# Design and Analysis of Some Nonanthropomorphic, Biologically Inspired Robots: An Overview

**Gregory S. Chirikjian**
*Department of Mechanical Engineering*
*Johns Hopkins University*
*Baltimore, Maryland*
*e-mail: gregc@jhu.edu*

Received 16 August 2001; accepted 16 August 2001

In this paper, two kinds of biologically inspired robots under investigation at Johns Hopkins University are reviewed. While these designs are reminiscent of designs in nature, they are, however, not anthropomorphic. Rather, they are zoomorphic in macroscopic structure. These two kinds of robots are snakelike "hyper-redundant" manipulators, and amoeboid "metamorphic" robots. In addition to issues in the design of the robots, issues in the dynamics and motion planning of these devices are reviewed.

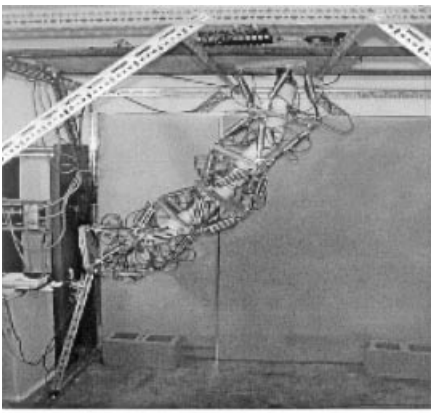

## 1. INTRODUCTION

Most serial robotic manipulator arms are reminiscent in structure of the human arm. For example, one refers to the shoulder, elbow, and wrist of the arm. In contrast, closed-loop kinematic structures, which have become popular in the kinematics literature in the past few years (see, for example, ref. 38) do not always have such clear analogs in nature. In this paper, we examine neither of these cases. Instead, the much less commonly studied example of very high-degree-of-freedom (dof) architectures developed by the author and others in previous work is reviewed, along with the corresponding mathematical modeling techniques.

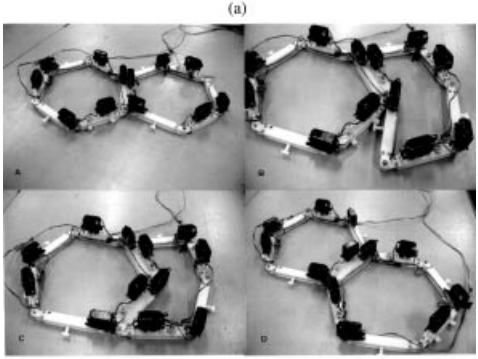
In particular, two kinds of robots are reviewed: (1) snakelike "hyper-redundant" manipulators, and (2) amoeboid "metamorphic" robots. These two concepts are illustrated in Figure 1. In practice, the manipulator in Figure 1(a) (which is binary-actuated) would be equipped with an end effector, and fixtures in the workspace onto which the end effector could lock. The metamorphic robot in Figure 1(b) would consist of many more than two units, and hence be able to "flow" through small openings and reconfigure itself as task requirements demand.

If one divides robotics problems roughly into two categories—the first being problems arising in well-structured environments, and the second being those arising in unstructured environments—the two kinds of robots reviewed here represent potential solutions to problems in the second domain. That is, manipulators and mobile robots with the ability to radically change geometry, either by bending and writhing, or

Journal of Robotic Systems 18(12), 701–713 (2001)



(a)



(b)

**Figure 1.** (a) A 36-dof discretely-actuated hyper-redundant manipulator; (b) two metamorphic robot modules.

by changing topology, provide the opportunity for the device to conform to an environment that may change in geometry in unforeseen ways. Of course, this flexibility in function and form must be weighed against the financial cost and increased possibility of actuator failure in such complex devices. For instance, it probably does not make sense to design such robots for use as fixed automation, but 20–50 years in the future, these kinds of devices may be optimal for operations such as search and rescue in collapsed buildings, adaptive self-healing space platforms, and microprosthetic or surgical devices that enter a patient through a small incision, self-locomote, and change shape to perform their function with minimal tissue damage.

Throughout this paper the progress that has been made to date on these kinds of systems is reviewed, and potential stumbling blocks are enumerated. The

format of the remainder of the paper is as follows: In Section 2, issues in the design, implementation, and analysis of snake-like robots are examined. In Section 3, the corresponding issues are addressed for metamorphic robots. Throughout the paper, current work by other research groups is examined in these research areas. Predictions regarding future directions are presented in Section 4, which summarizes the paper.

## 2. HYPER-REDUNDANT MANIPULATORS

hyper-redundant manipulators have a very large number of actuatable degrees of freedom. Applications of snakelike hyper-redundant manipulators include inspection in highly constrained environments, tentacle-like grasping of objects, and whole-arm manipulation. These are illustrated in Figure 2.

Computationally efficient modeling of the system kinematics and dynamics is necessary for hyper-redundant manipulators to be used effectively. In past work the author, together with Prof. J. Burdick, developed an efficient framework for the kinematics and motion planning of hyper-redundant manipulators and mobile robots.[5,9,10,11,12] More recently, the author extended this approach to include manipulator dynamics and control,[7] as well as to manipulators with discrete states.[8] A continuous curve (or "continuum") approximation that captures the manipulator's macroscopic geometric features is useful in these contexts. Subsections 2.2 and 2.3 review this framework, and also provides some new insights. But first, issues in the design of hyper-redundant manipulators are discussed in the following subsection.

