set at the center of mass of the torso and is used to specify the position of the figure in space. This convenient selection of the root link simplifies the

Figure 6.9: Five basic postures

coordinate transformation of the ends of limbs from the joint space to the global Cartesian space because the open loop chains from the origin of local coordinate to the ends of limbs are short. This data structure is able to represent many articulated figures having branching structure and to be applied to anthropomorphic or animal-like puppet figures with varied number of strings.

Ranges of Motion

The body movements must be guaranteed not to exceed the motion ranges allowed by the robotic puppet. The limits to the motion ranges come from two aspects: the extremities of its mechanical structure specified in terms of string lengths in the string space and joint angles in the joint space; the rationale of the puppet's postures in the human's eyes. The motion ranges constrained by the mechanics of the robot puppet are recorded in the body structure model. When the user sets the primitives manually, edits a behavior module, performs a motion transformation operation, and sends a motion sequence to execution, the motion ranges of the specified movements are compared with the records in the body structure model. The program does not have knowledge about the rationality of the behaviors which depends on the user.

Motor Capability Discovery

The motor capability discovery module works on the body structure model of the robotic puppet to calculate the specific positions that describe the range and discrete points in the workspace.

There are seven posture primitives found on each arm. Due to the properties of the body structure of the puppet figure, three basic posture primitives are generated on each arm by the hand string pulling the arm to do planar motion (Figure 6.9):

1. The posture of the arm lifted to the highest position.
2. The posture of the arm dropped down.
3. The middle position.

The elbow string dragging the arm backward brings about two posture primitives. The movable suspending point of the hand string induces another two posture primitives by opening the arm wide in 3D space. The pulley of the hand string has one DOF motion in the horizontal plane when the small platform rotates.

Each leg provides five posture primitives. Two posture primitives are formed on the leg by the knee strings in planar motion, one primitive by the foot string, and two primitives by the ankle string.

These posture primitives have already exhibited a rather large behavior diversity before more movements are built on them. By simply combining the posture primitives provided by the four limbs into full-body movements, there are totally $7 \times 7 \times 5 \times 5 = 1125$ distinct postures. These are only static gestures.

Two motion primitives, the reaching movement and the oscillation movement, are created based on these postures. The reaching movement interpolates between the starting postural primitive and the end postural primitive. The oscillation movement specifies the periodic motion of the limbs with four parameters: the magnitude, the central position, the starting position, and the speed of the oscillatory movement.

6.4.3 User Interface

Through the window-type user interface programmed with Motif, the user can directly control the behaviors of the robotic puppet. Besides displaying the feedback of system states, the user interface module allows the user to manually set motor primitives and edit

Figure 6.10: The user interface of the behavior editing program

Figure 6.11: The teaching panel

motion sequences. Involvement of a human in the process of behavior generation gives the movements of the robotic puppet human feature which is essential to illusion of life and readability of behavioral expression. Readability refers to how well the observer can correctly interpret the intended expression (Appendix C).

Manually Setting Primitives with Teaching Panel

With the teaching panel (Figure 6.11), the user can directly specify the postures which will be stored into the behavior repertoire as primitives. Manually selecting and setting the postural primitives is an essential function of the behavior generation program. User can select featured postures that may be meaningful in to human observers within the range of the workspace. Manually setting primitives is not only a major method to give the robotic puppet new behaviors, but also make the movements human-alike. The new primitives are incorporated into the behavior repertoire for motion editing.

Motor Speed Calibration

The speeds of the servo motors are not very accurate. The center of the zero-speed region varies from one motor to another. It roughly correspond to the 1.5ms pulse width of the PWM signals. Due to the differences on the zero-speed regions, one or two motors may begin to rotate though most motors halt when the power is on. The deviation of the center of the zero-speed region may further cause errors on predicting the motor speed. So the motor calibration module let the user compensate the speed error cause by the bias on the zero-speed region.

6.4.4 Motion Synthesis Module

The motion synthesis module prepares a list of behavior modules to ease the work of motion sequence generation (Figure 6.10). The motion synthesis is performed through interactive graphic operations with the mouse and the keyboard. The motion synthesis module has three main functions: *Movement Input*, *Edit*, and *Display*. The display function lists all the available behavior modules for the user's selection. With the *Input* and *Edit* functions, the user can compose a motion sequence for the robotic puppet. The behavior modules in

Figure 6.12: The robotic puppet ROMS-II can take a lot of postures

the behavior repertoire provide the user the building blocks of complex movements. It is easy to combine the primitives into a sequential motion sequence with the motion synthesis module. The reach movement will interpolate between two postures when the user skips the short transition.

If the motion sequence requires synchronization among the behavior modules working on different body parts, the user has to orchestrate them carefully. Besides a list of postures, the behavior repertoire also provides the user five behavior modules to produce coordinated and symmetrical body movements. They are the *radial* module, *waving* module, *segment*

module, *ipsilateral* module, and *cross* module. A *radial* module applies the movement specified to one limb to all other limbs. A *waving* module adds a small delay between the movements of the limbs. The temporal information is specified by human as the durations of the behavior modules.

6.5 Puppet Show

The visual results of the robotic puppet engaging behavioral expression are captured in a number of postures and a short motion.

Figure 6.13: A short puppet show. The puppet lifts up one leg, stretches it, and abduces it.

The postures shown in Figure 6.12 are designed as those that can be directly obtained by combining existing posture primitives. They are a small part of the behaviors the robotic puppet can perform. If the method of motion synthesis is applied to them, the potential variations of the behaviors from these postures and the primitives are enormous. These postures illustrate the diversity of behaviors and the remarkable capability of the puppet figure in encoding information into body movements.

In the short motion sequence (Figure 6.13), the puppet figure first lifts up its left leg, extends it to the extreme, and then abduces it to the left side. Several motors and strings are involved in this coordinated movement that is build on the basic behavior modules. This motion sequence demonstrates that complex behaviors could be generated and presented

on the robotic puppet with the aid the behavior editing software when time is combined into the movement generation.

6.6 Summary

The robotic puppet is implemented with robotics technology as an example of the physically embodied agents designed for expressive tasks. It has a rich collection of behaviors and enormous potential of expressing information through its gestures and body movements. This robotic puppet is used as avatar communicating with the human through nonverbal communication channel that is intuitive to the human.

