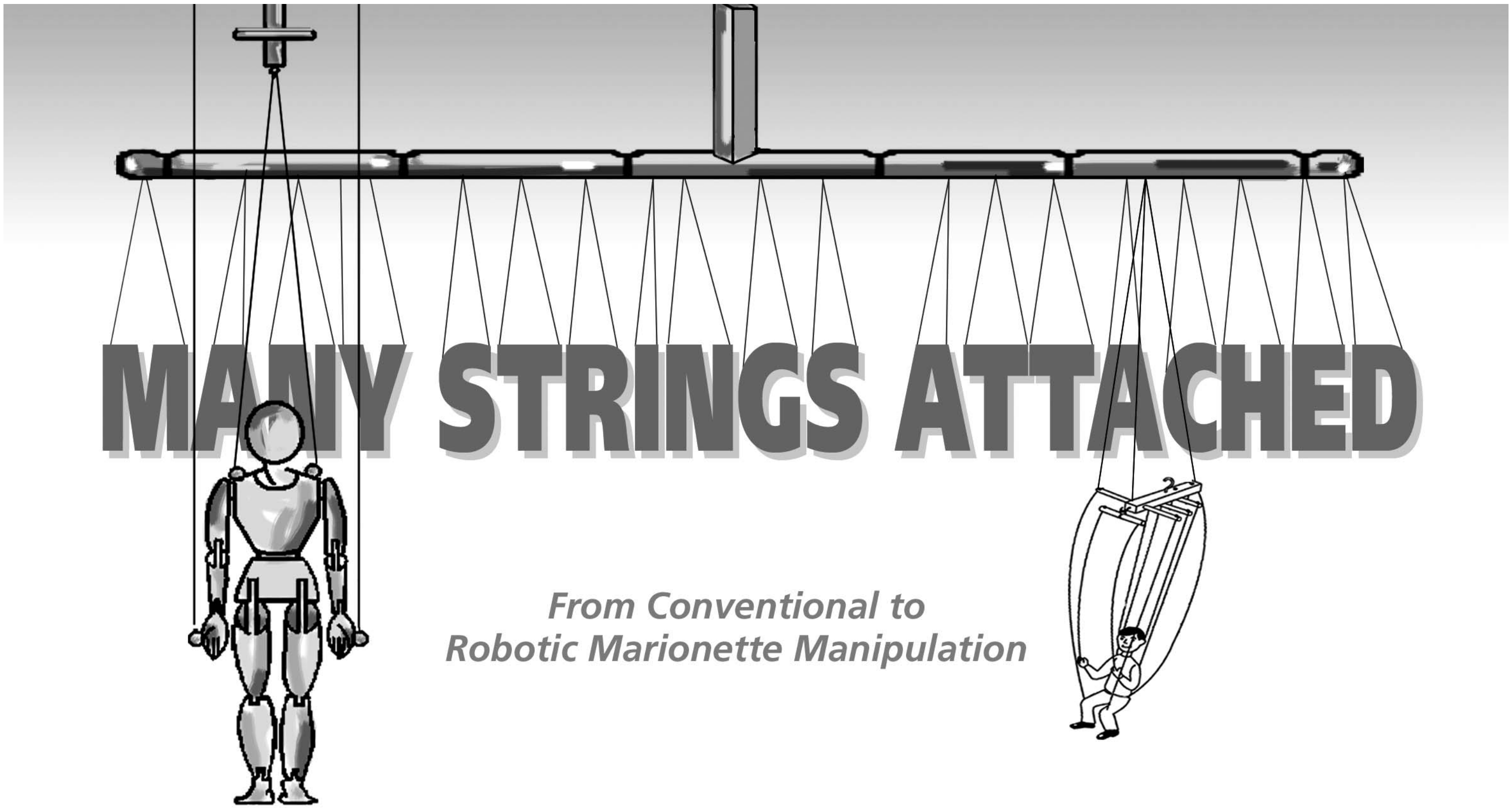
Many strings attached – From conventional to robotic marionette manipulation

Article *in* IEEE Robotics & Automation Magazine · April 2005

DOI: 10.1109/MRA.2005.1411420 · Source: IEEE Xplore



MANY STRINGS ATTACHED

From Conventional to Robotic Marionette Manipulation

BY I-MING CHEN, SHUSONG XING, RAYMOND TAY, AND SONG HUAT YEO

Puppetry and puppet theater are popular art forms having a long and fascinating heritage in many cultures. The survival of this art form is due to man's fascination with the inanimate object animated in a dramatic manner [1], and humans' curiosity in producing an exact artificial copy of themselves. Recent advances in the humanoid robots speak for this quest of curiosity. Most types of puppets in use today fall into four broad categories [1]:

- hand (or glove) puppets
- rod puppets
- marionettes
- shadow puppets.

The glove puppet is used like a glove on the operator's hand. The rod puppet is held and moved by rods, usually from below but sometimes from above. The marionette is a puppet on strings, suspended from a control mechanism held by the puppeteer. Shadow puppets are usually flat cut-out figures held against a translucent, illuminated screen. In general, people are largely interested in the theatrical and artistic content of puppetry, though basic puppet fabrication and manipulation techniques follow physical laws and engineering principles. From an engineering point of view, the puppet, the puppet control device, and the puppeteer form an interesting, complex mechanical system that produces lifelike movement in artificially made entities. Although there are studies in the literature dedicated to the practice of puppetry, it is hard to find any significant engineering study of puppet-making and manipulation skills. This is probably because specific puppet-making and manipulation skills are usually passed down from the puppet masters through apprenticeship and usually are not recorded and made publicly available.

This article serves two purposes: to introduce the evolution of traditional marionette design and manipulation skills from the engineering perspective and to describe a novel system developed in Nanyang Technological University. This system, RObotic Marionette Systems (ROMS), is capable of manipulating the puppet through mechatronic means based on the mechanics of the marionette rather than human operation. The hardware of ROMS, consisting of the puppeteer mechanism and motor control network is described later.

The marionette is chosen as the object of study for three reasons: 1) mechanically, it is equivalent to a cable-operated multiple rigid body system that exhibits rich kinematic and dynamic behaviors; 2) theatrically, the marionette performances are graceful, charming, and sometimes mystical because of the invisible string control; 3) the marionette is versatile and can be simple or complex in both construction and control, depending on usage. The marionette may create lifelike movements according to programmed motion commands issued from the computer and a motor-driven puppeteer mechanism. In the "Puppet Motion Design and Puppeteer Software" section of this article, the marionette motion generation is studied and demonstrated through three phases: posture primitives, multilayered motion synthesis, and motion transformation. This article ends with a discussion of the current capability and potential of the system. Through the development of this novel system, we hope to compliment the arts of traditional puppetry using a mechatronic approach and bring puppetry to a different level.

Anatomy of Marionettes

The term *marionette* was first associated with string puppets in

1070-9932/05/$20.00©2005 IEEE

16th century Europe. The origin of the word may be traceable to the Virgin Mary, often the principal character of puppet plays during the 1500s, either as a diminutive of "Maria," or in its literal translation "little Marys," from the French reference to the Virgin [2]. In China, it is called *Xuan Si Mu Ou* or *Ti Xian Mu Ou*, meaning "string-suspended puppet." In the early history of Europe, marionettes were used to entertain people, whereas in China, marionettes were mainly used for ceremonial purpose. The marionette theaters were used to entertain gods instead of ordinary people [3]. Hence, the appearance and fabrication techniques of marionettes have strong cultural influences. However, despite cultural differences, the basic structure and anatomy of the marionettes are similar. In fact, this structure also changed very little through history. A marionette usually consists of three essential elements: a puppet figure, a puppet control device, and a bundle of strings tied to the control device and various locations on the puppet figure, as shown in Figure 1(a).

Puppet Figure

The puppet figure is a collection of joined rigid bodies, usually a duplication or simplification of human, animal, or other kind of living creatures. The mechanical design of the puppet covers the task of determining the number and shapes of limbs for the figure, types of joints connecting the limbs, and additional features on the puppet, such as blinking eyes, moveable mouthpieces, and articulated palms for human figures. The dexterity of puppet movement increases with the addition to the figure of more segments of limbs. One particular example is the inclusion of the torso segment in a human figure to enable turning and bowing motions. The design of the puppet joints is versatile. Joints with one degree of freedom (DOF) revolute motion; two- and three-DOF spherical motions are frequently used in marionettes. Practical construction methods of the puppet figures and joints can be found in [1] and [4]–[8]. As shown in Figure 1(b), the shoulder, torso, and limbs of a human puppet are all connected to each other through joints of different DOFs. A famous modern Czech marionette, Hurvinek, and a traditional Chinese marionette, Zhong Kui (a ghost-fighting god) [9] are shown in Figure 1(c) and (d), respectively. The size of marionette figures is usually 45–75 cm in height with consideration of the visual effect and convenience for puppeteer operations.

Strings

In robotics terms, the strings of the marionette allow the puppeteer to remotely control the puppet from a control device. By manipulating the control device and plucking the strings, the puppet will produce lifelike motions. The strings act like actuators to the puppet.