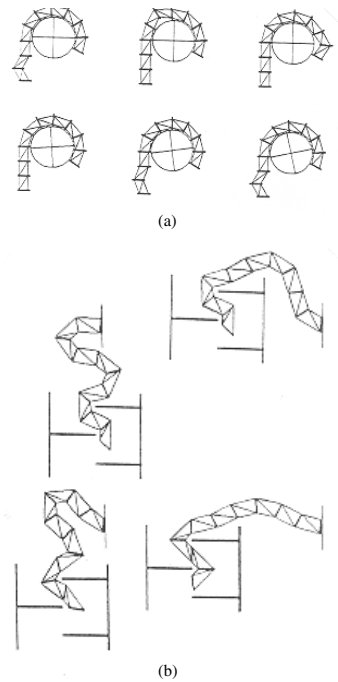



**Figure 2.** Some applications of hyper-redundant manipulator arms: (a) Obstacle avoidance; (b) Wavelike grasping and whole-arm manipulation.

### 2.1. Design Issues

By definition, a hyper-redundant manipulator is one that has a very large number of degrees of freedom. Hence, problems of wiring, mechanical error propagation, actuator/sensor failure, and weight must all be addressed. With standard multiplexing techniques, wiring problems are reduced. Mechanical error propagation and payload are not major problems when the design consists of a cascade of parallel platforms. And hyperredundancy actually can provide better fault tolerance with respect to actuator failure, because the manipulator can still function well even if several actuators fail and are locked in position.

The author has previously studied two very different kinds of hyper-redundant manipulators: those with continuous motion actuators, and binary manipulators. Other interesting research in snakelike robot design and planning in the context of actuators with continuous motion includes the pioneering work of Anderson and Horn,[1] Hirose,[26,27] and Morecki,[39] as well as the more recent works of Lumelsky,[49] Lenarčič,[37] Kobayashi,[31] Choset,[25] Ostrowski,[45] and Mochiyama.[41] Particularly, interest in variable-geometry-truss (VGT) structures have been studied extensively by Hughes,[44] Hamlin and Sanderson,[24] Salerno and Reinholtz,[52] Tavakkoli and Dhande,[54] and Etemadi-Zanganeh and Angeles.[18] Studies on elastic tubelike structures used for manipulation include the works of Wilson[57] and Suzumori, Iikura, and Tanaka.[53]

Work in the area of discretely-actuated manipulators goes back to the 1960s at Stanford with the work of Roth, Rastegar, and Scheinman,[50] and Pieper.[48] Other work in this area includes that of Koliskor.[33] Most recently, an interest in active control of the pneumatic cylinders used in discretely actuated manipulators has also developed. Namely, Waldron[56] and Bobrow[2] have investigated the use of pneumatic cylinders as force elements in robots.

In the context of discrete actuation, an arm with no feedback control is only intended for use in discrete tasks such as pick-and-place and spot welding. In this scenario, a discretely-actuated arm will generally not be able to reach a specified point exactly, and the best configuration of the arm is defined as the one that minimizes a measure of distance, or a metric, between the desired and actual end-effector frame. Such metrics are examined in refs. 15, 19, 20, 30, 40, and 47.

One criticism of the use of binary actuation comes from the perception that two-state actuation is necessarily jerky. In this subsection, it is shown that by appropriate mechanical design, this need not be the case.

Consider the simplest and most inexpensive possible binary actuator. It consists of a pneumatic cylinder with an internal plunger. Air inlet and outlet are present at opposite ends of the tube. This flow is channeled in a binary way with a solenoid valve. It is assumed that the seal between the plunger and cylinder may leak, although only at a very slow rate compared to the rate at which air enters or exits the tube through the portals.

If the hollow cylinder is fixed in position and orientation, and the mass of the plunger is $m$, then in the absence of gravity the equation of motion for the cylinder is simply:

$$m\frac{d^2x}{dt^2} = F - c\frac{dx}{dt} \tag{1}$$

where $F = PA$ is the force, $P$ is the pressure in the tube, $A$ is the area of the plunger head, $c$ is a viscous friction coefficient, and $x$ is the amount the plunger is displaced from a datum. (In our designs, a viscous dashpot is placed in parallel with one or more pneumatic cylinders to provide a very large damping.)

Much can be learned from this simple model. For instance, if the datum is chosen such that $x(0) = 0$, and the viscosity is very high relative to the mass, then $m/c \to 0$ and we get

$$x(t) \approx \frac{F}{c}t \tag{2}$$

An elementary analysis of the more complicated case of a whole manipulator composed of highly damped pneumatic actuators yields essentially the same result.[8]

Since pneumatic cylinders have a very high force-to-weight ratio, and damping can be made as high as is required by appropriate choice of dashpot fluid, viewing $m/c$ as small is not unreasonable. Though energy dissipated due to damping might be considered wasteful, it is not necessarily more wasteful than energy dissipated due to resistive heating in motor coils and the power electronics used for control of continuous motion actuators.

The steady-state speed is adjusted by either changing the viscosity (changing the lubrication in the cylinder and/or the restrictiveness of the airflow) or changing the pressure $P$.

Currently, in our designs there is no active control of pressure ( except regulation to the set pressure), and there is no active change being made to the viscosity (as can be the case when using electrorheological fluids). However, these are opportunities for future research.

In order to avoid wear and tear and prevent high frequency transients, one can insert short springs (or rubber bumpers) in the cylinder at both ends. This helps to ramp the speed up to steady state when the actuator is starting from rest, and allows for gradual deceleration at the other end. These effects operate over a small portion of the total stroke.