Chapter 7

Conclusion

In the preceding presentation of the robotic puppets, we focus on the design of body structure and the behaviors for expressive tasks. A number of issues important to designing expressive physical agents that engage humans in natural, intuitive, and social communication through body movements are outlined. Some of these issues pertain to the design, especially behavioral design of the human-like physical agents, are the appearance, degrees of freedom, and motor capability. The basic requirements to the behaviors of human-like physical agents for expressive intention are the usability and comfortability. To effectively employ body movements as a natural communication channel in human-machine interaction, the physical agents must have a plenty of controllable and human-like movement patterns. As an example of the physical agent design, the robotic puppet systems are developed according to these principles for studying this form of non-verbal communication.

For the intended tasks of conveying information to human observers through body movements, the requirements to the design of the robotic puppets are specified with the design space that mainly includes the appearance, mechanical structure, and behavioral generation. The considerations in body structure design and the methods selected for behavior generation are all made towards a good match between the design space and the niche space.

The mechanical structure of the puppet figure has sufficient complexity to produce a wide range of gestures and body movements, but the large number of DOFs makes the behavior representation and control difficult. The degrees-of-freedom problem is solved in

three steps: constructing the robotic puppets in an underconstrained suspension structure, encapsulating the movements of local structures into motor primitives, and coordinating different body parts by adding constraints on them. The kinematics of the robotic puppets is established according to statics because the configurations of the puppet figures are determined partially by the gravitational force. To simplify the computation involved in the kinematics of the robotic puppets, some measures are taken to make the torso part stable during body movements, and then movements on each limb can be processed respectively. The behavior generation method for creating motor primitives takes an agent-centered perspective that considers the constraints and specific properties of the underlying mechanical structure of the robotic puppets. Some primitives in the behavior repertoire are produced based on its motor capability. Though this method provides a way to automatically generate behavior modules among the physical agents with various body structures, and the behavior produced with this method can be regarded as the innate behaviors of the robotic puppets, they do not have any feature of human movements which are necessary for the puppet figures to have human-like behaviors. Therefore, the robotic puppets provide three ways to merge human traits into its body movements: user can manually set the primitives; user can compose motion sequences with the behavior modules represented in a multilayered structure; captured or simulated human motion data can be transformed to the robotic puppets. Adapting human motion data is achieved through setting up correspondence between the motion sequence of the source model and the behavior modules of the target model because they do not share a common body structure.

The demonstrations of the robotic puppet ROMS-I and ROMS-II show that the structural and behavioral design of the robotic puppets approximately matches the requirements of the expressive tasks. The robotic puppets have remarkable potential to convey information through their body movements. The expressive effects are reached by causing 'illusion of life' on the observers but not 'imitation of life' [124], the prevalent behavior generation method in artificial life that imitates the underlying mechanisms of biological systems.

The degrees-of-freedom problem met in the physical agent design arises such a question: what are the methods may be used to solve the degrees-of-freedom problem? Rose [16] gave

three methods: structure design, kinematics calculation, and sensory feedback. Posner [17] proposed four methods: hierarchical structure, feedback, cost function, and inherent characteristics of body parts. Besides these methods, environment can also be used as a way to solve the degrees-of-freedom problem. It is the gravitational force that helps to constrain parts of the total DOFs of the suspended puppets, though their potential energies are minimum when they are in equilibrium.

7.1 Summary of Contributions

The making of the robotic puppets shows that structural and behavioral design of expressive physical agents involves a series of trade-offs between the multiple dimensions in the design space and the niche space. An excellent design must have precise identification to the requirements of behavior expression on the physical agents. Several important aspects associated with the usability and comfortability of behavior generation and expression are implemented on the robotic puppets. The salient features in the work towards a usable and comfortable physical agent include:

7.1.1 Complex Wire-Suspension Mechanisms

The traditional art form of puppetry is used here as a reference to the structures of the expressive physical agents. Knowledge from both robotics and puppet making was used in building the robotic puppets. This is the first systematic implementation of complex marionettes controlled with robotic techniques. The structural complexity of the robotic puppets exceeds previous implementations of wire-suspension robotic systems not only because of the multiple links of the puppet figures but also because of the behavior diversity for expressive tasks. For common robots, the number of the control variables equals to that of DOFs. The robotic puppets are designed to have an underconstrained wire-suspension structure whose configurations can be specified with a much smaller number of control variables than that of the puppet figures' DOFs. As an approach to address the degree-of-freedom problem, representing the movements of the puppets in the string space greatly

reduce the information required in behavior representation and generation. The puppet ROMS-I having 30 DOFs is controlled by 10 strings while ROMS-II has 35 DOFs and 14 strings. ROMS-II can exhibit a large number of body movements. In addition, the wire-suspension structure of the puppets allow the users to change the number of the strings according to the requirements of behavioral expression. The configurations of the robotic puppets are calculated with statics because the gravitational force takes a role in determining the postures of the puppet figures.

7.1.2 Agent-Centered Behavioral Design

In order to make the behaviors of the robotic puppet ROMS-II to match human's expectations, it is desirable to incorporate unique features of the physical agent into its behaviors besides the human-like movements. We use an agent-centered perspective in behavior design that the underlying body structures of the agents provide direct information for behavior generation. The expressive behaviors are not designated by human randomly. The primitives can be produced by examining the motor capability of the physical agents. The special configurations such as those corresponding to the boundary of workspace, singularity, and manipulability may be recruited as posture primitives. Because the behaviors obtained this way are closely correlated with the specific body structures, they can be regarded as representing the uniqueness or personality of the physical agents that is one major aspect of believable behaviors.

The posture primitives found with the method of motor capability discovery correspond to a series of discrete points in the workspace of the robotic puppets. They divide the whole workspace into many smaller regions where the kinematics is rather continuous. Within these regions between the primitives, inverse kinematics calculation needs not to be done on each point along the trajectories.

One major characteristic of the physical agents is that their body structures are highly diversified. The behaviors acquired from simulated characters or humans lack of portability among these physical agents whose structures and kinematics are different. Working on

the behavior primitives avoids to compare the details of the different structures. The behavior primitives serve as the basic building blocks and the bootstrap of behavior learning or transformation. The correspondence problem can be solved on the level of posture primitives but not kinematics. Though the posture primitives produced with the method of motor capability discovery are not portable among the physical agents, the method of motor capability discovery can be repeatedly applied to the new agents and automatically generate workable behavior modules on them.