Functions of Strings

The strings in a marionette can be classified into three categories: 1) strings for support/reference; 2) strings for motion control; and 3) strings for special effect. The supporting strings hold most of the weight of the puppet figure and are usually kept stationary during the performance. In the design of Western human marionettes, the two shoulder strings usually serve this purpose [Figure 2(a)]. The traditional Chinese marionette includes a single reference string, which acts as a *backbone* to the marionette [Figure 2(b)] to ensure that the marionette is in an upright position. All remaining strings will take reference from this string. The strings attached to the arms, legs, and head of the figure are mainly for motion-control purpose. The manipulation range of these strings attached to the limbs may be very large depending on the intended range of gestures of the figure. With the inclusion of head strings, the marionette may shake or nod its head. For a very basic human marionette, at least eight strings are needed for full-body motion; two for the shoulders (support), two each

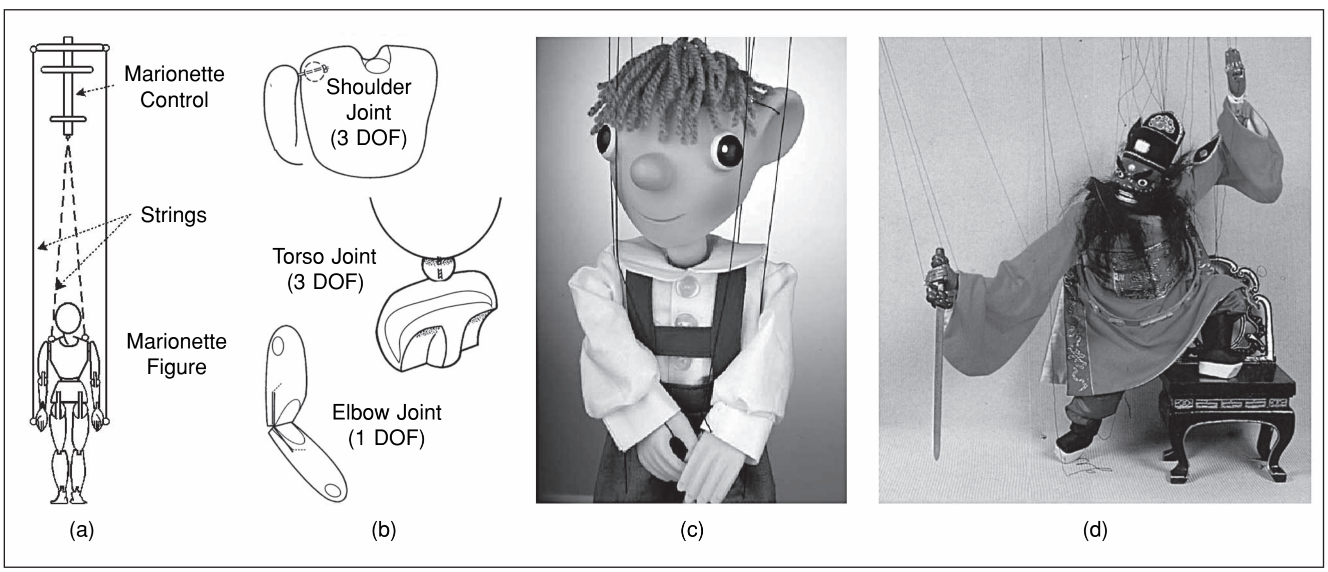


Figure 1. *(a) The anatomy of a marionette; (b) types of joints; (c) Czech marionette (Hurvinek by Sotak); and (d) Chinese marionette (Zhong Kui) [9].*

for the arms, legs, and head (control). Usually one back string is added to a Western marionette having a torso joint [Figure 2(a)]. With this additional back string, the puppet figure may perform a bowing motion.

Number of Strings

Using 8–9 strings in a generic marionette is only a rule of thumb. With simplified puppet figure design, the required number of strings can be reduced. The actual number of strings used in a marionette depends on the desired performance of the figure. A typical Western, human-figure marionette uses 8–12 strings. Traditional Chinese marionettes normally use 16–24 strings for producing very refined and delicate human-like manipulation. The detail controls usually focus on facial expressions and arm and palm movements. In video footage of a traditional Chinese marionette performance called *Drunken Zhong Kui,* produced by the puppet master Mr. Huang Yique from Quanzhou, China [10], the figure of Zhong Kui [Figure 1(d)] comprises over 50 strings. In the play, the grand master showed that, through these control strings, the marionette can perform very sophisticated and highly coordinated movements like picking up wine cups, pouring the wine, drinking the wine, and drawing out and holding back the sword from the sleeve.

Control Device

The control device for marionettes, referred to here as the *controller*, is a simple mechanism held by the puppeteer to operate the puppet through a bundle of strings. Strings attached to the puppet are joined to the controller, which is manipulated in order to produce movement in a specific part of the puppet. The design and geometry of the marionette controllers vary for different cultures because of the design and stringing of the marionettes as well as the philosophy of marionette performance.

Western Marionette Controllers

Modern marionettes are controlled by strings hung vertically and use the balance of the tensions in the strings and gravity. But, early in the history of marionettes, string-operated puppets were controlled by horizontally laid strings. Figure 3 illustrates that puppets controlled by horizontal strings performed in the streets of medieval France [11]. An early type of European marionette, called *A La Planchette* in French or *fantoccini* in Italian, was also controlled by horizontal strings [2]. A string was secured to a post at one end and passed through the body of a jointed puppet or a group of puppets. Manipulation of the free end of a string tied to the leg of a minstrel caused movement and action, thereby bringing the puppet to life.

Marionettes with vertical strings came in two forms in 19th century Europe: 1) dramatic marionettes in hybrid rod-string form that were designed to perform in the dramatic repertoire and 2) variety marionettes in full-string form that were mainly for tricks and variety [12]. The vertical control for the all-string marionettes appeared towards the end of 19th century. The control consisted of a vertical grip with a horizontal bar at the bottom to take head strings, hand strings, and other special strings and with a rocking bar at the top for operation of the legs. The vertical control later superseded the horizontal bars [7], [12]. Some rod-string and all-string marionette controllers used by European puppet theaters are illustrated in Figure 4.

The horizontal controller later proliferated in the United States in the early 20th century through European immigrants. Hence, today we see both vertical and horizontal types of marionette controllers in modern puppet troupes. Figure 5(a) and (b) demonstrates typical vertical and horizontal controllers. They are also called airplane controls for the marionettes because their shapes resemble that of

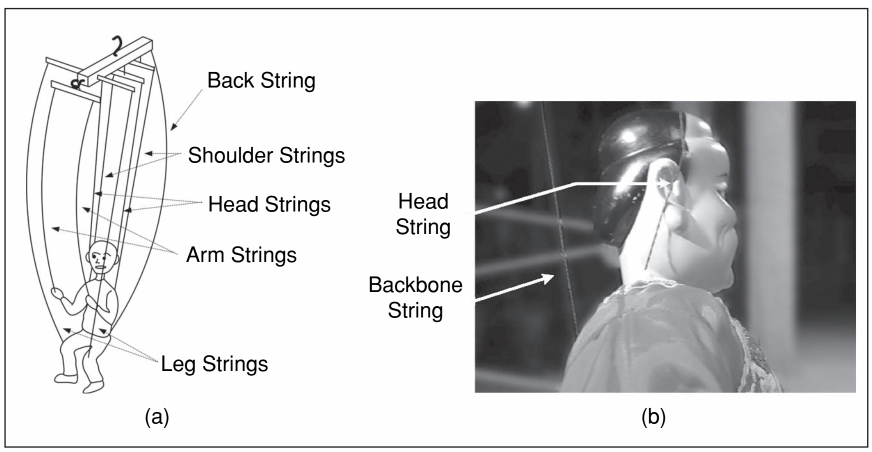


Figure 2. *(a) Functions of strings and (b) head and backbone strings on a traditional Chinese marionette.*

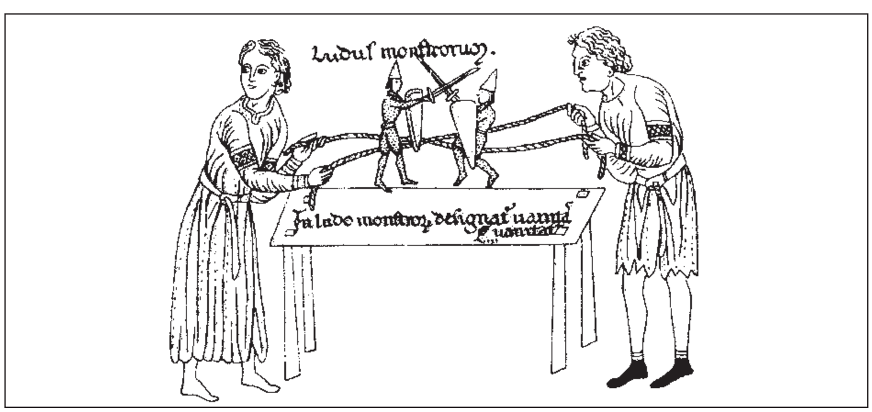


Figure 3. *Puppets on a table controlled by horizontal strings [11].*

The puppet, the puppet control device, and the puppeteer form a complex mechanical system that produces lifelike movement in artificially made entities.

aircraft. Today, most human figures are manipulated through vertical controllers, and animal figures are manipulated using the horizontal counterparts. A variation of the airplane controller is the angle controller, devised by F.H. Bross [6], [7] [Figure 5(c)]. A third type of marionette controller is the paddle controller, invented by W.A. Dwiggins [13] as shown in Figure 5(d). This tiny (a little over six inches) controller is shaped like a sweep-wing aircraft and can make the subtlest movements in a 12-inch-tall puppet. Dwiggins' controller embodies an important principle: the marionette controller should be hand sized or, in modern terms, ergonomically designed. For more variations of the vertical, horizontal, angle, and paddle controllers, refer to [7]. More details regarding the evolution of marionette controllers can be found in [14].

Chinese Marionette Controller

Most of the repertoires in Chinese marionette theaters were adopted from those performed by the traditional theaters, like the Peking Opera. Therefore, the marionette figures had strong connections to their human counterparts, and the design of these figures aimed for realism. And, the number of strings used by Chinese marionettes is usually more than those used by European marionettes. With so many strings to control, Chinese puppeteers made use of a simple controller called the *Gou Pai* (the hooked plate) [15]. The Gou Pai is a plate made from a single piece of wood on which all the strings are carefully arranged in a systematic and symmetrical sequence at the same level [Figure 6(a) and (b)] [16]. The plate is about 25 cm long and 15 cm wide and is attached to a long handlebar to be held by the puppeteer. A hook is nailed to the upper side of the plate so that the controller can hang on a beam on the stage for ease of manipulation. This basic marionette controller design has been passed down for about 1,000 years from the Song Dynasty (Song period: AD 960–1279) [17].