Figure 1 illustrates a binary manipulator designed in the author's laboratory.[17] In this design, parallel (platform) mechanisms are cascaded to form a macroscopically serial structure (i.e., one that appears serial when viewed from a distance). If one considers that muscle tissue is composed of binary force elements (muscle fibers) arranged in massively parallel arrays, then this binary-actuation concept can be viewed as biologically inspired.[56]

The kinematic design problem for such a manipulator reduces to the question of how kinematic parameters should be chosen so that the distribution of frames reachable by the end of the arm meets some desired task requirements. For instance, if great accuracy is needed in a particular region of the workspace, then greater *density* of frames may be needed. In a series of papers, the author and his students have explored computational methods to rapidly generate the density distribution of frames,[13] as well as solving the mathematical inverse problems arising in the kinematic design of manipulators for desired density.[36] This requires the use of group-theoretical methods,[36] which have natural

extensions to fields of study outside of the robotics area.[35]

### 2.2. Inverse Kinematics and Motion Planning

When considering continuously-actuated hyper-redundant manipulators, the issue of coordinating the degrees of freedom to perform a specified task naturally arises. The continuum approach to the resolution of this hyperredundancy is based on a two-step modeling and computation procedure.

In the first step, we assume that, regardless of the mechanical implementation (e.g., serial-revolute, VGT, rubber-actuator, etc.), the hyper-redundant robot can be modeled by a continuous *backbone curve* that captures the robot's macroscopic geometric features (Figure 3). A backbone curve parameterization and an associated set of reference frames that evolve along the curve are collectively referred to as the *backbone reference set*. In this paradigm, hyperredundancy resolution is reduced to the determination of the proper time-varying behavior of the backbone reference set.

Points, $\mathbf{x}(s,t)$, on a spatial backbone curve can be parameterized as follows:

$$\mathbf{x}(s,t) = \int_0^s l(\sigma,t)\mathbf{u}_1(\sigma,t)\,d\sigma \tag{3}$$

where $s$ parameterizes the backbone curve, and $t$ is time. $\mathbf{u}(s,t)$ is the unit vector tangent to the curve at $s$. We may write

$$l(s,t) = 1 + \epsilon(s,t) > 0$$

where $\epsilon(s,t)$ is the *local extensibility* of the curve (and hence the manipulator) at point $s$ and time $t$. $\epsilon < 0$ indicates a local contraction, whereas $\epsilon > 0$ corresponds to a local expansion. The backbone curve arc-length between the backbone curve base ($s = 0$) and any point along its length is:

$$L(s,t) = \int_0^s l(\sigma,t)\,d\sigma \tag{4}$$

Any parameterization of the unit sphere can be used to parameterize $\mathbf{u}_1(s,t)$. In the planar case we have

$$\mathbf{u}_1(s,t) = \begin{pmatrix} \cos\theta(s,t) \\ \sin\theta(s,t) \end{pmatrix}$$

where

$$\theta(s,t) = \int_0^s \kappa(\sigma,t) l(\sigma,t)\,d\sigma$$

is the counterclockwise measured angle that the tangent vector makes with the $x_1$ axis, and $\kappa$ is the curvature function for the curve (with counterclockwise bending measured as positive).

A *backbone reference frame* at $s$ has right-handed orthonormal basis vectors, $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$, and its origin coincides with point $\mathbf{x}$. The set of backbone frames can be parameterized as:

$$\mathbf{Q} = (\mathbf{u}_1 \;\; \mathbf{u}_2 \;\; \mathbf{u}_3) \in SO(3) \tag{5}$$

where $\mathbf{Q}(0,t) = \mathbf{1}$.

The backbone reference set can also be expressed as a parameterized set of homogeneous transformation matrices:

$$\mathcal{H} = \begin{pmatrix} \mathbf{Q}(s,t) & \mathbf{x}(s,t) \\ \mathbf{0}^T & 1 \end{pmatrix} \tag{6}$$

where $\mathbf{x}(\cdot)$ and $\mathbf{Q}(\cdot)$ are as defined in Eqs. (3) and (5).

Consider the $i$th of $n$ modules (platforms) in a hyper-redundant arm. Attach a frame, $\{F_{i-1}\}$, to the "input," or base, of the module, and a frame, $\{F_i\}$, to the "output," or top, of the module. For the manipulator configuration to conform to the continuous curve geometry, the frames $\{F_{i-1}\}$ and $\{F_i\}$ are chosen to coincide with the backbone reference frames at a set of $n+1$ "fitting" points: $\{s_i\}$. We typically choose $s_i = i/n$ for $i = 0, \ldots, n$. Recall that equal partitioning of the curve parameter need not imply equal physical spacing along the curve, because $L(\cdot)$ can be chosen from a broad class of functions.

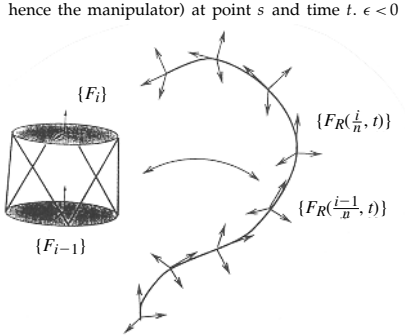



**Figure 3.** The backbone curve model.

The $4 \times 4$ homogeneous transform relating $\{F_i\}$ to $\{F_{i-1}\}$ is denoted by $\mathbf{H}_{i-1}^{i}$. This consists of the relative translation, $\mathbf{r}_{i-1}^{i}$, and rotation, $\mathbf{R}_{i-1}^{i}$, of $\{F_i\}$ with respect to $\{F_{i-1}\}$, as measured in $\{F_{i-1}\}$.

$\mathbf{q}^{M_i} \in \mathbb{R}^6$ is the vector of joint displacements that determine the geometry of the $i$th module. It is assumed that the inverse kinematics of the module, which relates $\{F_i\}$ to $\{F_{i-1}\}$, can be solved in a closed or numerically efficient form.