7.1.3 Multilevel Behavior Representation

The large number of DOFs of the robotic puppets make the movements difficult to control and represent. Inspired by the modular feature and the hierarchical organization in animal's behaviors, we exploit the behavior modules and a layered structure in representing and organizing the movement patterns of the robotic puppet. This is an effective means to reduce the complexity of motion control while preserving the diversity of synthesized movements. The degree-of-freedom problem is addressed through a bottom-up process of motion synthesis that compresses the large number of DOFs at mechanical level to a smaller number of control variables at the behavioral level. The method of constraining the underlying mechanical structure of the robotic puppet in order to produce full body coordinated movements is motivated by the ethological study of living creatures. It provides the users a method to represent the behaviors of the puppets with less information. The behavior repertoire provides the users a list of available behavior modules represented in different levels of abstraction. These behavior modules may be produced either through the method of motor capability discovery or manually assigned by the users. In behavior design, the body movements of higher level abstraction can be separated from the details of the underlying mechanical structures, so part of the behavior repertoire is reusable for the physical agents with different body structures. The spatial and temporal features of body movements are specified respectively. People have the flexibility to select different control methods to achieve the functions of behavior modules according to the mechanical

structures of the puppets and intended tasks.

7.2 Assumptions and Limitations

The torso of the puppet figures is assumed to be fixed in space so that the full body can be divided into several groups of body parts that are processed individually in the inverse kinematics calculation and motion generation. Measurements, such as adding more weight to the torso, are taken to reduce the errors of body movements caused by the possible interactions between the arms and the legs. The assumption of considering the torso as a static object reduces some expressive capability of the robotic puppets. The flow of full body movements produced by the body parts joining in motion orderly can be blocked by the static torso.

The inverse kinematics is calculated according to statics. The movements of the robotic puppets are restricted to quasi-statics motion, such as taking postures and making posture transformations under low speed. Comparing with fully constrained mechanisms, where the motion states can be determined by kinematics, the dynamic properties of the under-constrained wire-suspension mechanisms have obvious influences to their motion states. Selecting the special motion state of equilibrium highly restricts the variations of body movements and thus the expressiveness of behaviors. They can not actively make body movements that are required in expressing forces and speed. Besides the robotic puppets, physical agents may take other forms. Dynamic analysis to the physical agents may be required by the expressive tasks.

A list of assumptions to the mechanical properties of shape, materials and connections are made in order to simplify the kinematics calculation. The strings is assumed to be inelastic. The changes of the string lengths and the energy stored in the strings are ignored. The frictions on the joints and pulleys are so small that can be ignored. The diameters of the limbs are treated as zero and the masses on them are regarded as equally distributed. The frictions of the clothes and the ground are ignored in kinematics calculation. They have limited effects in producing realistic movements on the robotic puppets.

As a way to address the complexities in representation and generation of possibly a great number of expressive behaviors, a limited number of primitives are used to build the behavior repertoire through combination. The primitives are also a way to reserve the specific properties of the mechanical structures in the expressive behaviors, and to isolate them from the general behavior generation processes. The primitives are conceptually generated based on the information exhibited by the underlying mechanical structures. In implementing the primitives on the robotic puppets, the configurations having special features, such as the boundary of the workspace, are selected as the primitives. The primitives generated this way are posture primitives. The workspace of one single arm of the robotic puppets which has only two degrees of freedom is approximately continuous. When the number of degrees of freedom on the body parts increases, the workspace quickly becomes so complex that the method of looking for special configurations in the workspace may not be able to work.

The large number of degrees of freedom causes two problems: computational complexity in specifying the configurations, the indeterminacy of possible configurations. The second aspect need not be discussed for the robotic puppets due to the assumption of fixed torso and rather simple structure of single limbs. For physical agents with other body structure, the indeterminacy may become unavoidable. The issue of indeterminacy is closely related with motor equivalence, one of the four major features of human movements.

7.3 Future Directions

The design of the robotic puppets fulfills the requirements of usability and comfortability. But we found there are still some problems and deficits needed to be eliminated through further work. For example, the behavior generation methods are still not convenient for the users. People have to spend much time to compose motion sequences until they have expected performances. There are many directions worth of further study. Some are related with the robotic puppets, such as dynamic modelling and control. The body movements of the robotic puppets are limited to static postures. Others are more general to the

physical agents in human-machine communication: the quantitative descriptions to the design matrices, the methods for measuring people's responses to the shown behaviors of the physical agents, the influences of social and cultural environments [4] to the expressive behaviors, synchronization between body movements and speech, movement generation and behavior expression on physical agents that do not resemble human, and bidirectional behavior communication. Two promising possibilities are described here.

7.3.1 Design Methodology

In traditional puppet making, a puppeteer with rich experience on both performance and design determines the appearance, clothes and mechanical structure of a puppet and the stage according to the story the puppet will show. After the puppet is completed and the strings are tied, the puppeteer will find out the movements the puppet can do through repeated trials. If the movements or appearance of a puppet fail to produce expected visual effects, it has to be redesigned. A puppeteer's experience on puppet's structure and performance is crucial to the success of puppet design. Similarly, when designing an expressive physical agent, the effects and limitations of the physical agent often become clear when the design and construction procedures have finished and the physical agent has been able to move. Then the designer continuously updates the physical agent iteratively until the desired performance is reached, as those have been done in [7] [69] [125]. The design process of the robotic puppet is such a case in which behavioral design succeeds structural design and conforms to all the constraints of its mechanical structure. We have designed three sets of robotic puppets. The complexity of physical agent design not only lies in a series of steps from morphology selection, behavioral design, to expression in real world, but also arises from the uncertainty in behavior selection in communication. Because of their tasks for social interaction, the appearance and structure of the physical agents have many variances, the structures designed this way may not always ensure the desired behaviors. Methods appropriate for behavior expression tasks are required to closely combine structural and behavioral design in order to generate satisfying expression effects.

Though many researches have contributed a lot to the issue of physical agent construction [43] [50] [22] [26] [84] [126] [127], and some ideas can be borrowed from product, interaction and stylized design, less work has been done on the design aspect, especially in traditional sense of industrial design.

7.3.2 Behavior Generation through Self-Organization

To generate complex behaviors on the physical agents, the primitives and body parts must be transiently or permanently coupled in some fashion. According to dynamic systems theory of motor control [15] [16], the environment is an important source of information for behavior generation, while movement is the result of the interaction between the goal of the performer and the constraints imposed by the environment. This theory explores how complex behavior patterns are formed through appropriate organizing and coordinating primitives and body parts. The behaviors are produced and coordinated through self-organization [119] but not a careful control of actions. Self-organization refers to the spontaneous formation of patterns and pattern changes in open and disequilibrium systems. Complex behaviors are the emergent outcomes from the interaction of multiple subsystems involved in motor control. With this method, autonomous behavior generation can be achieved and the resulting behaviors will be in accordance with the environment. As the dynamic systems theory represents the behaviors with abstract pattern variables, either the motor behaviors or the non-verbal communication can be modelled and studied as dynamic processes [83] [128] rather than as only static conditions. The major points discussed in behavioral design of the robotic puppets can be reexamined from the perspective of dynamic systems theory.