Conventional Marionette Manipulation

Unlike the humanoid robots where all limb joints are fully actuated and controlled, the marionettes are only actuated and controlled by a handful of strings attached to the limbs. Usually, the number of control strings is less than the total

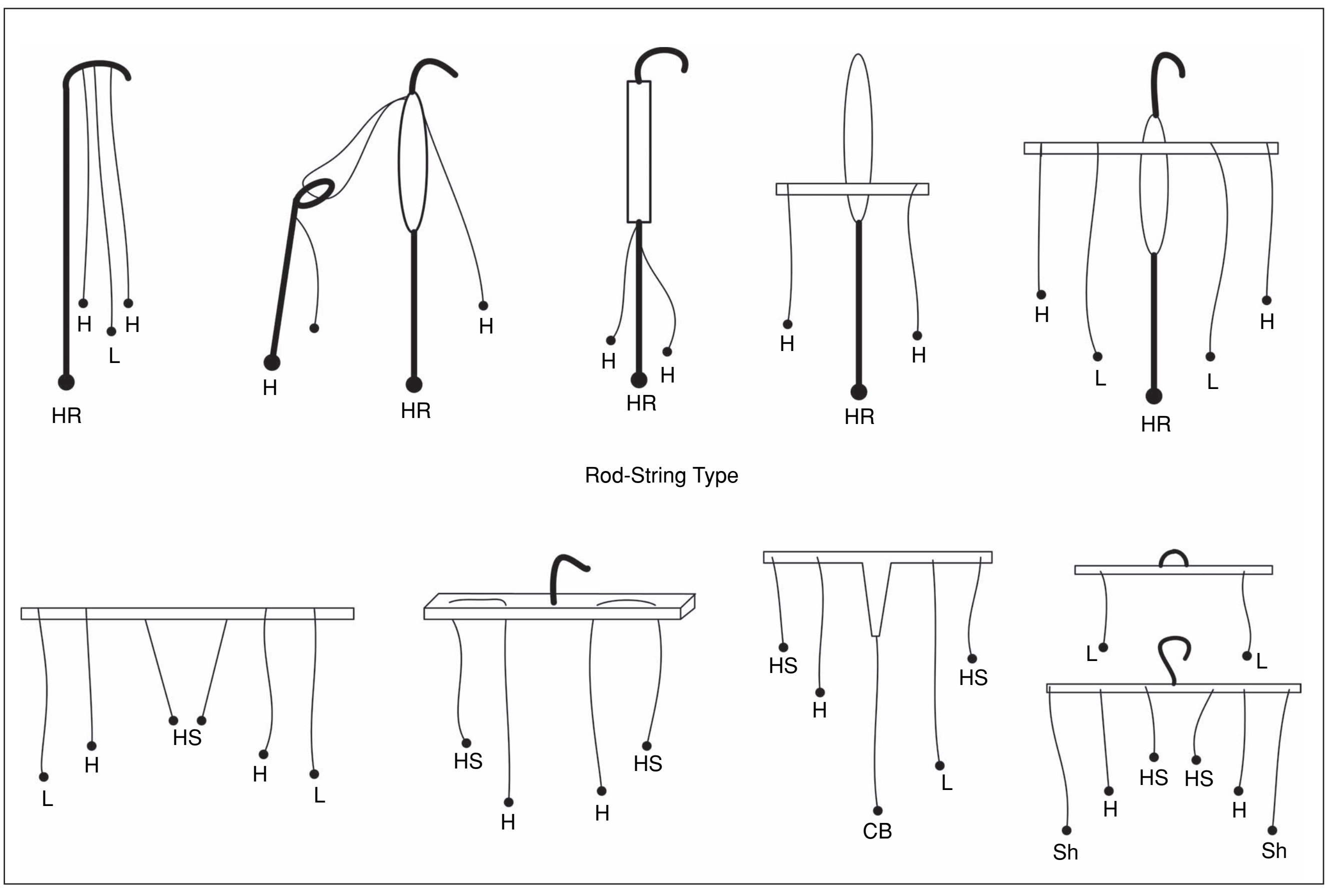


Figure 4. *Various marionette controllers used in 19th century Europe (adapted from [12]). Central balance (CB), head (H), head rod (HR), head string (HS), and shoulder (Sh).*

DOFs of the puppet figure. Assume that the puppet figure consists of n rigid segments (or rigid bodies) and j joints, and the DOF of each joint is f_j. According to the Kutzbach criteria for computing the DOFs of spatial mechanisms [18], the DOF of the puppet figure without any string connections, M, can be obtained by

$$M = 6(n - j) - \sum_{i=1}^{j} f_i. \tag{1}$$

From the perspective of mechanisms, each string connection between the puppet figure and the controller can be considered as a 1-DOF linear actuator that varies only the distance between the two connecting points. Therefore, it is necessary to have M strings to have full control and manipulation of the puppet movements. However, in practice, the number of strings is always less than M, and hence there are uncontrollable DOFs in the marionette. If the number of strings is k, where $k < M$, the uncontrollable DOFs will be $M - k$. The marionette can therefore be described as an underactuated, string-operated, multilimbed mechanism. The configuration of the uncontrollable DOFs will be determined by the gravity or other external factors. Because the marionette is underactuated, manipulation and control of the marionette movement is a very tricky technique. In fact, not only are marionette control and manipulation important in producing lifelike movements, but the locations of the strings on the puppet figure and the center of mass of the limbs also play crucial roles in producing realistic marionettes due to the uncontrollable DOFs.

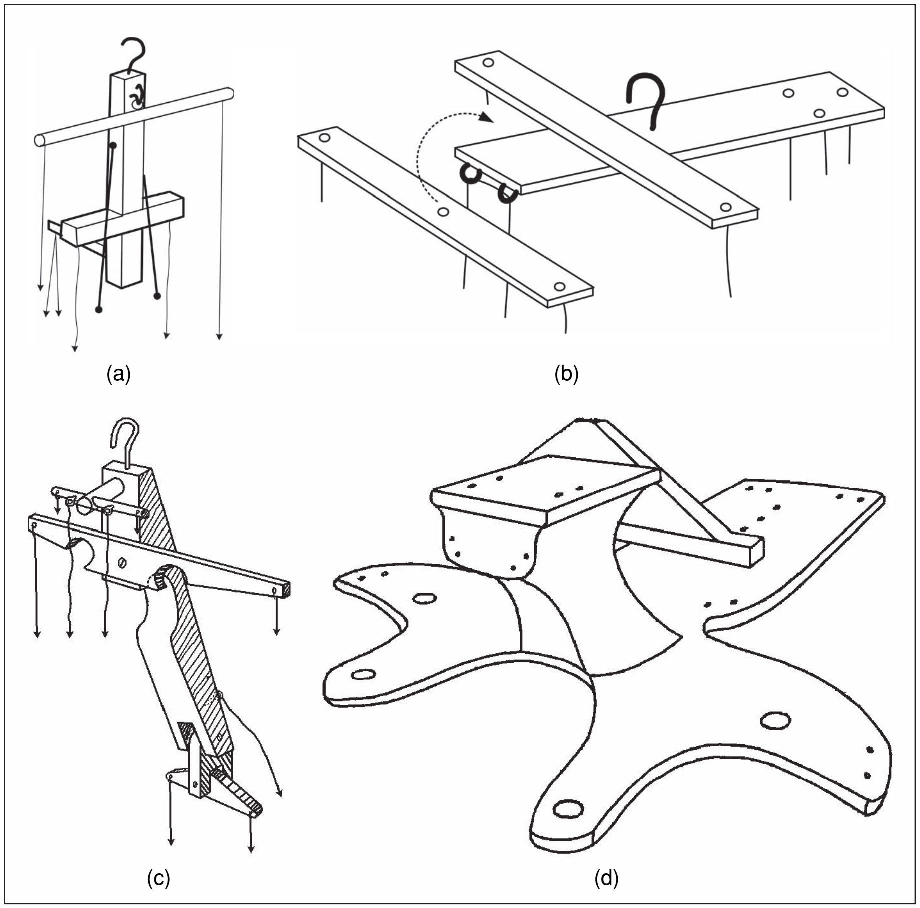


Figure 5. *(a) A vertical controller; (b) horizontal controller; (c) angle controller [8]; and (d) paddle controller [8].*

Manipulation of the marionette through the controller usually is achieved in three ways: pure maneuvering of the controller, plucking the strings, and the combination of both. With a properly designed controller, a wide range of puppet

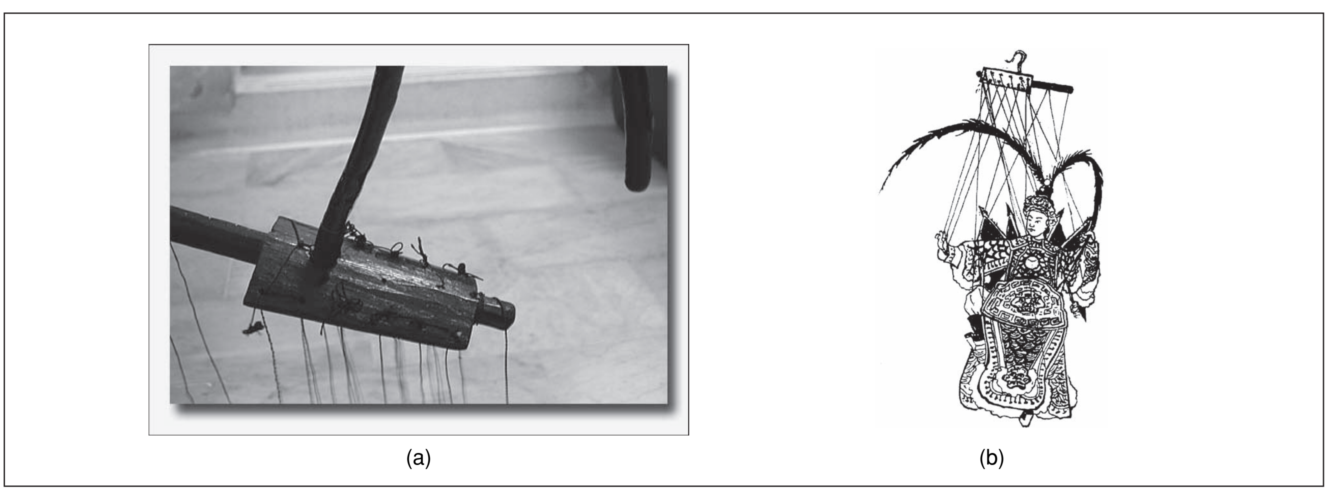


Figure 6. *(a) Chinese Gou Pai and (b) its operation.*

The strings of the marionette allow the puppeteer to remotely control the puppet from a control device.

movement can be achieved by simply tilting or turning the controller. For example, Figure 7(a)–(c) shows how to nod, incline, and turn the head, respectively, by maneuvering the controller [19]. Such maneuvering usually produces fine puppet movement because the string lengths are unchanged. Figure 7(d)–(f) illustrates how to use the combined hand and controller motion to make the puppet bow [19], lift its head, and walk. To bow the body, one needs to tilt the controller and pull the back string [Figure 7(d)]. To keep the head upright while bowing, it is necessary to pull the back string and lower the controller [Figure 7(e)]. In order to make large and prominent gestures—such as walking, stretching the arms, sitting, or kneeling—plucking the strings or pulling the detachable horizontal bars away from the controller is required [Figure 7(f)].