The manipulator configuration will exactly conform to the backbone reference set at points $\{s_i\}$ if

$$\mathbf{H}_{i-1}^{i}\left(\mathbf{q}^{M_i}(t)\right) = \mathcal{H}^{-1}(s_{i-1}, t)\mathcal{H}(s_i, t) \tag{7}$$

where $\mathcal{H}(s, t)$ is as defined in Eq. (6). That is, the right-hand side of (7) expresses the relative displacement of the backbone curve reference frame at $s_i$ with respect to the backbone curve reference frame at $s_{i-1}$, whereas the left-hand side describes the relative displacement of the $i$th module output frame with respect to its input frame. When the two are equated, the manipulator aligns exactly with the continuous backbone curve at the $n$ fitting points. We typically choose one of the fitting points to be the end-effector frame, so that position and orientation of the distal ends of both the continuous backbone curve and the manipulator are in exact alignment. An example of this method is given in ref. 11.

The computational burden of the fitting procedure may be distributed over $n$ processors to result in $\mathcal{O}(1)$, or *constant*, time. This is generally not true for other kinds of redundancy resolution techniques.

### 2.2.1. Quantifying Backbone Curve Optimality

Deformation of the backbone curve (and the resulting change in hyper-redundant manipulator configuration) results from mechanism bending, twisting, rolling, and extension/contraction at each $s$. Hence, it makes sense to consider backbone curves that change as little as possible relative to a home (reference) state. A dimensionally consistent cost function that, when minimized, achieves these effects is

$$I = \frac{1}{2}\int_0^1 [\mathrm{tr}(\dot{\mathbf{Q}}\mathbf{W}\dot{\mathbf{Q}}^T) + \beta(\dot{L} - 1)^2]\, ds \tag{8}$$

where $\mathrm{tr}(\mathbf{A})$ denotes the trace of matrix $\mathbf{A}$ and the superior dot represents differentiation with respect to $s$. The cost function and constraints are functions of time, but since we are extremizing from point to point in time, the calculus of variations formulated for a single dependent variable is directly applicable. $\mathrm{tr}(\dot{\mathbf{Q}}\mathbf{W}\dot{\mathbf{Q}}^T)$ is a measure of mechanism bending and twisting. $\mathbf{W}$ is a $3 \times 3$ symmetric positive semidefinite weighting matrix. We make the reasonable assumption that there is no preferred direction of bending, and hereafter $\mathbf{W}$ is restricted to the isotropic form $\mathbf{W} = \alpha\mathbf{1}$, where $\mathbf{1}$ is the $3 \times 3$ identity matrix. Similarly, $(\dot{L} - 1)^2$ is a measure of a mechanism's extension and contraction from its nominal length. Thus, $\alpha > 0$ weights the relative cost of bending, twisting, and rolling, whereas $\beta > 0$ weights extension/contraction.

At $s = 0$, the backbone reference frame must coincide with the base frame. At $s = 1$, the backbone reference frame must correspond to the desired end-effector orientation, $\mathbf{Q}_D$. Thus, the boundary conditions:

$$\mathbf{Q}(0, t) = \mathbf{1}; \quad \mathbf{Q}(1, t) = \mathbf{Q}_D(t) \tag{9}$$

are imposed on the Euler-Lagrange equations. The minimum bending problem can be stated as the minimization of Eq. (8) subject to these boundary conditions and the end-effector position constraint $\mathbf{x}(1, t) = \mathbf{x}_D(t)$. See refs. 5 and 11 for explicit examples.

### 2.2.2. Using Waves in Obstacle Avoidance, Locomotion, and Grasping

In the context of locomotion and grasping, the continuum model provides a natural way to describe and implement the kind of peristaltic motion that is so common in nature. In particular, the curvature and extensibility of the backbone curve that defines the shape of a hyper-redundant arm undergoing such waves is of the form of a traveling wave:

$$\kappa(s, t) = \phi_1(s - a(t))$$
$$\epsilon(s, t) = \phi_2(s - a(t))$$

An example of a situation in which this is useful is illustrated in Figure 2(b). Here a circular object is being reoriented by a sequence of pure bending waves, ($\epsilon(s, t) = 0$). First, a wave forms by having part of the manipulator contract in a section that is not in contact with the object. Simultaneously with this, part of the arm in contact with the object straightens, providing some slack, so that the position of the object remains constant. In the second sequence in Figure 2(b), the wave travels along the length of the arm, resulting in an extended grasp over the object. The process repeats, with the amount of straightening of a segment of the arm balancing the amount of coverage gained by the locomotion wave. The arm thus undergoes a cyclic change in shape, while the

object being grasped is rotated in finite increments. The benefit of this formulation is that the grasp is statically stable and very robust while the object is being manipulated.

It is not difficult to see that this was inspired by the locomotion gaits of caterpillars and snakes.[12] What is perhaps less obvious is that the same kind of curvature waves may be used to describe the path of a hyper-redundant arm to circumvent obstacles in a known environment.[9] This is illustrated in Figure 2(a).