Bibliography

- [1] B. Reeves and C. Nass, *The Media Equation: How people treat computers, television, and new media like real people and places*. Cambridge, UK: Cambridge University Press, 1996.
- [2] J. Cassell, “Nudge Nudge Wink Wink: elements of face-to-face conversation for embodied conversational agents,” in *Embodied Conversational Agents*, J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, Eds. Cambridge, MA: MIT Press, 2000, pp. 1–27.
- [3] C. L. Breazeal, “Sociable machines: Expressive social exchange between humans and robots,” Ph.D. dissertation, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 2000.
- [4] J. K. Burgoon, D. B. Buller, and W. G. Woodall, *Nonverbal Communication: the Unspoken Dialogue*. New York: Harper and Row Publishers, 1989.
- [5] B. R. Duffy, “Anthropomorphism and the social robot,” *Robotics and Autonomous Systems, Special Issue on Socially Interactive Robots*, vol. 42, no. 3-4, pp. 177–190, 2003.
- [6] T. Fong, I. Nourbakhsh, and K. Dautenhahn, “A survey of socially interactive robots,” *Robotics and Autonomous Systems, Special Issue on Socially Interactive Robots*, vol. 42, no. 3-4, 2003.
- [7] K. Severinson-Eklundh, A. Green, and H. Huttenrauch, “Social and collaborative aspects of interaction with a service robot,” *Robotics and Autonomous Systems, Special Issue on Socially Interactive Robots*, vol. 42, no. 3-4, pp. 223–234, 2003.

- [8] S. Hashimoto, "Humanoid robots in Waseda University - Hadaly-2 and WABIAN," *Autonomous Robots*, vol. 12, no. 1, pp. 25–38, 2002.
- [9] K. Kawamura *et al.*, "ISAC: foundations in human-humanoid interaction," *IEEE Intelligent Systems*, vol. 15, no. 4, pp. 38–45, 2000.
- [10] D. Canamero, J. Fredslund, and R. B. Lunding. (2003) Feelix is a humanoid robot built out of lego. [Online]. Available: <http://www.pervasive.dk/projects/feelix/feelix-summary.htm>
- [11] Sarcos. (2003). [Online]. Available: <http://www.sarcos.com>
- [12] Honda. (2003). [Online]. Available: <http://www.honda.com.jp/ASIMO/>
- [13] Sony. (2002) Sony develops small biped entertainment robot SDR-4X. [Online]. Available: <http://www.sony.net/SonyInfo/News/Press/200203/02-0319E/>
- [14] F. Yamasaki *et al.*, "PINO the Humanoid: A basic architecture," in *Proceedings of The Fourth International Workshop on RoboCup*, Melbourne, Australia, August 2000, pp. 269–278.
- [15] H. N. Zelaznik, *Advances in Motor Learning and Control*. Champaign, IL: Human Kinetics, 1996.
- [16] D. J. Rose, *A Multilevel Approach to the Study of Motor Control and Learning*. Needham Heights, MA: Allyn and Bacon, 1997.
- [17] M. I. Posner, *Foundations of Cognitive Science*. Cambridge, MA: MIT Press, 1989.
- [18] C. Nehaniv and K. Dautenhahn, "The correspondence problem," in *Imitation in Animals and Artifacts*, K. Dautenhahn and C. L. Nehaniv, Eds. Cambridge, MA: MIT Press, 2002, pp. 41–62.
- [19] A. Sloman, "Exploring design space and niche space," in *Proceedings of the Fifth Scandinavian Conference on Artificial Intelligence*, Trondheim, Norway, May 1995, pp. 1–8.
- [20] P. Fraser, *Puppets and Puppetry*. London: B T Batsford Ltd, 1980.

- [21] S. S. Xing and I.-M. Chen, "Design expressive behaviors for robot puppets," in *The Seventh International Conference on Control, Automation, Robotics, Vision*, Singapore, December 2002, pp. 378–383.
- [22] J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, *Emobodied Conversational Agents*. Cambridge, MA: MIT Press, 2000.
- [23] K. R. Thorisson, "Communicative humanoids: A computational model of psychosocial dialogue skills," Ph.D. dissertation, Program of Media Arts and Sciences, School of Architecture and Planning, MIT, Cambridge, MA, 1996.
- [24] D. M. Dehn and S. V. Mulken, "The impact of animated interface agents: A review of empirical research," *Journal of Human Computer Studies*, vol. 52, no. 1, pp. 1–22, 2000.
- [25] J. Cassell *et al.*, "Human conversation as a system framework: Designing embodied conversational agents," in *Embodied Conversational Agents*, J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, Eds. Cambridge, MA: MIT Press, 2000, pp. 29–63.
- [26] R. Picard, *Affective Computing*. Cambridge, MA: MIT Press, 1997.
- [27] G. Ball and J. Breese, "Emotion and personality in a conversational agent," in *Embodied Conversational Agents*, J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, Eds. Cambridge, MA: MIT Press, 2000, pp. 190–219.
- [28] D. W. Massaro, *Perceiving Talking Faces: from Speech Perception to a Behavioral Principle*. Cambridge, MA: MIT Press, 1998.
- [29] J.-J. Lien *et al.*, "Detection, tracking, and classification of action units in facial expression," *Robotics and Autonomous Systems, Special Issue on Human Robot Interaction*, vol. 31, no. 3, pp. 131–146, 2000.
- [30] M. Coen, "Design principles for intelligent environments," in *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI, September 1998, pp. 547–554.
- [31] A. Pentland, "Smart rooms," *Scientific American*, vol. 274, no. 4, pp. 68–76, 1996.