Manipulating the marionette using a Chinese Gou Pai follows similar processes. Figure 8 shows a walking movement sequence using a simple Gou Pai made in our laboratory with only six control strings. Because of the special layout, walking the puppet is achieved by rolling the Gou Pai clockwise and counterclockwise alternately. Generally, sophisticated movements can only be obtained by plucking various strings on the Gou Pai. Due to the large number of strings used in Chinese marionettes, maneuvering the Gou Pai becomes less obvious.

Robotic Marionette System

The original thought on developing a robotic marionette system stems from the study of a tangible information display system that can interact with human beings through a physical agent [20], [21]. It turns out that robotizing a human-figure marionette (or the figure of any other living creature) is the most economical and viable solution for this problem compared to various types of humanoid robots and pet-like robots that are on the market and in laboratories around the world.

The basic robotic marionette system consists of a puppet figure modified from a toy figure; a puppeteer mechanism hosting an array of inexpensive RC servo motors, pulleys, and strings; a motor control network; and a Pentium II 200-Hz PC running a Linux operating system (Figure 9) [22]. The puppet can produce lifelike movements according to programmed motion commands issued from the computer through the puppeteer mechanism. Besides the standard human gestures and motions that a humanoid robot can do, the robotic marionette can defy gravity and fly in the air to perform various stunts.

The main focus of this system is on the development of the robotic puppeteer mechanism. We intend to make it portable and universal. Portability means that the system can be used as a standalone unit or can perform puppet shows side by side with puppeteers using manually operated marionettes. A multiple robotic marionette configuration is achievable by introducing additional multi-DOF puppeteer mechanism units. Universality means that the system adopts a generic and modular design and layout so that different puppet figures can be mounted under the same puppeteer mechanism to perform different shows by simply changing the computer programs. Certain puppet figures with strong cultural, artistic, or historical values can be mounted under the puppeteer mechanism to provide dynamic demonstration instead of static display.

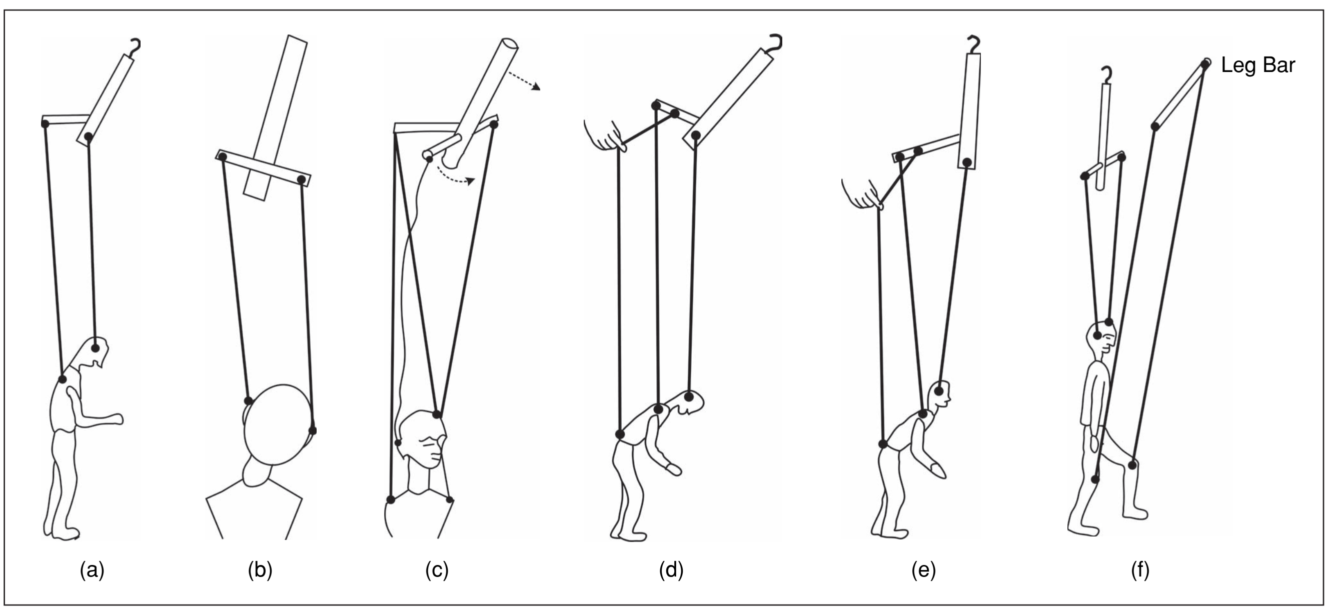


Figure 7. *Manipulating the controller to (a) nod the head; (b) incline the head; (c) turn the head; (d) bow the body; (e) keep the head upright; and (f) walk the puppet (adapted from [19]).*

Altogether, we have developed three robotic marionette systems, ROMS-I, II, and III [23] (Figures 9 and 10). ROMS-I is the first prototype used to explore the pulley-motor concept for puppeteering tasks and the basic structure of the mechatronic system. The puppet figure of ROMS-I is modified from a small wooden human dummy, and fishing wires are used as the control strings. ROMS-II and ROMS-III are improved versions of ROMS-I with inputs from a professional puppet master and were developed simultaneously. The puppet figures of ROMS-II and III are lifelike figures modified from toy soldiers and are covered in fabrics for showing expressive behaviors. Standard embroidery threads for marionettes are used as the control strings. The major difference between the first prototype and the two later versions is in the design and layout of the puppeteer mechanism. The pulleys are mounted horizontally in ROMS-I, whereas in ROMS-II and III, the pulleys are placed in the vertical position. The portability and modularity of the puppeteer mechanism are also taken into consideration. The basic specifications of the three ROMS are listed in Table 1.

Puppeteer Mechanism

Here we focus on the puppeteer mechanism of ROMS-II and III, which becomes our reference design for further development. As we intend to develop the puppeteer mechanism as a generic platform for both Chinese and Western puppet performance, determining the number of strings the mechanism can control is crucial. Based on our study described earlier, the maximum number of strings is set to be approximately 24–30, and each string is to be independently controlled.

String-Retracting Device

As mentioned previously, the strokes of the strings can be large due to the movement of the puppet limbs. The maximum strokes of the strings can be approximately determined based on the working envelopes of the arms and legs. Various concepts have been proposed for the string-retracting device, including a linear actuator, a linkage mechanism assisted pulley system, and a simple pulley-motor assembly [22]. Taking into consideration the stroke of the string, interference of the devices in motion, and the space required, the simple pulley-motor assembly is chosen as the string-retracting device. The pulley-motor unit is designed to be modular. Every string can be easily controlled by one pulley-motor unit so that the string can be lengthened or shortened by simply rotating the motor in opposite directions. The diameter of the pulley is determined by the speed and resolution of the puppet motion. The speed and resolution require a balance between large motion range

Figure 8. *Walking a six-string puppet using self-made Gou Pai.*

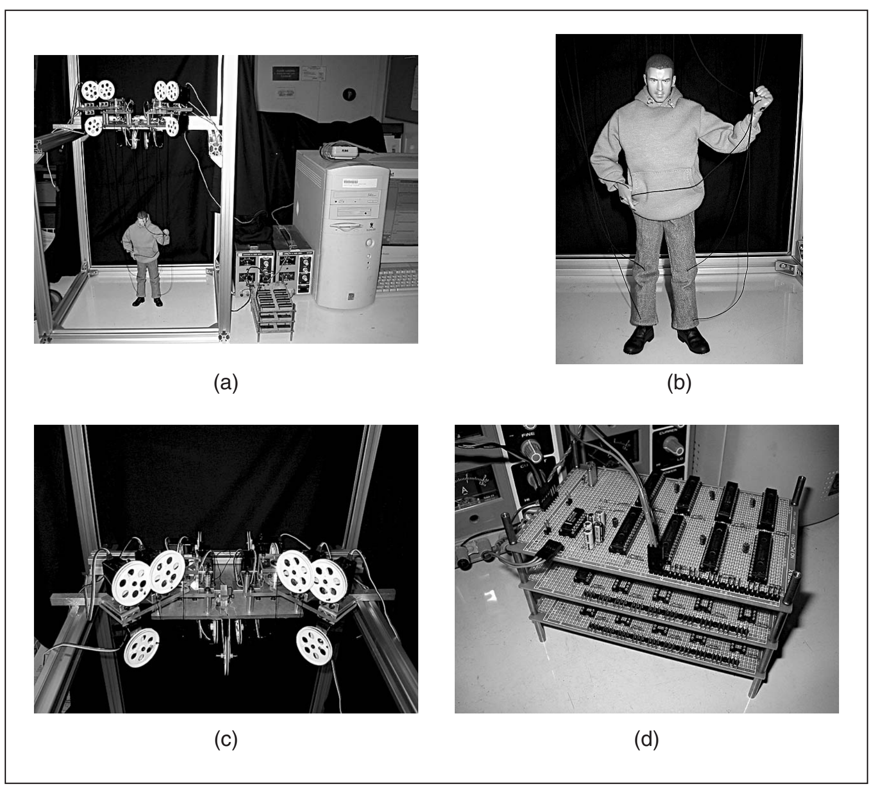


Figure 9. *(a) An overall marionette system (ROMS-II); (b) a puppet figure; (c) the robotic puppeteer mechanism; and (d) the motor control network.*

and fine smooth motion. In addition, the space constraint is to be considered. Based on the size of the puppet figure, the diameter of the pulley used in ROMS-II and III is fixed at 50 mm. The selection of the motor follows similar considerations. The motor chosen for the pulley-motor assembly is the Parallax 900 series continuous rotation servo motors. These motors accept speed control through continuous pulse-width-modulation (PWM) signals at a frequency of 50 Hz.

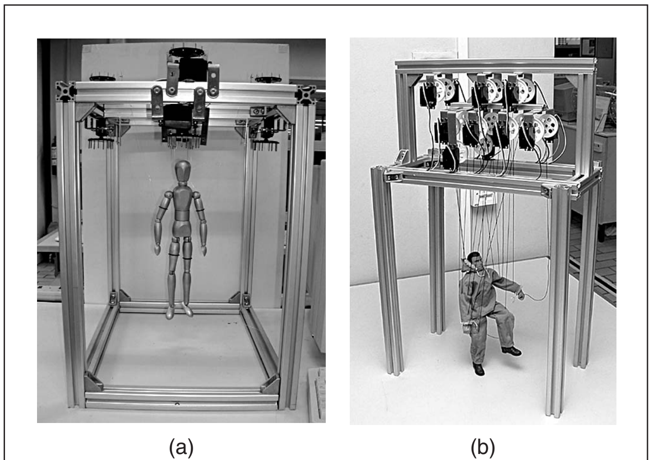


Figure 10. Robotic marionette systems: (a) ROMS-I and (b) ROMS-III.