### 2.3. Continuum Dynamics

In addition to use as a tool in kinematics and motion planning, the backbone curve model can be used to capture the inertial properties of a hyper-redundant arm. Hence, the static and dynamic loads on actuators due to the inertia of the manipulator can be approximated by observing the loads due to an imaginary backbone curve with mass density, $\rho(s)$. The benefit of this approach is that the inverse dynamics problem (i.e., the problem of finding actuator torques and forces that produce a desired motion) is achieved in $\mathcal{O}(n)$ computations that can be evenly distributed over $n$ parallel computers. Hence, the inverse dynamics of a hyper-redundant manipulator can be calculated in $\mathcal{O}(1)$ time. This is in comparison to the now-standard method of iterative Newton-Euler dynamics, which is by nature an $\mathcal{O}(n)$ serial computation.[16]

The momentum balance equations provided by continuum mechanics take on a particular form when combined with the backbone model. Namely:

$$\frac{d}{dt}\int_{\sigma}^{1}\rho(s)\frac{\partial \mathbf{x}}{\partial t}(s,t)\,ds = \mathbf{F}(\sigma,t) + \int_{\sigma}^{1}(\mathbf{t}+\rho\mathbf{b})\,ds \quad (10)$$

Equation (10) corresponds to a force balance in a free-body diagram resulting from an imaginary cut made normal to the backbone curve at the point at which $s = \sigma$. This is just Newton's laws for a continuous filament. The vector $\mathbf{F}(\sigma,t)$ is the internal force transmitted to the distal end of the manipulator ($s \in [\sigma, 1]$) by the lower end of the manipulator ($s \in [0, \sigma]$). The left-hand side of the equation represents inertial forces due to the motion of the curve, $\mathbf{F}$ is the reaction force at the imaginary cut, $\mathbf{t}$ represents surface tractions such as friction, and $\rho\mathbf{b}$ are body forces such as gravity.

The angular-momentum balance equations provided by continuum mechanics also have a special form for the case of hyper-redundant manipulator backbone curves:

$$\begin{aligned}\frac{d}{dt}\int_{\sigma}^{1}\mathbf{x}(s,t)\times\rho(s)\frac{\partial \mathbf{x}}{\partial t}(s,t)\,ds &= \mathbf{M}(\sigma,t) + \mathbf{x}(\sigma,t)\times\mathbf{F}(\sigma,t) \\ &\quad + \int_{\sigma}^{1}\mathbf{x}(s,t)\times(\mathbf{t}+\rho\mathbf{b})\,ds \end{aligned} \quad (11)$$

Again referring to the imaginary cut made normal to the backbone curve at the point at which $s = \sigma$, the vector $\mathbf{M}(\sigma,t)$ is the internal moment transmitted to the distal end of the manipulator.

Equations (10) and (11) furnish all the tools needed to compute hyper-redundant manipulator dynamics.

In order to make use of the continuum model, there must be a way to transfer the dynamical information from the continuum model to the actual physical manipulator under consideration. In broad terms, "projecting" the dynamics of the continuum model onto the actual manipulator is achieved by again making an imaginary cut in the continuum model. Now, however, the forces and moments at the cut will be matched with those in the actual hyper-redundant structure at corresponding locations along the length of the manipulator. Inertial forces, body forces, and surface tractions accumulated from the distal end of the manipulator to the cross-section under investigation are approximated using the backbone curve model. The resulting reaction forces are calculated in the physical structure at the imaginary cutting plane. For example, the rules of structural analysis are used when considering the forces on a variable-geometry truss. For manipulators with a macroscopic serial structure, the imaginary cutting planes are located at the interface between links or modules. Therefore,

$$\frac{d}{dt}\int_{\frac{i}{n}}^{1}\rho(s)\frac{\partial \mathbf{x}}{\partial t}(s,t)\,ds - \int_{\frac{i}{n}}^{1}(\mathbf{t}+\rho\mathbf{b})\,ds = \mathbf{F}_i \quad (12)$$

and

$$\begin{aligned}\frac{d}{dt}\int_{\frac{i}{n}}^{1}\mathbf{x}(s,t)\times\rho(s)\frac{\partial \mathbf{x}}{\partial t}(s,t)\,ds - \int_{\frac{i}{n}}^{1}\mathbf{x}(s,t)\times(\mathbf{t}+\rho\mathbf{b})\,ds \\ - \mathbf{x}(i/n,t)\times\mathbf{F}_i = \mathbf{M}_i \end{aligned} \quad (13)$$

where $\mathbf{F}_i$ and $\mathbf{M}_i$ are the continuum approximation of the force and moment exerted by the $i$th module (or link) on the $i+1$st module of a hyper-redundant manipulator.

Each of the preceding integrals can be evaluated separately for $i \in [0, \ldots, n-1]$, and so the dynamics problem can be evenly distributed over $n$ processors. The key to this approach is the continuum model, and the assumption that the actual manipulator does not stray from the curve. Without these assumptions, serial computations would have to be performed and a Newton-Euler style algorithm would result. With the continuum model, closed-form solutions or quadrature approximations to the integrals can be computed in many cases, and so there is no need for iteration.

Assuming that the inertial forces, body forces, and surface tractions computed from the continuum model are representative of the actual manipulator, the reactions present in the manipulator structure at the $i$th module are equated to the preceding quantities. It is then simply a matter of matching forces in the actual structure to those generated from the continuum model in Eqs. (12) and (13). This is called projecting the dynamics of the continuum model onto the physical structure.

In general, the resulting forces in the actuators are found separately for each module by solving a matrix equation of the form

$$\mathbf{K}_i \tau_i = \$_i$$

where $\mathbf{K}_i$ is a matrix that depends on the kinematics of module $i$, $\tau_i$ is a vector of generalized actuator forces in module $i$, and $\$_i$ is the wrench $(\mathbf{F}_i, \mathbf{M}_i)$ that the continuum model estimates is the applied load on the $i$th module. See ref. 7 for an implementation of this idea for the particular example of a variable-geometry truss, and a numerical comparison of the continuum model with a Lagrangian lumped-mass model of manipulator dynamics.

## 3. METAMORPHIC ROBOTS

A *metamorphic* robotic system[6] is a collection of independently controlled units, or modules, each of which has the ability to connect, disconnect, and climb over adjacent modules. Each module allows power and information to flow through itself and to its neighbors. A change in the metamorphic robot topology (i.e., a change in the relative way modules within the collection are connected) results from the locomotion of each module over its neighbors.