- [32] T. Mori and T. Sato, "Robotic room: Its concept and realization," *Robotics and Autonomous Systems*, vol. 28, no. 2-3, pp. 141–148, 1999.
- [33] M. Inaba *et al.*, "A 35 DOF humanoid that can coordinate arms and legs in standing up, reaching and grasping an object," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Osaka, Japan, November 1996, pp. 29–36.
- [34] S. Kotosaka *et al.*, "Humanoid robot DB," in *Proceedings of International Conference on Machine Automation*, Osaka, Japan, September 2000, pp. 21–26.
- [35] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of Honda humanoid robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998, pp. 1321–1326.
- [36] B. Adams *et al.*, "Humanoid robot: A new kind of tool," *IEEE Intelligent Systems*, vol. 15, no. 4, pp. 25–31, 2000.
- [37] G. Tevatia and S. Schaal, "Inverse kinematics for humanoid robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 294–299, April 2000.
- [38] Y. Kuniyoshi and A. Nagakubo, "Humanoid as a research vehicle into flexible complex interaction," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robotic Systems*, Grenoble, France, September 1997, pp. 811–819.
- [39] F. Hara and H. Kobayashi, "A face robot able to recognize and produce facial expression," in *Proceedings of International Conference on Intelligent Robotic Systems*, Osaka, Japan, November 1996, pp. 1600–1607.
- [40] A. Billard, "Play, dreams and imitation in Robota," in *Socially Intelligent Agents: Creating Relationships with Computers and Robots*, K. Dautenhahn *et al.*, Eds. Boston, MA: Kluwer, 2002, pp. 165–172.
- [41] R. D. Schraft and G. Schmierer, *Service Robotics*. Natick, MA: AK Peters, 2000.
- [42] A. Druin and J. Hendler, *Robots for Kids: Exploring new technologies for learnings*. San Fransisco, CA: Morgan Kaufmann Publishers, 2000.

- [43] K. Dautenhahn, "Design spaces and niche spaces of believable social robots," in *Proceedings of the Eleventh International Workshop of Robot and Human Interactive Communication*, Berlin, Germany, September 2002, pp. 192–197.
- [44] C. Breazeal, "Toward sociable robot," *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 167–175, 2003.
- [45] M. Scheeff *et al.*, "Experiences with Sparky, a social robot," in *Proceedings of the Workshop on Interactive Robotics and Entertainment*, Pittsburgh, PA, April 2000, pp. 1–6.
- [46] E. Paulos and J. Canny, "Ubiquitous tele-embodiment: applications and implications," *International Journal of Human-Computer Studies*, vol. 46, no. 6, pp. 861–877, 1997.
- [47] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends in Cognitive Sciences*, vol. 6, no. 11, pp. 481–487, 2002.
- [48] S. Schaal, "Is imitation learning the route to humanoid robots," *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233–242, 1999.
- [49] G. Hayes and J. Demiris, "A robot controller using learning by imitation," in *Proceedings of the Second International Symposium on Intelligent Robots Systems*, Grenoble, France, July 1994, pp. 198–204.
- [50] C. Breazeal, *Designing Sociable Robots*. Cambridge, MA: MIT Press, 2002.
- [51] S. Kawamura, W. Choe, S. Tanaka, and S. Pandian, "Development of an ultrahigh speed robot FALCON using wire drive system," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1995, pp. 215–220.
- [52] S. Kawamura and K. Ito, "A new type of master robot for teleoperation using a radial wire drive system," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1993, pp. 55–60.
- [53] J. W. Jeong, S. H. Kim, and Y. K. Kwak, "Kinematics and workspace analysis of a parallel wire mechanism for measuring a robot pose," *Mechanism and Machine Theory*, vol. 34, no. 6, pp. 825–841, 1999.

- [54] R. L. William, "Cable-suspended haptic interface," *International Journal of Virtual Reality*, vol. 3, no. 3, pp. 13–21, 1998.
- [55] H. Kino and S. Kawamura, "Study on multiple degree-of-freedom positioning mechanism using wires," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, DC, May 2002, pp. 829–834.
- [56] J. Albus, R. Bostelman, and N. Dagalakis, "The NIST ROBOCRANE," *Journal of Robotic Systems*, vol. 10, no. 5, pp. 709–724, 1993.
- [57] T. Arai and B. Osumi, "Three wire suspension robot," *Industrial Robot*, vol. 19, no. 4, pp. 17–22, 1992.
- [58] A. Fattach and S. K. Agrawal, "Design of cable-suspended planar parallel robots for an optimal workspace," in *Proceedings of the Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, Quebec, Canada, October 2002, pp. 195–202.
- [59] J. K. Gorman, K. W. Jablokow, and D. J. Cannon, "The cable array robot: theory and experiment," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001, pp. 2804–2810.
- [60] L. Maeda *et al.*, "On design of a wire-driven parallel robot WARP manipulator," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, MI, May 1999, pp. 895–900.
- [61] N. Yanai, M. Yamamoto, and A. Mohri, "Inverse kinematics analysis and trajectory generation of incompletely restrained wire-suspended mechanisms," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001, pp. 3489–3494.
- [62] R. Roberts, T. Graham, and T. Lippitt, "On the inverse kinematics, statics, and fault tolerance of cable-suspended robots," *Journal of Robotic Systems*, vol. 15, no. 10, pp. 581–597, 1998.
- [63] M. Yamamoto and A. Mohri, "Inverse kinematics analysis for incompletely restrained parallel wire mechanisms," in *Proceedings of the IEEE/RSJ International Conference*

- on Intelligent Robots and Systems*, Kagawa University, Japan, October 2000, pp. 504–509.
- [64] A. Ming and T. Higuchi, “Study on multiple degree-of-freedom positioning mechanism using wires,” *International Journal of the Japan Society for Precision Engineering*, vol. 28, no. 2, pp. 131–138, 1994.
 - [65] H. Kitano *et al.*, “Sony legged robot for RoboCup challenge,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998, pp. 2605–2612.
 - [66] T. Nakata, “Producing animal-likeness on artifacts and analyzing its effect on human behavioral attitudes,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Kyongju, Korea, October 1999, pp. 549–554.
 - [67] J. K. Hodgins *et al.*, “Animating human athletics,” *Computer Graphics, Proceedings of SIGGRAPH 95, Los Angeles, CA*, pp. 71–78, 1995.
 - [68] K. Hirai, “Current and future perspective of Honda humanoid robot,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Grenoble, France, September 1997, pp. 500–508.
 - [69] T. Furuta *et al.*, “Design and construction of a series of compact humanoid robots and development of biped walk control strategies,” *Robotics and Autonomous Systems*, vol. 37, no. 2, pp. 81–100, 2001.
 - [70] M. Kim, S. Kang, and S. Lee, “Design and control of a humanoid robot CENTAUR,” in *Proceedings of the Second International Symposium on Humanoid Robots*, Tokyo, 1999, pp. 96–101.
 - [71] R. C. Arkin, *Behavior-based Robotics*. Cambridge, MA: MIT Press, 1998.
 - [72] N. Hogan, “Impedance control: An approach to manipulation,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 1–24, 1988.
 - [73] M. J. Mataric, V. B. Zordan, and M. M. Williamson, “Making complex articulated agents dance: An analysis of control methods drawn from robotics, animation, and biology,” *Autonomous Agents and Multi-Agent Systems*, vol. 2, no. 1, pp. 23–43, 1999.