Mounting Platform with Extra DOF

A mounting platform combining the features of Chinese and Western marionette controllers is designed to house all pulley-motor units. As pulley-motor units can handle large as well as delicate string motions, motor units can remain stationary on the platform, similar to having all the strings being tied to the Gou Pai. For this automated system, it is not necessary to mimic puppet masters' maneuvering of the controller. However, as strings drop vertically under gravity, performing puppet movements along the vertical plane, such as walking, nodding the head, or bowing, is easier than sideways movements, such as opening the arms and side-kicking. Therefore, two swinging armatures that can rotate horizontally are added to the mounting platform of ROMS-II to enable limbs to move sideways [Figure 11(a)]. The movement of the two armatures are controlled by the servo motors. Three pulley-motor units controlling the arm and leg strings are mounted on each armature. In a way, the rotating arms on the mounting platform resemble the horizontal bar of the airplane control, and yet, the extra degrees of freedom introduced on the mounting platform enable the marionette to produce more expressive behaviors [Figure 11(b)].

Table 1. Specifications of ROMS-I, II, and III.

	ROMS-I	ROMS-II	ROMS-III
Motors	8	16	16
Strings	8	14	16
DOF of puppet	30	35	23
Links of puppet	9	11	10
Joint of puppet	8	10	9
Type of joint	8 s-joint (3-DOF)	9 s-joint (3-DOF)	s-joint (3-DOF)
		1 u-joint (2-DOF)	5 r-joint (1-DOF)
Height (cm)	31.2	30.5	29.5
Weight (g)	314	214	259

Motor/String Functions

The assignment of limb movement to the corresponding pulley-motor unit is similar to the traditional marionette control. Among the 16 motors installed on ROMS-II, there are two motors on each arm, one to control the elbow and one to control the hand; three motors on

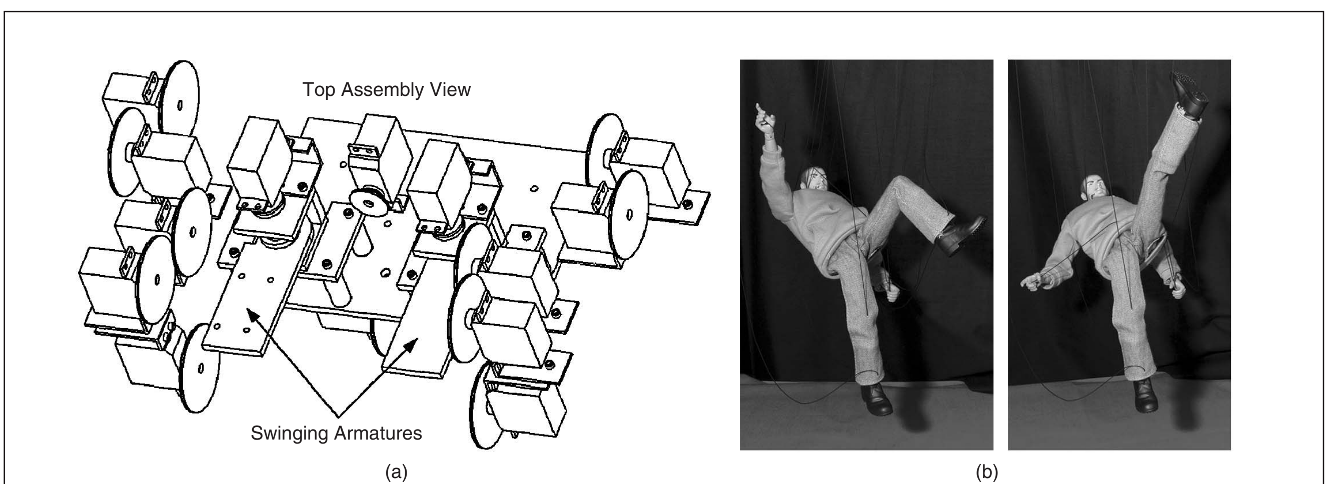


*Figure 11. (a) An exploded view of the ROMS-II platform and (b) puppet poses for movie-like (*The Matrix*) stunts.*

each leg, one each to control the knee, lower leg, and the ankle; two for the head; one for the backbone string; one for the trick string for clapping hands; and two for the swinging armatures. The allocation of motors on ROMS-III is similar to that of ROMS-II, except that the two motors controlling the swinging armatures now control the shoulder of the puppet figure.

Motor Layout Issue

The design of the pulley-motor unit is modular in that each unit can be taken out and assembled onto the platform easily. It is possible to reconfigure the pulley-motor arrangement for marionettes with a different number of strings and different characters. Due to the space and weight constraints of the entire puppeteer mechanism, the motor layout is optimized based on experience. We adopt a layered motor mounting method to keep the footprint of the puppeteer mechanism small. Though all motors turn the pulleys in an upright position, it is possible to stack the pulley-motor units on top of each other without interference. ROMS-II has two motor-mounting layers on and below a single platform, whereas ROMS-III has three motor mounting layers using extended fixtures. With the layered mounting configuration, more pulley-motor units can be added to the puppeteer mechanism for more sophisticated shows [22].

Motor-Control Network

To cope with the large number (up to 30) of pulley-motor units and the reconfigurability of the puppeteer mechanism, we developed a networked servo motor controller from scratch instead of using off-the-shelf products [20], [21]. The networked controller allows the RC servo motor to be easily added or removed from the controller when the configuration of the marionette changes. The networked motor controller consists of 16 PICmicro 16F876 microcontrollers, each controlling one servo motor for position and velocity control. The microcontrollers communicate through a common I2C bus network. One idle microcontroller with no motor attached serves as the bridge between the PC and the motor controller on the I2C bus. The communication between the bridging microcontroller and the PC is through RS-232 serial communication. The microcontroller converts motor-control instructions coming from the PC into corresponding PWM signals to drive the servo motors. The I2C bus is an industrial standard for interchip-level communication with a maximum of 124 devices. It 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. The networked motor controller can incorporate 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 motion behaviors.

Manipulation of the marionette through the controller usually is achieved in three ways: pure maneuvering of the controller, plucking the strings, and the combination of both.

Interface Programming

Low-level servo motor control based on PWM is coded and implemented on the PIC microcontroller using the proprietary assembly language for this class of chips. The firmware development tools include the MPLAB IDE integrated development environment software and the PICSTART Plus programmer from Microchip Corporation.

Puppet Motion Design and Puppeteer Software

One critical issue in the robotic marionette system design is the puppet motion generation. Here we describe how the computer is used to manipulate the puppet through motor commands. We explore three behavior generation methods to produce lifelike and versatile body movements: posture primitives, multilayered motion synthesis, and motion transform. The posture primitives bring out unique features of the puppet according to its basic structure. Combining different posture primitives, we are able to create versatile puppet movements. The primitives also become the fundamental layer—the basic behavior modules—in multilayered motion synthesis. Complex movements are the outcome of flexible combinations of these primitives. The method of motion transform merges the posture primitives and physical human motion data to produce human-like movements on the puppet through imitation learning. All three motion-generation methods are implemented on the PC to command the marionette.

Posture Primitives

Our posture primitive design method takes an agent-centered perspective, which depends on discovering the motor capability of the physical agent. Because the puppet is operated through strings and motorized pulleys, and not directly from the motorized joints, 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 affects the performance of body movements, we only consider motor capability provided by the mechanical structure of the puppet. Based on this consideration, the posture primitives are mostly chosen to be at the extreme positions within the boundary of the workspace. For example, as illustrated in Figure 12, the posture primitives of the puppet

With the layered mounting configuration, more pulley-motor units can be added to the puppeteer mechanism for more sophisticated shows.

arm of ROMS-II can be selected based on the fully lifted position and the fully fall-down position due to stretching the arm string. A transition between the two arm poses also can be chosen as a primitive. Though this is the simplest approach for motor-capability discovery, a large amount of puppet movements can be created due to the structural complexity of the puppet. For a puppet figure with four limbs, with each having only three primitives or innate postures, there are 81 ($= 3^4$) postures that can be formed in total. Excluding mirrored postures with respect to the sagittal plane, there are still 36 distinct postures. Through exploring its motor capability, the robotic puppet can automatically create a large number of posture primitives; enough for many expressive tasks.

Posture primitives based on the mechanical structure of the limb are closely associated with the design of the puppet figure. In principle, these primitives have no direct relationships with human behavior generation. 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.

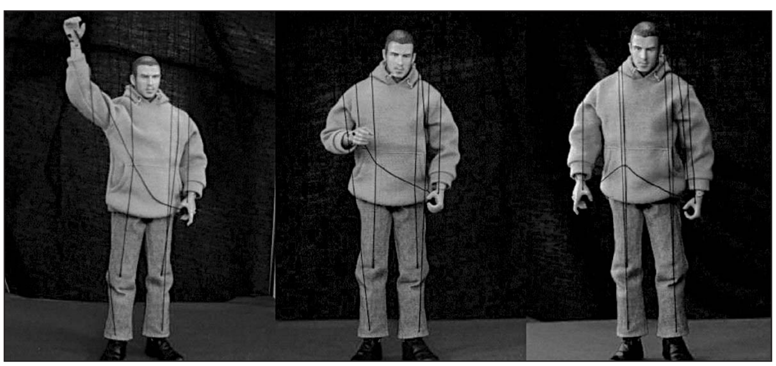


Figure 12. Three posture primitives for the right arm of ROMS-II.

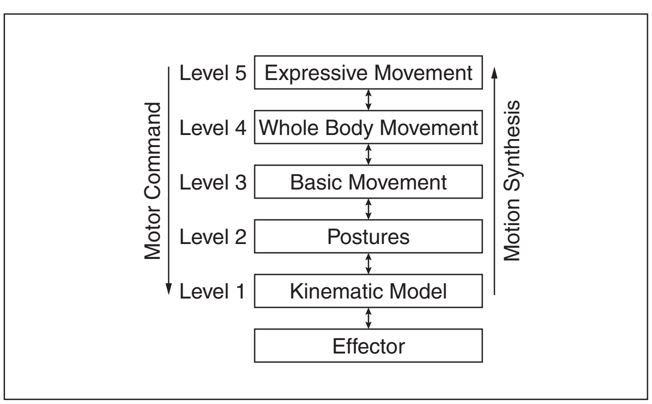


Figure 13. Multilayered motion synthesis for ROMS.