Metamorphic systems can be viewed as a large swarm (or colony) of connected robots that collectively act as a single entity. The difference between metamorphic systems and other reconfigurable robots found in the literature is that they possess all of the following properties:

1. All modules have the same physical structure, including the same hardware for computation and communication.
2. The mechanical structure of the modules possesses symmetries that ensure that they fill planar and spatial regions with minimal gaps and form a lattice.
3. The modules must each be kinematically sufficient with respect to the task of locomotion; that is, they must have sufficient degrees of freedom to be able to "walk" over adjacent modules, so that they can reconfigure without outside help.
4. Modules must adhere to adjacent modules, for example, there must be electromechanical or electromagnetic connectors between modules, which can carry load. This causes the collection of modules to act as a single physical object.

The idea of a metamorphic robotic system differs from related concepts presented in the literature. Three types of modular reconfigurable robotic systems have been proposed in the literature: (1) robots in which modules are reconfigured using external intervention (see, for example, refs. 3 and 31); (2) cellular robotic systems in which a heterogeneous collection of independent specialized modules are coordinated (see, for example, the pioneering work of Fukuda[21]); (3) swarm intelligence in which there are generally no physical connections between modules.[22,23]

Recently, several other types of modular reconfigurable robotic systems have been considered. Yim[58] considered modular robots composed of a few basic elements that can be composed into complex systems, and used for various modes of locomotion. Yim's recent work[59] is an extension of the metamorphic robot concept to three dimensions. Kokaji,[32] Murata, Kurokawa, and Kokaji,[42] and Tomita et al.[55] considered a "fractal" system composed of modules with zero kinematic mobility, but which can walk over each other in discrete quanta due to changes in the polarity of magnetic fields. Recently they and their coworkers have developed a three-dimensional design with mechanical rather than electromagnetic couplings.[43,60] Rus and coworkers have also been developing three-dimensional metamorphic robot units.[34,51] Asama and coworkers have recently developed interesting metamorphic robots as well.[28]

As the number of modules in a metamorphic system approaches infinity, the manipulator can be viewed as a "mechatronic amoeba" because the

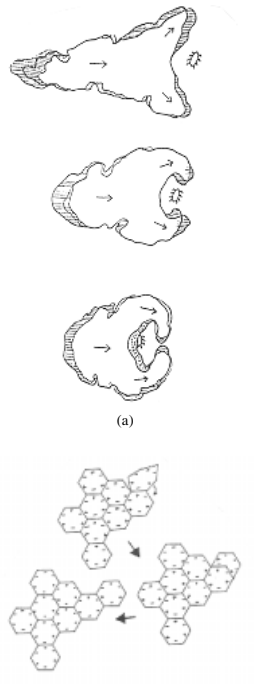


**Figure 4.** (a) An amoeba; (b) A metamorphic robot reconfiguring.

manipulator takes on a continuous appearance. Figure 4(a) illustrates a real amoeba; Figure 4(b) shows a metamorphic robot reconfiguring. Thus, the idea of metamorphic structures is not foreign to the natural world.

### 3.1. Design Issues

In this subsection, two metamorphic robot designs developed by students in the author's laboratory (as reported in refs. 4 and 46, and references therein) are described. Regardless of the particular design, a number of basic principles must be observed if a metamorphic robot is to function reliably. These include:

- Connections between modules must be tolerant of small errors in relative position and orientation.
- Communication and coordination between modules must be robust with respect to variations in environment.
- Load-carrying capability of a configuration of units must satisfy task requirements, and the mechanical design of the modules must ensure that they can carry a load.

The recent design of Murata et al.[43] is an excellent example that satisfies these constraints. In the following, we review the author's previous designs in this area.

***Regular Hexagonal Linkages*** One of the designs that satisfies the defining properties of a metamorphic robot in the planar case involves the use of hexagonal modules. Each module, as shown in Figure 1(b), consists of six links of equal length, forming a six-bar linkage. Because of the hexagonal shape, the modules completely fill the plane without any gaps. In this design, each module possesses three degrees of freedom that are controlled by placing actuators at alternate joints. This enables each module to move around another while remaining connected at all times during this motion. The modules are provided with male-female coupling mechanisms actuated by DC motors. Because of the symmetry of the module, male connectors always meet female connectors and vice versa. In this particular implementation, each male connector (passive T-shaped protrusion) is spring-loaded to allow for alignment errors and to provide passive compliance during the reconfiguration sequence. The female connector is active, and consists of a motor that drives dual cams to envelop the male.

***Rigid Sliding Squares*** A second planar design developed in the author's laboratory consists of square units with the ability to slide, without deforming, over each other. This locomotion is achieved via a rack-and-pinion drive that moves a slider that locks the units together. As one unit is moved to the corner of another, a second slider locks into the moving unit transversally. Then the slider responsible for the motion releases the moving module. Details of this design can be found in ref. 4, and references therein.

In both designs, each module must also contain a microprocessor that controls the link actuators and the connector motors, making the module computationally self-contained.

### 3.2. Kinematic Modeling and Motion Planning

Potential applications of metamorphic systems composed of a large number of modules include: (1) obstacle avoidance in highly constrained and unstructured environments; (2) "growing" structures composed of modules to form bridges, buttresses, and other civil structures in times of emergency; (3) envelopment of objects, such as recovering satellites from space; and (4) performing inspections in constrained environments such as nuclear reactors. Ilustrations of some of these applications are shown in Figure 5.

Use of metamorphic robots in these application areas requires an understanding of both how configurations are described, and how reconfiguration sequences are automatically generated. One technique for the automatic self-reconfiguration of a metamorphic robot is to define a measure of distance between the current and desired position, and to seek module motions at each instance that minimize the distance between the current shape and the desired one.