- [74] M. Kawato, "Internal models for motor control and trajectory planning," *Current Opinion in Neurobiology*, vol. 9, no. 6, pp. 718–727, 1999.
- [75] B. Bodenheimer *et al.*, "The process of motion capture: Dealing with the data," in *Eurographics Workshop on Animation and Simulation*, Budapest, Hungary, September 1997, pp. 3–18.
- [76] A. Bruderlin and L. Williams, "Motion signal processing," *Computer Graphics, Proceedings of SIGGRAPH 95, Los Angeles, CA*, pp. 97–104, August 1995.
- [77] A. Gildfind *et al.*, "Evolving performance control systems for digital puppetry," *Journal of Visualization and Computer Animation*, vol. 11, no. 4, pp. 169–183, 2000.
- [78] M. Riley and C. G. Atkeson, "Robot catching: Towards engaging human-humanoid interaction," *Autonomous Robots*, vol. 12, no. 1, pp. 119–128, 2002.
- [79] J. K. Hodgins and N. Pollard, "Adapting simulated behaviors for new characters," in *Computer Graphics, Proceedings of SIGGRAPH 97, Los Angeles, CA*, August 1997, pp. 153–162.
- [80] M. Gleicher, "Retargeting motion to new characters," in *Computer Graphics, Proceedings of SIGGRAPH 98, Orlando, FL*, July 1998, pp. 33–42.
- [81] Z. Popovic and A. Witkin, "Physically based motion transformation," in *Computer Graphics, Proceedings of SIGGRAPH 99, Los Angeles, CA*, August 1999, pp. 11–20.
- [82] K. Yamane, J. K. Hodgins, and H. B. Brown, "Controlling a marionette with human motion capture data," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003, pp. 3834–3841.
- [83] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Trajectory formation for imitation with nonlinear dynamical systems," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, Hawaii, October 2001, pp. 752–757.
- [84] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MA: MIT Press, 1984.
- [85] J. Bates, "The role of emotion in believable characters," *Communications of the ACM*, vol. 37, no. 7, pp. 122–125, 1994.

- [86] B. Abernethy, *Biophysical Foundations of Human Movement*. Champaign, IL: Human Kinetics, 1997.
- [87] X. Tu, “Artificial animals for computer animation: Biomechanics, locomotion, perception and behavior,” Ph.D. dissertation, Department of Computer Science, University of Toronto, Canada, 1996.
- [88] J. S. Albus, *Brains, Behavior, and Robotics*. Peterborough, NH: BYTE Books, 1981.
- [89] F. A. Mussa-Ivaldi, “Modular features of motor control and learning,” *Current Opinion in Neurobiology*, vol. 9, no. 6, pp. 713–717, 1999.
- [90] M. M. Williamson, “Postural primitives: interactive behavior for a humanoid robot arm,” in *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, Cape Cod, MA, September 1996, pp. 124–131.
- [91] M. J. Mataric, M. Williamson, J. Demiris, and A. Mohan, “Behavior-based primitives for articulated control,” in *From Animals to Animats, Proceedings of Fifth International Conference on Simulation of Adaptive Behavior*, R. Pfeifer *et al.*, Eds. Cambridge, MA: MIT Press, 1998, pp. 165–170.
- [92] A. J. Ijspeert, “A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander,” *Biological Cybernetics*, vol. 84, no. 5, pp. 331–348, 2001.
- [93] M. J. Mataric, “Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics,” in *Imitation in Animals and Artifacts*, K. Dautenhahn and C. L. Nehaniv, Eds. Cambridge, MA: MIT Press, 2002, pp. 391–422.
- [94] C. Breazeal and B. Scassellati, “Infant-like social interactions between a robot and a human caretaker,” *Adaptive Behavior*, vol. 8, no. 1, pp. 49–74, 2000.
- [95] E. Bizzi and F. A. Mussa-Ivaldi, “The acquisition of motor behavior,” *Daedalus: The Brain*, vol. 127, no. 2, pp. 217–232, 1998.
- [96] C. G. Atkeson and S. Schaal, “Robot learning from demonstration,” in *Proceedings of the Fourteenth International Conference of Machine Learning*, Nashville, Tennessee, July 1997, pp. 12–20.

- [97] K. Dautenhahn, B. Ogden, and T. Quick, "From embodied to socially embedded agents: Implications for interaction-aware robots," *Cognitive Systems Research*, vol. 3, no. 3, pp. 397–428, 2002.
- [98] A. B. Loyall, "Believable agents: Building interactive personalities," Ph.D. dissertation, Computer Science Department, Carnegie Mellon University, Pittsburg, PA, 1997.
- [99] N. Badler, C. Phillips, and B. Webber, *Simulating Humans: Computer Graphics, Animation and Control*. New York: Oxford University Press, 1993.
- [100] E. Wise, *Animatronics: a guide to animated holiday displays*. Indianapolis, IN: Prompt Publications, 2000.
- [101] M. Mori, *The Buddha in the Robot: A Robot Engineer's Thoughts on Science and Religion*. Boston, MA: Charles E. Tuttle Co., 1982.
- [102] R. G. Douglas, *Behavioral and Brain Sciences*. New York: Cambridge Press, 1972.
- [103] B. Sheridan, *Telerobotics, Automation and Human Supervisory Control*. Cambridge, MA: MIT Press, 1992.
- [104] V. D. Nguyen, "Constructing force-closure grasps," in *International Journal of Robotics Research*, vol. 7, no. 3, 1988, pp. 3–16.
- [105] Y. Shen, H. Osumi, and T. Arai, "Manipulability measures for multi-wire driven parallel mechanisms," *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 550–554, December 1994.
- [106] M. Yamamoto, N. Yanai, and A. Mohri, "Inverse dynamics and control of crane-type manipulator," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Kyonjyu, Korea, October 1999, pp. 1228–1233.
- [107] V. M. Zatsiorsky, *Kinematics of Human Motion*. Champaign, IL: Human Kinetics, 1998.
- [108] R. Laban, *The Mastery of Movement*, 4th ed. Plymouth, UK: Northcote House, 1992.