Multilayered Motion Synthesis

Puppeteers spend many years to study and explore puppet's behaviors. They learn the puppet movements through extensive trial-and-error, guided by experiences and intuitions. This motion exploration process is a mixture of heuristics and rule-based methods. Nevertheless, this kind of approach is very difficult to implement on the robotic puppet for two reasons: 1) operators of the robotic marionette usually are inexperienced puppeteers, and 2) robotic marionette devices are fundamentally different from the manual controller. The overwhelming computation effort of using a 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 multilayer behavior structure to address the computation complexity problem.

The spatial characteristics in the behavior of the robotic puppet lead to full-body movement in space while the movement is proceeding along the timeline. 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 a five-layer structure (Figure 13). The lowest "effector layer" refers to the mechanics part of the robotic puppet, which is not part of the five-layer motion synthesis process. 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 is a bottom-up approach. As their building blocks, lower level behavior units provide services to higher level movements, while movement instructions descending from higher levels are executed.

Kinematic Model Layer

This 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 feed-forward effect of the internal model reduces the delay of feedback loop and amount of communication between layers of the hierarchical control structure [24]. Similarly, the behavior modules in this layer keep the kinematics models of the associated body parts of the puppet figure and execute inverse kinematics calculations.

Postures Layer

The second layer is represented with the discrete and static postures obtained from examining the motor capability of corresponding body parts described earlier. These postures are the innate postures reflecting the mechanical properties of the robotic puppet. Postures for expression may be assigned by the human operators with any mean.

Basic Movement Layer

Human movements can be categorized into discrete and continuous movements [25]. Arm reaching and camera-taking poses are examples of discrete movements. One common type of continuous movement is the rhythmical or oscillatory movement, such as walking, swimming, and waving hands. This layer implements point-to-point discrete movements and continuous oscillatory movements for limbs.

- Discrete puppet movements are generated based on the posture primitives. A discrete movement is represented as a state transition from a starting posture to a goal posture within a given amount of duration. Via points obtained through the interpolation between the starting and destination postures are used to smooth the motion [21].
- The rhythmic puppet motion is produced based on prescribed periodical functions on the puppet joint displacement because of ease of description. The trajectory of joint angles is first produced using a periodic function. Along the timeline of the periodic movement, each transient arm pose is treated as a low-level posture primitive, and the required string length is calculated through inverse kinematics. For small amplitude oscillation, the string lengths during the movement can be obtained through linear interpolation without using an inverse kinematics calculation all the time. Figure 14 shows that the arm is making a periodic movement vertically, in a range of 30°, by periodically changing the length of the string.

Full-Body Movement Layer

This layer coordinates basic movement modules on different body parts to produce harmonious full-body movement. The primitive acting on part of the puppet figure usually employs a small set of links and strings. For example, waving an arm only needs one string and the links associated with the arm. The behavior modules created from a disassembled figure cannot guarantee coordinated full-body movement unless the distributed structure is organized systematically. Coordination requires adding constraints on the body structure in order to associate different movements and body parts. Our coordination method takes the functional perspective to Cohen's theory [26], [27] about movement involving multiple DOFs over the full body. By comparing and analyzing animals with different evolutionary histories, we see that full-body movements performable for a human or the animal are constrained by the body's morphology. 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. This forms the fourth layer of the behavior structure of the robotic puppet, synchronizing 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 symmetry in the body structure and movement reduces the amount of information required to specify the motion states of the DOFs spread over the entire body. The five behavior modules are

- the radial module, which makes all limbs retract towards or stretch from the center of the body (Figure 15)
- the waving module, which generates a swaying movement on the body by moving each limb rhythmically (Figure 16); the motion instruction for a single limb is repeatedly applied to each limb under synchronization of a periodic function
- the segment module, which divides the body into the upper and the lower part; the arms of the upper body are constrained to a single movement unit and the legs of the lower body are treated identically (Figure 17)
- the ipsilateral module, which groups the left arm and

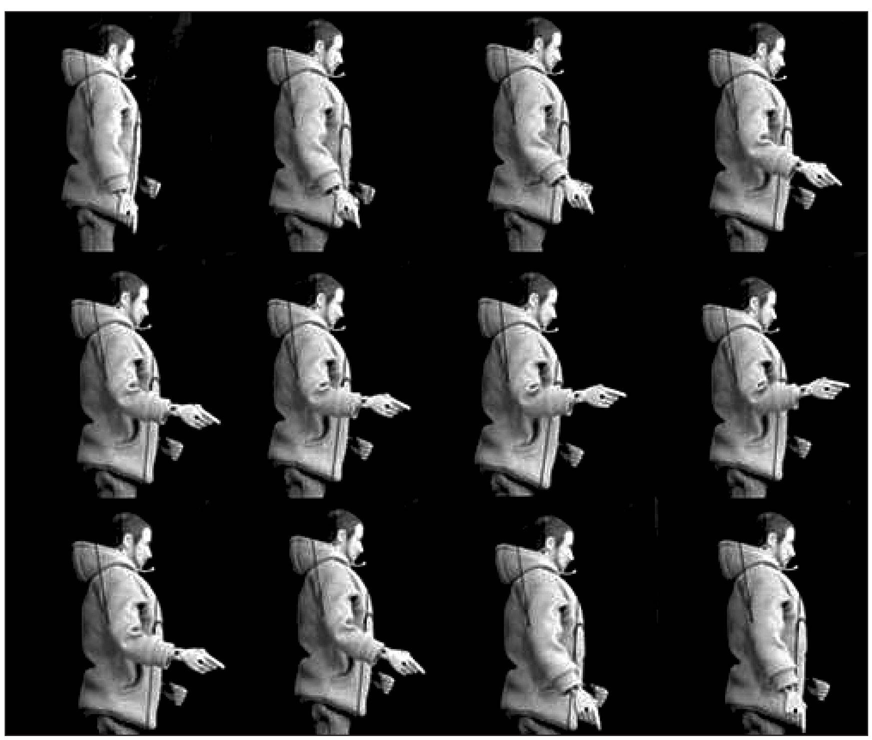


Figure 14. The arm moves in rhythmic motion.

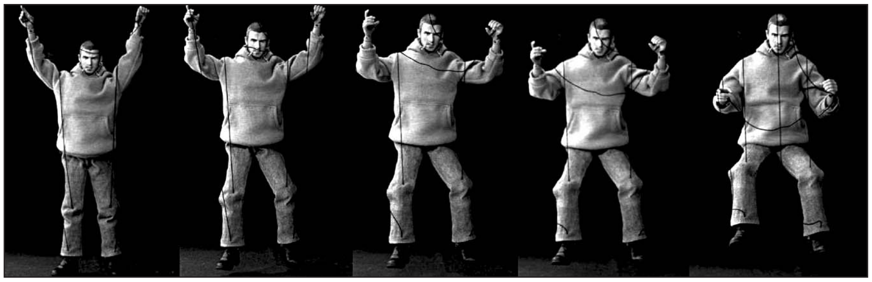


Figure 15. *The limbs move towards the center of the body.*

Figure 16. *One arm precedes the other.*

Figure 17. *The independent movements of the upper and lower body parts.*

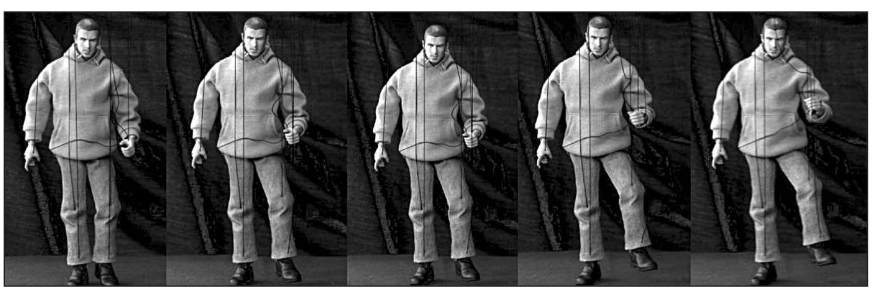


Figure 18. *The arm and the leg on the same side move together.*

Figure 19. *The arm and the leg on the opposite side move together.*

the left leg as one movement unit and the right arm and the right leg as the other (Figure 18)

- the cross module, which groups the left arm and the right leg as one movement unit and the right arm and the left leg as the other (Figure 19).

Expressive Movement Layer

This layer produces expression movement for the robotic puppet. Simple expressive movement is generated through the motion transformation method described in the following.

The multilayered structure of the behavior representation provides flexibility for designers to take different engineering methods of motor control. To a certain degree, the behavioral design process is independent from the underlying mechanical structure. This gives behavioral design considerable portability, which is a salient feature for expressive physical agents made into a highly diversified body structure.

Motion Transform on Robotic Puppet

The human-like appearance of the robotic puppet evokes people's expectations of human-like behaviors. Body movements with human traits are necessary for the expression task of the robotic puppet—not only because they produce the illusion of life but also because many fundamental movements are exclusively used by humans. The body movements of humans have distinctive patterns due to many factors, such as the physical properties of body structures, psychological states, and cultural backgrounds. The behaviors made up from motor primitives of the robotic puppet exhibit specific properties of the mechanical structure. Usually they do not have the features of human behaviors nor are they human-like. There exists human motion data obtained from real-time human motion capture, computer simulation, and animation. They can be used as plentiful resources to implement human-like movements on the robotic puppet. Following the imitative learning mechanisms of humans and animals, we implement motion transform from human motion data to motion sequences on the robotic puppet in two steps. The first is to acquire new primitives from the displayed motion sequences, and the second 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 target model robotic puppet truly learn the behaviors of the source model, but merely mimic the behaviors superficially. As the imitative learning is not sensitive to the differences on the body structures of the source and the target models, the source model may be a real person, a simulated one, or even an animated character. In our practice, we choose simulated human data as the source model due to its ease of access.