In order to define distance between configurations, we will first need a concept of distance between modules. The regular Euclidean metric is an acceptable choice, but one that more accurately reflects the least number of moves required by a module to move between two points is defined as follows: First, we construct a *lattice connectivity graph*, that is, a graph with vertices at lattice points, and edges that are straight lines connecting all neighboring vertices. The distance measured along *shortest paths* connecting two lattice points in this graph is what we will refer to as the distance between two lattice points/modules. For example, if a metamorphic robot is composed of square or cubic modules, distance between modules would be given by the Manhattan/Taxicab metric in $\mathbb{R}^N$ (see ref. 14 for an explanation and other examples). We call this measure of distance a *lattice metric*, and denote it simply as $d(a, b)$, where $a$ and $b$ are lattice points. By definition, the lattice metric yields the minimal distance between lattice points, while defining a path connecting all intermediate lattice points. We may "induce" a metric (measure of distance) in the set of all configurations of $n$ modules by using the lattice distance in the following way.

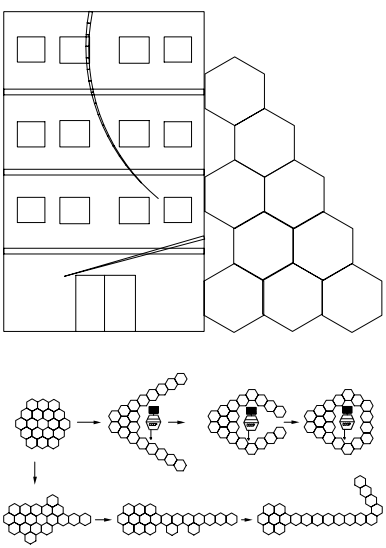


**Figure 5.** Potential application of metamorphic robots.

Let $\{a_1, a_2, \ldots, a_n\}$ denote the modules in $A$ and $\{b_1, b_2, \ldots, b_n\}$ denote the modules in $B$, where $A$ and $B$ represent two configurations. An assignment from $A$ to $B$ is a bijective function from the members of $A$ to the members of $B$. It can be represented by permutation on the indices $\{1, 2, \ldots, n\}$ of $A$ to the indices $\{1, 2, \ldots, n\}$ of $B$, as

$$\pi = \begin{pmatrix} 1 & 2 & \cdots & n \\ \pi(1) & \pi(2) & \cdots & \pi(n) \end{pmatrix}$$

For $n$ modules there exist exactly $n!$ different permutations, that is, $n!$ different ways to rearrange the numbers in the set $\{1, 2, \ldots, n\}$. The set of all permutations of $n$ elements, denoted $\Pi_n$, is a group under the operation of composition.

We are interested in the sum of the distances of the matched module pairs,

$$f_\pi(A, B) = \sum_{i=1}^{n} d\left(a_i, b_{\pi(i)}\right)$$

From all possible permutations, we take the one that gives us the minimal value (optimal assignment),

$$\delta_C(A, B) = \min_{\pi \in \Pi_n} f_\pi(A, B) = \min_{\pi \in \Pi_n} \left( \sum_{i=1}^{n} d\left(a_i, b_{\pi(i)}\right) \right)$$

$\delta_C(.,.)$ is a metric on the set of all possible configurations; that is, (1) it is positive definite, (2) it is symmetric, and (3) the triangle inequality holds (see ref. 46 for proof). This metric can be calculated in $\mathcal{O}(n^3)$

operations. Furthermore, it provides a lower bound on the minimal number of moves required to reconfigure from $A$ to $B$ (or $B$ to $A$)[14]. This is easy to see, because if one were able to track the number of moves that each module makes during an optimal reconfiguration sequence, it would be clear that each module travels *at least* the lattice distance between its initial and final configuration. Hence, minimizing over all possible sums of lattice distances cannot give a value greater than the total number of single-module moves in a reconfiguration sequence.

For an illustration of the computation of this metric using the Hungarian algorithm, and the fact that the metric gives a lower bound on the number of moves required to reconfigure, consider the following example from ref. 46. Figure 6 shows an optimal reconfiguration sequence under the constraints that the unmarked hex is a fixed base, all modules move only themselves (without carrying other modules), and each moves at most one lattice space at a time.

The matrix $\mathbf{D}$ in Eq. (14) is formed by considering all the distances between various modules with the initial and final configurations overlaid. More precisely, the entry in the $i$th row and $j$th column is $d(a_i, b_j)$:

$$\mathbf{D} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1' & 2' & 3' & 4' \end{array} \\ \begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 2 & 1 & 1 \\ 2 & 1 & 1 & 2 \\ 1 & 1 & 2 & 3 \end{pmatrix} \end{array} \tag{14}$$

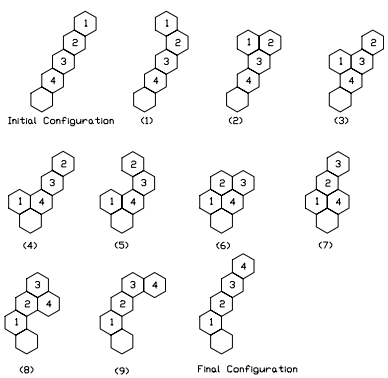



**Figure 6.** An optimal reconfiguration sequence.

Performing column operations (subtracting the minimum element of each column from each column respectively), we get the matrix in Eq. (15). In this case, we find an independent set of zeros (boxed entries) at this step in the procedure. If this had not happened, we would continue by subtracting the minimal element from each row of the matrix in (15) to get the *reduced matrix* (which is again Eq. (15) in this case).

$$\mathbf{D}' = \begin{pmatrix} 3 & 2 & 1 & \boxed{0} \\ 2 & 1 & \boxed{0} & 0 \\ 1 & \boxed{0} & 0 & 1 \\ \boxed{0} & 0 & 1 & 2 \end{pmatrix} \tag{15}$$

If an independent set of zeros still had not been found, the next step would be to modify the *reduced matrix* by covering all the zeros with lines. Then we would subtract the value of the smallest element not covered by the lines from all the uncovered elements and add it to each twice-covered element (i.e., the element lying at the intersection of two covering lines). Doing this we get the *modified matrix* $\hat{\mathbf{D}}$. This matrix generally will contain several combinations of independent 0s, any of which solves the problem and gives the value of $\delta_C(A, B)$.