- [109] J. D. Faires and R. Burden, *Numerical Methods*, 2nd ed. Pacific Grove, CA: Brooks/Cole Publishing Company, 1998.
- [110] R. R. Murphy, *Introduction to AI Robotics*. Cambridge, MA: MIT Press, 2000.
- [111] A. Fod, M. J. Mataric, and O. C. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous Robots*, vol. 12, no. 1, pp. 39–54, 2002.
- [112] S. F. Giszter, K. A. Moxon, and J. K. Chapin, "A neurobiological perspective on humanoid robot design," *IEEE Intelligent Systems*, vol. 15, no. 4, pp. 64–69, 2000.
- [113] T. Nakata, T. Mori, and T. Sato, "Analysis of impression of robot bodily expression," *Journal of Robotics and Mechatronics*, vol. 14, no. 1, pp. 27–36, 2002.
- [114] S. P. Wise and R. Shadmehr, "Motor control," in *Encyclopedia of the Human Brain*, V. S. Ramachandran, Ed. Elsevier Science, 2002, vol. 3, pp. 1–21.
- [115] A. P. Georgopoulos, "Cognitive motor control: Spatial and temporal aspects," *Current Opinion in Neurobiology*, vol. 12, no. 6, pp. 678–683, 2002.
- [116] A. Billard, "Learning motor skills by imitation: A biologically inspired robotic model," *Cybernetics and Systems, Special Issue on Imitation in Animals and Artifacts*, vol. 32, no. 1-2, pp. 155–193, 2001.
- [117] R. A. Schmidt and T. D. Lee, *Motor Control and Learning: A Behavioral Emphasis*, 3rd ed. Champaign, IL: Human Kinetics, 1999.
- [118] M. T. Turvey, "Coordination," *American Psychologist*, vol. 45, no. 8, pp. 938–953, 1990.
- [119] J. A. S. Kelso, *Dynamic Patterns: the Self-organization of Brain and Behavior*. Cambridge, MA: MIT Press, 1995.
- [120] Hackney, *Making Connections - Total Body Integration Through Barteneiff Fundamentals*. New York: Gordon and Breach Science Publishers, 1998.
- [121] W. Wyrwicke, *Imitation in Human and Animal Behavior*. New Brunswick, NJ: Transaction Publisher, 1996.

- [122] K. A. Thoroughman and R. Shadmehr, "Learning of action through adaptive combination of motor primitives," *Nature*, vol. 407, no. 6805, pp. 742–747, 2000.
- [123] S. Trenholm and A. Jensen, *Interpersonal Communication*. Belmont, CA: Wadsworth Publishing, 1999.
- [124] W. G. Walter, "An imitation of life," *Scientific American*, pp. 42–45, May 1950.
- [125] T. Willeke *et al.*, "The history of the mobot museum robot series: an evolutionary study," in *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conferenc*, Key West, FL, 2001, pp. 514–519.
- [126] C. Bartneck, "eMuu: an embodied emotional character for the ambient intelligent home," Ph.D. dissertation, Technical University of Eindhoven, The Netherlands, 2002.
- [127] E. Falcone *et al.*, "The personal rover project: the comprehensive design of a domestic personal robot," *Robotics and Autonomous Systems: Special Issue on Socially Interactive Robots*, vol. 42, no. 3-4, pp. 245–258, 2002.
- [128] J. H. Watt and C. A. VanLear, *Dynamic Patterns in Communication Processes*. Thousand Oaks, California: SAGE Publications, Inc., 1996.

Appendix A

Figures of Puppet Dimensions

Figure A.1: The size of ROMS-I's puppet body

Figure A.2: The size of ROMS-II's puppet body

Figure A.3: The size of ROMS-III's puppet body

Appendix B

Data Structures

The data structures of the *Servo*, *Wire*, *Joint*, *Link* represent the geometrical and physical properties of the components of a robotic puppet system. The data structures related to body movements include *Synergy*, *Posture*, and *Sequence*.

```
// Definition of one servo motor.
typedef struct{
    int      servo_ID;           // ID number of one servo motor.
    char     servo_name[];      // name of this motor.
    float    x_distance;        // position of the motor in the
    float    y_distance;        // Cartesian space.
    float    z_distance;
    float    diameter;          // diameter of the pulley.
    int      string_ID;         // the string this motor pulls.
    int      init_position;     // initial motor angular position.
    int      current_position;  // current motor angular position.
    int      current_speed;     // current motor speed.
    int      zero_speed;        // a value making the motor stop.
    int      lower_limit;       // lower boundary of zero speed area.
    int      upper_limit;       // upper boundary of zero speed area.
    int      speed;             // default motor speed.
} Servo;

// Definition of one string.
typedef struct{
    int      wire_ID;           // ID number of one string.
    char     wire_name[];       // name of this string.
    int      link_ID;           // the link this string connects.
    float    suspending_x;      // suspending position on platform in
    float    suspending_y;      // the Cartesian space.
    float    suspending_z;      //
    float    tie_x;             // attaching position on body part in
```



```

float    tie_y;           // the Cartesian space.
float    tie_z;           //
float    max_length;     // maximum string length.
float    length;         // current string length.
float    alpha;          // rotation angles in the joint space of
float    beta;           // the puppet.
float    tension;        // tension along the string (N).
} Wire;

```

```

// Definition of one joint.
typedef struct{
    int    joint_ID;      // ID number of one joint.
    char    joint_name[]; // name of this joint.
    int    joint_type;    // hinge=1, universal=2, spherical=3.
    float   diameter;     // diameter of the joint.
    float   mass;         // mass of the joint.
    float   x;            //
    float   y;            // attaching position on body part in
    float   z;            // the Cartesian space.
    int     inner_link_ID; //
    int     outer_link_ID; //
} Joint;

```

```

// Definition of one link.
typedef struct{
    int    link_ID;      // ID number of one link.
    char    link_name[]; // name of this link.
    float   length;      // size of the link.
    float   width;       //
    float   depth;       //
    float   mass_distance; // mass center position on the link.
    float   wire_distance; // wire attaching point on the link.
    float   mass;        //
    float   joint_alpha;  // rotation angles with respect to
    float   joint_beta;   // the previous link.
    float   center_x;     // position of the mass center in
    float   center_y;     // the Cartesian space.
    float   center_z;     //
    float   tension;      // tension along the string (N).
} Link;

```

```

// Definition of the robotic puppet.