As 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 will be 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 result of the motion transformation through imitative learning based on the primitives is shown in Figure 20. The puppet movement is

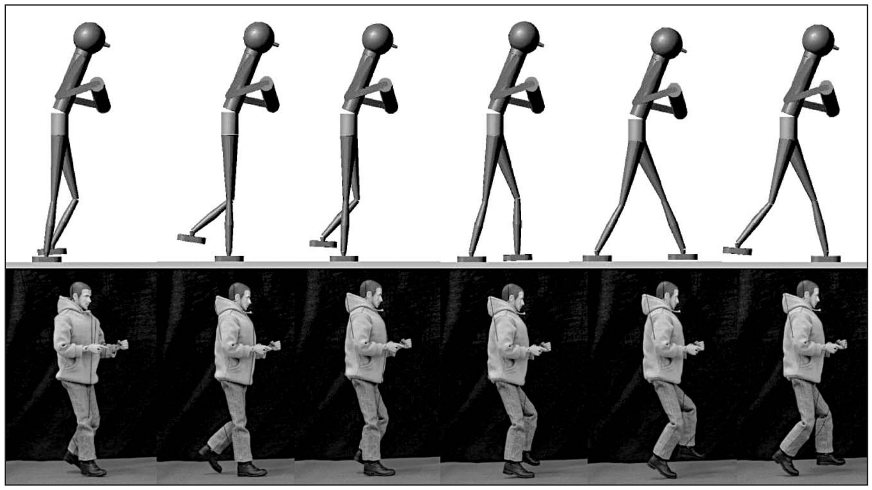


Figure 20. *Motion transfer: ROMS-II puppet follows the simulated human figure.*

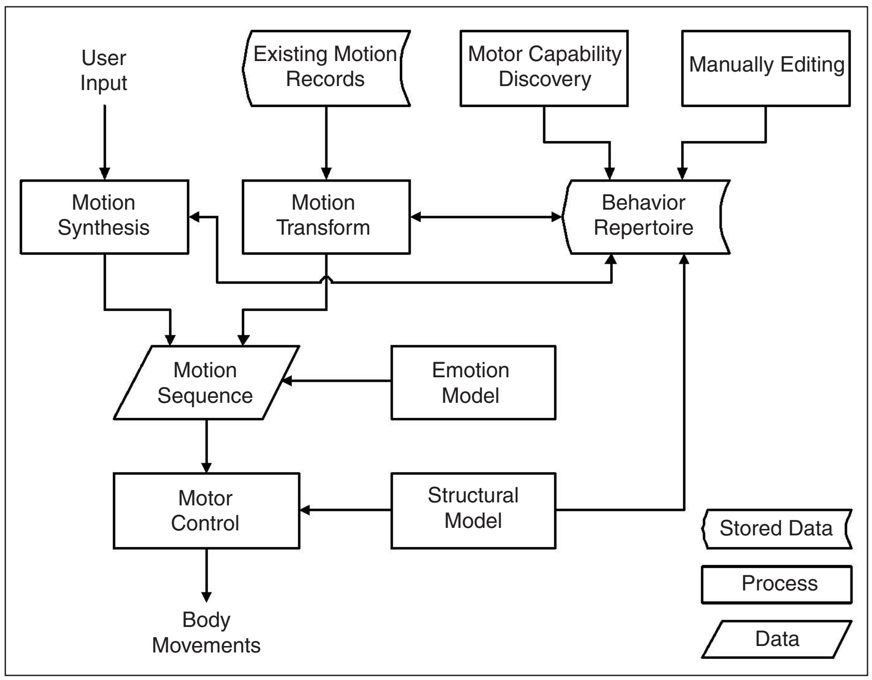


Figure 21. *Puppeteer software structure.*

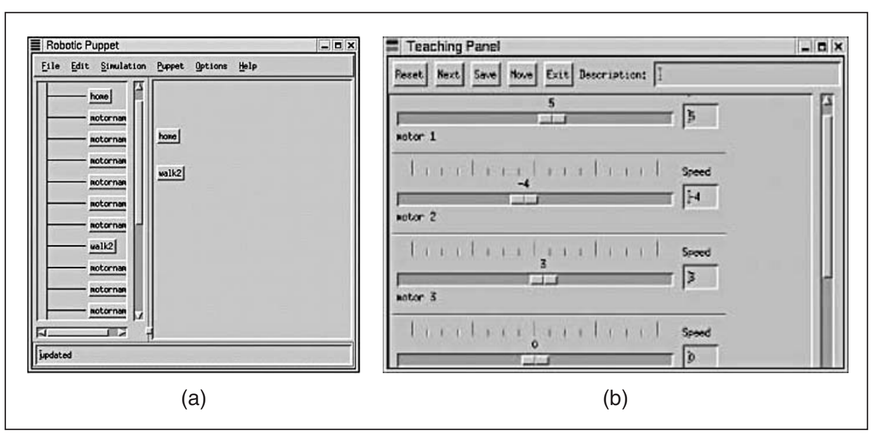


Figure 22. *GUI of Puppeteer software: (a) behavior editing program and (b) teaching panel.*

not a perfect mapping to the source movement of the simulated character. As one can observe, there is some mismatch between the simulated figure and the puppet in the last two postures of Figure 20. The motion transformation from the simulated character to the robotic puppet relates to the correspondence problem. There are obvious differences between the body structures of the simulated character and the robotic puppet. The puppet has to transform the postures represented in joint angles into string lengths to drive the motors. Because this transformation is not strictly linear, there are differences in the corresponding movements. However, the motion sequence acquired by the puppet could be regarded as a walking behavior of the original simulated figure [21].

Puppeteer Software

As the marionette is mechanically controlled by an array of pulley-motor units, coordination of pulley-motor actions to create marionette movements is achieved through Puppeteer, a high-level marionette motion-generation program [21]. The Puppeteer software developed on the Linux PC for the robotic marionette provides an interface to accept motion-control instructions from the user. The user has three ways to input and edit the robotic marionette movements based on the three methods mentioned previously: posture primitives, motion synthesis, and motion transformation (Figure 21). Different motion-generation methods give the user some selections of varied workload in order to make the motion input process comfortable and effective. The primitives are simple marionette gestures currently hand-coded into the program and stored in the behavior repertoire, such as lifting the arm or stretching the leg. The user can manually edit and combine the primitives in the behavior repertoire to produce motion sequences or drive the puppet with existing motion sequences obtained from motion transformation. The marionette movement can also be generated automatically through motion synthesis. The major modules of the puppeteer software include:

- the graphic user interface (GUI) for primitive setting and motion editing
- the internal structural model of the robotic puppet
- layered representation of behaviors
- motion transformation
- motion synthesis
- motion output and motor control
- motor speed calibration.

Except for the motor-control module, most parts of the software were developed in C language with GCC 2.2 on a PC running Redhat 6.1 Linux OS. The GUIs of the behavior editing program and the teaching panel are implemented in X/Motif 2.1 [Figure 22(a)

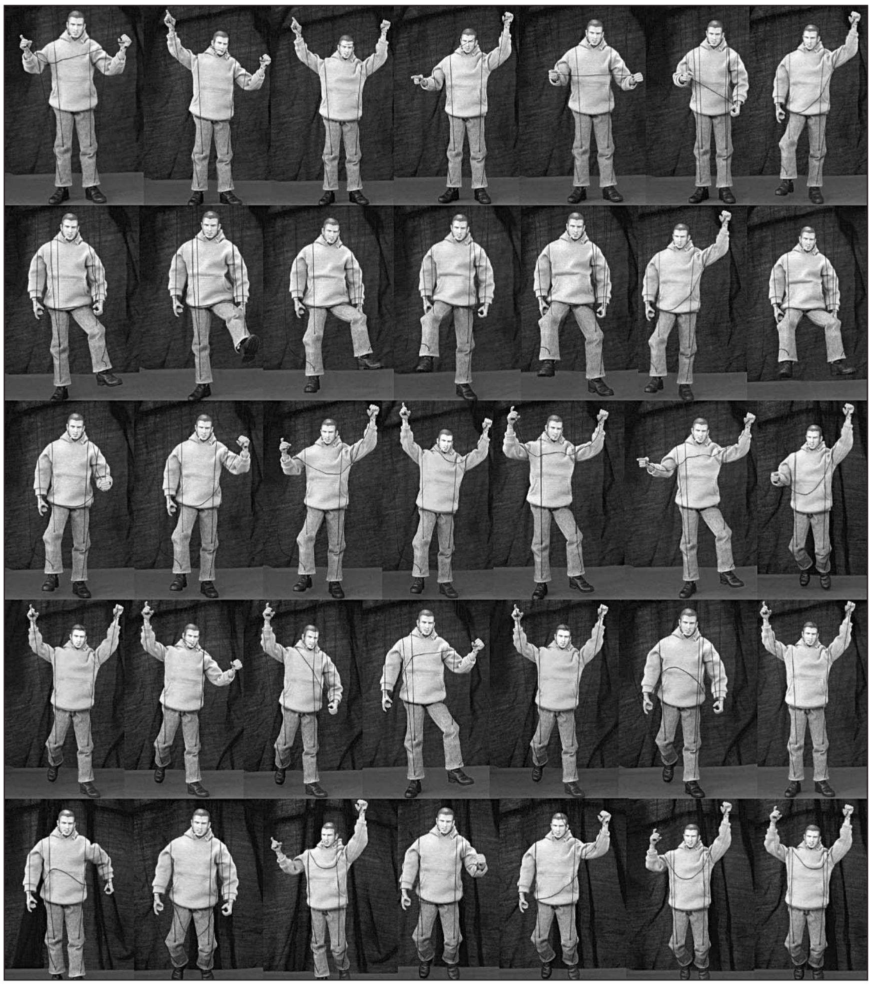


Figure 23. *ROMS-II puppet postures.*

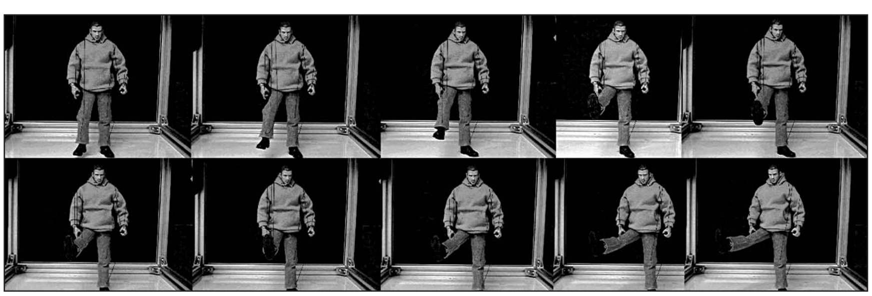


Figure 24. *Puppet postures obtained from combining primitives.*

and (b)]. Currently, the Puppeteer software can be used to control a single robotic puppet to produce various live or prerecorded motion sequences. It is suitable for the dynamic display of marionettes. At this stage, it does not possess the capability to edit and complete an entire puppet show, which requires knowledge of puppet theatre production: script writing, choreography, background music, and directorship.

Demonstration of Robotic Marionette Behaviors

The visual results of the robotic puppet engaging behavioral expressions are captured in a number of postures and a short motion sequence. The postures shown in Figure 23 are directly obtained by combining existing posture primitives. They are a small part of the behaviors the robotic puppet can perform. When 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 24), the puppet figure first lifts up its left leg, extends it to the extreme, and then moves it to the left side. Several motors and strings are involved in this coordinated movement that is built on the basic behavior modules. This motion sequence demonstrates that complex behaviors could be generated and presented on the robotic puppet with the aid of the behavior editing software.

Due to the limitation of the current Puppeteer software, we can only produce short motion sequences for the robotic marionette. Interested readers can find video footage of these marionette motions on our Web site [29]. It is not possible to program the puppet to complete an entire puppet show automatically through programming. However, with its current capability, it is possible to use the robotic marionette along with human puppeteers as part of the show and new features.

Audience Response

Direct feedback from the collaborating puppet master on the capability of the robotic marionette is that the pros and cons of ROMS ideally compliment those of the traditional puppeteer. ROMS can control delicate marionette movement that a human operator may find very difficult to achieve manually. As the human hand has limited capacity, the robotic marionette may produce some movement that cannot be achieved using the manual marionette controller. For example, the stunt of "motion freeze" in midair shown in the movie, *The Matrix* was hand-coded into ROMS-II by one of our students [Figure 11(b)] to demonstrate that our marionette can "fly." However, ROMS does have drawbacks, mainly in its dynamic behavior. When the string motion becomes very drastic, the marionette may swing in undesirable directions, frequently in addition to its controlled movement [28]. This is fundamentally similar to the pendulum effect that needs to be compensated for with more advanced control algorithms. For human puppeteers, this kind of effect can be minimized as the human has multimodal sensory feedback, such as force sensing and visual feedback.

Conclusion

Marionette puppetry, a disappearing art form in Singapore and other countries, is a universal cultural theme. The puppeteer techniques involved become part of the cultural or national heritage; the making of marionette figures becomes fine folk art. In modern elementary education, puppetry is also a very effective tool for shaping children's personality, creativity, and thinking. This work examined the basic marionette techniques (design, anatomy, and manipulation) from an engineering perspective. Through the development of robotic marionette systems, ROMS-I, II, and III, and the marionette motion-generation methods, we would like to explore the possibility of infusing modern robotic and mechatronic technology into the traditional art form of marionette performance, to evoke and stimulate public interests in this art form, and to provide the puppetry and puppet theatre new elements and features. As pointed out by the puppet master we consulted, the range and features of robotic manipulation can indeed compliment the traditional marionette due to its system characteristics, and eventually this kind of technology infusion may stand side by side with traditional puppetry. In addition to performing arts, we would also like to explore the possibility of using robotic marionettes for multidisciplinary education, since the nature of the system is a mixture of robotics and performing arts. The knowledge of robotics, various aspects of performing arts, ergonomics, and biology becomes indispensable in the development and application of this system. How these fundamental subjects are related to the tasks of the system development and educational use becomes an important issue. Lastly, lessons learned from the robotic marionette systems can provide insight for research in cable-driven multiple rigid body systems and other areas like artificial intelligence, media, and design technology.

Acknowledgments

The authors appreciate the invaluable comments and suggestions on puppeteer techniques provided by Ms. Beng Tian Tan of The Finger Players (http://www.fingerplayers.com), Singapore, and the encouragement from Prof. M. Ceccarelli. Efforts made by other members in this project, S. K. Tan, Ronald Yeu, Stefan Künzler, Wei Ji, and Wesley Chia, are also appreciated.

Keywords

Robotic marionette system, multilayered motion synthesis, mechatronic, puppet, string motion.

References

[1] D. Currell, *Puppets and Puppet Theatre*. Wiltshire, UK: Crowood Press, 1999.

[2] D.E. Hodges, *Marionettes and String Puppets Collector's Reference Guide*. Norfolk, VA: Antique Trader, 1998.

[3] K.D. Sun, *Origin of Chinese Puppetry* (Chinese). Shanghai, China: Shanghai Publishing, 1952.
[4] C. Flower and A. Fortney*, Puppets: Methods and Materials*. Worcester, MA: Davis, 1983.
[5] G. Latshaw, *The Complete Book of Puppetry*. New York: Dover, 1978.
[6] O. Batek, *Il Teatro Delle Marionette*. Milan, Italy: Ottaviano, 1981.
[7] L. Coad, *Marionette Sourcebook*. Vancouver, Canada: Charlemagne, 1993.
[8] B. Frascone and D. Frascone, *The Art and Technique of Marionette Making*, vol. 1. Pommeraye, France: Franscone, 2002.
[9] Q. Liu and S.L. Jiang, *Chinese Puppetry Art* (Chinese). Beijing, China: China World Language, 1993.
[10] Y.Q. Huang, *Drunken Zhong Kui* [Video]. Quanzhou Puppet Theater, Quanzhou, China.
[11] M. Violette, "History of French puppetry," in *Proc. Int. Puppet Theatre Conf.*, Taipei, Taiwan, 1999, pp. 234–261.
[12] J. McCormick and B. Pratasik, *Popular Puppet Theatre in Europe, 1800–1914.* Cambridge, U.K.: Cambridge Univ. Press, 1998.
[13] D. Abbe, *The Dwiggins Marionette*s. New York: Harry N. Abrams, 1970.
[14] I.-M. Chen, R. Tay, S.S. Xing, and S.H. Yeo, "Marionette: From traditional manipulation to robotic manipulation," in *Proc. Int. Symp. History of Machines and Mechanisms*, M. Ceccarelli, Ed., Cassino, Italy, pp. 119–133, 2004.
[15] Q.Y. Yang, *History of Puppet Theater in Hong Kong* (Chinese). Hong Kong: Cosmos Books, Ltd., 2001.
[16] P. Tso, *Puppet Theatre in Hong Kong and Their Origins*. Hong Kong: Hong Kong Municipal Government, 1987.
[17] Anonymous, *Children Playing with String Puppets*. Song Dynasty painting, National Palace Musuem, Taiwan.
[18] J.E. Shigley and J.J. Uickers, *Theory of Machines and Mechanisms*, 2nd ed. New York: McGraw-Hill, 1995.
[19] D. Currell, *The Complete Book of Puppetry*. New York: Pitman, 1974.
[20] S.S. Xing and I.-M. Chen, "Design expressive behaviors for robot puppets," in *Proc. 7th Int. Conf. Control, Automation, Robotics, Vision*, Singapore, 2002, pp. 378–383.
[21] S.S. Xing, "Behavior-based physical agents as information display devices," Ph.D. dissertation, School of Mech. Prod. Eng., Nanyang Technological Univ., Singapore, 2004.
[22] B.K. Tay, "Development of a Robotic Marionette System," Final Project Rep., School Mech. Prod. Eng., Nanyang Technological Univ., 2003.
[23] S. Künzler, "Development of Programmable Puppeteer Mechanism for Robotic Marionette Theatre," diploma thesis, School Mech. Prod. Eng., Nanyang Technological Univ., Singapore, and Hochschule Rapperswil, Switzerland, 2003.
[24] M. Kawato, "Internal models for motor control and trajectory planning," *Current Opinion Neurobiology*, vol. 9, no. 6, pp. 718–727, 1999.
[25] R.A. Schmidt and T. D. Lee, *Motor Control and Learning: A Behavioral Emphasis*, 3rd ed. Champaign, IL: Human Kinetics, 1999.
[26] T. Nakata, T. Mori, and T. Sato, "Analysis of impression of robot bodily expression," *J. Robotics Mechatronics*, vol. 14, no. 1, pp. 27–36, 2002.
[27] P. Hackney, *Making Connections—Total Body Integration Through Barteneiff Fundamentals*. New York: Gordon and Breach Science, 1998.
[28] K. Yamane, J.K. Hodgins, and H.B. Brown, "Controlling a marionette with human motion capture data," in *Proc. IEEE Int. Conf. Robotics Automation*, Taipei, Taiwan, 2003, pp. 3834–3841.
[29] Modular Robotics and Robot Location. [Online] Available: http://155. 69.254.10/users/risc/www/enter-intro.html

I-Ming Chen received the B.S. degree from National Taiwan University in 1986 and M.S. and Ph.D. degrees from California Institute of Technology, Pasadena, California, in 1989 and 1994, respectively. He is currently an associate professor in the School of Mechanical and Production Engineering of Nanyang Technological University in Singapore. In 1999 he was JSPS Visiting Scholar in Kyoto University, Japan, and currently is Fellow of Singapore-MIT Alliance under the Innovation in Manufacturing Systems and Technology (IMST) Program. He is also an adjunct professor of Xian Jiao Tong University, China. His research interests are in reconfigurable automation, biomedical applications of reconfigurable robotic systems, parallel kinematics machines (PKM), biomorphic underwater robots, and smart material based actuators. He is now serving on the editorial board of *IEEE/ASME Transactions on Mechatronics*. He was the program cochair of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) in Hawaii in 2001, and the program cochair of the 7th International Conference on Control, Automation, Robotics and Vision, Singapore, in December 2002. He is Member of IEEE and ASME, was chair of the Singapore Chapter of the IEEE Robotics and Automation Society 2001–2003, and is the chair of Prototyping for the Robotics and Automation Technical Committee under the IEEE Robotics and Automation Society.

Shusong Xing received the B.E. degree from Tianjin University Metallurgical College, China, in 1992, and the M.E. degree from Nankai University in 1997. He has been a Ph.D. research student in the School of Mechanical and Production Engineering of Nanyang Technological University in Singapore since 1999. His research interests are in behavior-based robotics, action generation on the physical agents, and human machine interaction. He has a special interest in the application of motor-control theory developed by cognitive scientists to the physical agents with an understanding of the inherent constraints of mechatronic systems.

Raymond Tay received the bachelor of engineering degree in mechanical engineering (mechatronics specialization) from Nanyang Technological University in 2003. His research interest are in machine design, mechatronics, and puppetry.

Song Huat Yeo received his B.Sc. and Ph.D. degrees in mechanical engineering from the University of Birmingham, UK, in 1983 and 1987, respectively. He joined Nanyang Technological University in Singapore in 1992 and is currently an associate professor in the School of Mechanical and Production Engineering. His research interests are in synthesis of mechanisms, cable-driven mechanisms, wearable haptic devices, kinematics of modular reconfigurable robots, the mechanics of gripping, and gripper design.

Address for Correspondence: I-Ming Chen, School of Mechanical and Production Engineering, Nanyang Technological University, 50 Nanyang Ave, Singapore 639798. Phone: +65 67906203. Fax: +65 67911859. E-mail: michen@ntu.edu.sg.