```

```

struct puppet{
    char    name[];           //
    Servo    motor_list[];    // lists of the system parts.
    Wire     string_list[];   //
    Joint    joint_list[];    //
    Link     link_list[];     //
    Link     torso;           // the root link.
}

// Definition of one local couple of links and strings.
typedef struct{
    int      synergy_ID;      // ID number of one couple.
    char     synergy_name[];  // name of the couple.
    int      joint_ID[];      // list of joints in the couple.
    int      link_ID[];       // list of links involved.
    int      wire_ID[];       // list of wires involved.
} Synergy;

// Definition of one posture primitive.
typedef struct{
    int      posture_ID;      //
    char     posture_name[];  // name of the posture.
    Servo    motors[];        // list of motor states.
    Wire     strings[];       // string configuration.
    Link     links[];         // links in the joint space.
    int      duration;        // used for step motion.
} Posture;

// Definition of one motion sequence.
typedef struct{
    int      motion_ID;       //
    char     motion_name[];   // name of the motion sequence.
    Posture  *postures;       // list of postures.
} Sequence;

```

Appendix C

Function Definitions

The functions used by the puppeteer software provide the user a GUI interface, behavior input and editing, and motor control capability. A typical module in the puppeteer software has four parts supported with the functions. They are the functions that create a GUI interface, the callback functions for making response to the user's input, behavior calculation, and low-level motor control.

```
/* Functions for manually setting the primitives in teaching.c */

//---- Create the GUI. ----//

//create servo item list in calibration panel.
void create_teach_items( Widget * )

//create GUI of motor calibration panel.
void create_teach_panel( Widget * )

//----- Add callback functions to the GUI. ----//

// Callback for saving posture input.
void save_teach_posture_cb( Widget *, XtPointer *, XtPointer * )

// Callback for resetting servo speed values to the default values.
void reset_teach_posture_cb( Widget *, XtPointer *, XtPointer * )

// Callback for returning and exiting the teaching panel.
void exit_teach_panel_cb( Widget *, XtPointer *, XtPointer * )

// Callback for temporarily store posture input.
void update_teach_scale_cb (Widget *, XtPointer *, XtPointer *)
```

```

    // Callback for moving one motor according to user's assigned speed.
void move_teach_cb ( Widget *, XtPointer *, XtPointer * )

    // ---- Execute the operations according to user's input. ----//

    // Output the operation states.
int display_status_message_teach( char * )

    // Run one motor to the expected position.
int motor_to_position( Posture * )

    // Save motion sequence data to file.
int save_teach_to_file( Posture * )

    // Clear the motion sequence for new input.
int reset_teach_posture( void )

    // Combine motor action into current posture.
int update_posture ( Posture *, Servo * )

    /* Functions of movement calculation, timing, emotion, primitives in utility.c
    */

    // Output status message on the main window.
void display_status_message( char * )

    // Cause delay in millisecond.
int delay_millisecond( int )

    // Create a timer.
int new_timer ( void )

    // Initialize global variables of motion sequence, emotion, primitives.
int initialize_global_variables( void )

    // Check if the selected file is writable.
int is_writable ( char * )

    // Create new behavior file.
int new_file( char * )

    // Open a behavior file.
int open_file( char * )

    // Save motion sequence into a file.
void save_behavior_file ( char *, Sequence * )

```

```
// Move the puppet to the initial posture.
int home_puppet ( Posture * )

// Open and set initial states to the puppet.
int start_puppet( Posture * )

// Transition between two postures.
int change_posture( Posture *, Posture *, int * )

// Transform posture primitive to motor instructions.
int transform_motor_instruction( Posture * )

// Execute motion sequence.
int run_puppet ( Sequence * )

// Loop motion sequence.
void loop_puppet ( Sequence *, int * )

// Set emotion states.
int set_emotion_states( Emotion * )

// Modify the emotion states.
int change_emotion ( Emotion * )

// Bias motion states according to the emotion states.
int emotionalize( Sequence *, Emotion *, int * );

// Produce motion sequence through linear interpolation.
int interpolate( Posture *, Posture * );

// Produce motion sequence of periodic movement.
int periodic( Synergy *, float *, int, Sequence * )

// Produce contraction or extension.
int radial_motion( Synergy *, Sequence * );

// Produce delayed movements on the limbs.
int waving_motion( Synergy *, Sequence * );

// Produce synchronized motion on arms and legs.
int segment_motion( Synergy *, Synergy *, Sequence * );

// Produce synchronized motion on the left and right.
int ipsi_motion( Synergy *, Synergy *, Sequence * );

// Produce synchronized motion across the body.
int cross_motion( Synergy *, Synergy *, Sequence * );
```

```
/* Functions for motion synthesis with primitives in mainwindow.c */

//---- Create the GUI. ----//

// Create help information dialog.
void build_about_dialog( Widget * )

// Create menu items.
Widget build_menu_bar( Widget * )

// Initialize application state display.
Widget build_status_message( Widget * )

// Display behavior primitives on the left side of the main window.
void build_primitive_outline( Widget *, Posture * )

// Create behavior sequence window on the right part of the main window.
void build_behavior_workarea( Widget * )

// Create the window for motion synthesis.
Widget build_work_area( Widget * )

// Create the main window.
void puppet_window( Widget * )

//----- Add callback functions to the GUI. ----//

// Callback function for primitive selection.
void primitive_select_cb( Widget *, XtPointer *, XtPointer * )

// Callback function for motion sequence editing.
void sequence_select_cb( Widget *, XtPointer *, XtPointer * )

// Callback functions for menu operation.
void file_menu_cb( Widget *, XtPointer *, XtPointer * )
void edit_menu_cb( Widget *, XtPointer *, XtPointer * )
void simu_menu_cb( Widget *, XtPointer *, XtPointer * )
void puppet_menu_cb( Widget *, XtPointer *, XtPointer * )
void option_menu_cb( Widget *, XtPointer *, XtPointer * )
void help_menu_cb(Widget *, XtPointer *, XtPointer * )

// ---- Execute the operations according to user's input. ----//

// Rread in avaiable primitives.
void read_primitives ( char *, Posture * )
```

```
// Redisplay current motion sequence in the behavior window.
int update_behavior_workarea( Sequence * )

// Insert posture primitive into motion sequence.
int copy_primitive( Posture *, Sequence * )

// Delete posture primitive from motion sequence.
int delete_primitive( int, Sequence * )

// Set speed of a motion sequence
int set_motion_speed( Sequence *, float )

// create internal model of the puppet system.
int init_link_values( char * )

// Set servo variables with the default values.
int init_servo_values( char * )

// Set joint variables with the default values.
int init_joint_values( char * )

// Set string variables with the default values.
int init_wire_values( char * )

// Read in synergies or set with default values.
int init_synergy_values( char * )

/**** Functions for motor control in motor.c ****/

// Open serial port for motor command execution.
int open_serial_port( void )

// Send motor commands through serial port with flow control.
int write_motor_instruction( int, int * )

// Display current motor states.
void print_motor_position ( int * )

// Move the motors to initial positions.
void MotorInit( int * )

// Initialize motor positions and velocities.
void motor_variable_initialize( void )

// Initialize serial port.
int communication_initialize( void )
```

```
// Internal motor position transformation.
void to_position( Posture *, int * )

// Insert transition in motion sequence.
int sequence_check ( Sequence * )

// Halt all the motors.
int halt_motors( Posture *, int )
```