The value of $\delta_C(A, B)$ is given by summing the values in the original matrix (14) in the positions of the independent zeros. In this example, it corresponds to the boxed elements in Eq. (15):

$$\delta_C(A, B) = d_{14'} + d_{23'} + d_{32'} + d_{41'} = 1 + 1 + 1 + 1 = 4$$

In contrast, the optimal reconfiguration sequence illustrated in Figure 6 required 10 moves. Hence, we observe for this example the general result that

$$\delta_C(A, B \leq \min(\text{Moves}(A, B))$$

In fact, it can be shown[14,46] that this lower bound is worst (in the sense of tightness) in cases where the initial and final configurations are long serial structures in close proximity to each other. In most other cases it is very tight. Upper bounds on min(Moves($A$, $B$)) are found in ref. 14 as well.

## 4. SUMMARY AND CONCLUSIONS

In this paper, two kinds of biologically inspired robots investigated in the author's previous work were reviewed: snakelike hyper-redundant manipulators

and amoeboid metamorphic robots. A survey of research in these areas being performed in other groups throughout the world was also provided. Applications of both kinds of robots were examined, and issues in the design, dynamics, and motion planning of these devices were reviewed.

As the cost and size of computational and electronic hardware continues to decrease, the realization of highly complex robotic systems in real-world applications becomes more likely. In just the past five years, the number of research groups worldwide pursuing work in the area of metamorphic robots has dramatically increased. Meanwhile, investigations into the use of snakelike robots in industrial inspection and nuclear waste remediation indicate that this area of study is in the process of transition between basic research and application-specific development.

## REFERENCES


1. V.V. Anderson and R.C. Horn, Tensor arm manipulator design, ASME Trans 67-(DE-57) (1967), 1–12.
2. J.E. Bobrow and F. Jabbari, Adaptive pneumatic force actuation and position control, ASME J Dyn Syst 113:(2) (1991), 267–272.
3. I.-M. Chen and J.W. Burdick, Enumerating the non-isomorphic assembly configurations of modular robotic systems, Int J Robot Res 17:(7) (1998), 702–719.
4. C.-J. Chiang and G.S. Chirikjian, Similarity metrics with applications in modular robot motion planning, Autonomous Robots 10:(1) (2001), 91–106.
5. G.S. Chirikjian, Theory and applications of hyper-redundant robotic manipulators, Ph.D. dissertation, Department of Engineering and Applied Science, California Institute of Technology, May 1992.
6. G.S. Chirikjian, Kinematics of a metamorphic robotic system, Proc 1994 IEEE Int Conf on Robotics and Automation, San Diego, CA, 1994, pp. 449–455.
7. G.S. Chirikjian, Hyper-redundant manipulator dynamics: A continuum approximation, Adv Robotics (special issue on highly redundant manipulators) 9:(3) (1995), 217–243.
8. G.S. Chirikjian, Inverse kinematics of binary manipulators using a continuum model, J Intell Robot Syst 19 (1997), 5–22.
9. G.S. Chirikjian and J.W. Burdick, A geometric approach to hyper-redundant manipulator obstacle avoidance, ASME J Mech Design 114 (1992), 580–585.
10. G.S. Chirikjian and J.W. Burdick, A modal approach to hyper-redundant manipulator kinematics, IEEE Trans Robot Automat (1994), 343–354.
11. G.S. Chirikjian and J.W. Burdick, Kinematically optimal hyper-redundant manipulator configurations, IEEE Trans Robot Automat (1995), 794–806.
12. G.S. Chirikjian and J.W. Burdick, Kinematics of hyper-redundant robot locomotion. IEEE Trans Robot Automat 11 (1995), 781–793.
13. G.S. Chirikjian and I. Ebert-Uphoff, Numerical convolution on the Euclidean group with applications to workspace generation, IEEE Trans Robot Automat 14:(1) (1998), 123–136.
14. G.S. Chirikjian, A. Pamecha, and I. Ebert-Uphoff, Evaluating efficiency of self-reconfiguration in a class of modular robots, J Robot Syst 13:(5) (1996), 317–338.
15. G.S. Chirikjian and S. Zhou, Metrics on motion and deformation of solid models, ASME J Mech Design 120:(2) (1998), 252–261.
16. J.J. Craig, Introduction to robotics: Mechanics and control, Addison-Wesley, Reading, MA, 1989.
17. I. Ebert-Uphoff, On the development of discretely-actuated hybrid-serial-parallel manipulators, Ph.D. dissertation, Department of Mechanical Engineering, Johns Hopkins University, June 1997.
18. K. Etemadi-Zanganeh and J. Angeles, Instantaneous kinematics of general hybrid parallel manipulators 117:(3) (1995).
19. K.R. Etzel and J.M. McCarthy, Spatial motion interpolation in an image space of SO(4), Proc 1996 ASME Design Eng Tech Conf and Comput Eng Conf, Irvine, CA, August 18–22, 1996.
20. P. Fanghella and C. Galletti, Metric relations and displacement groups in mechanism and robot kinematic, ASME J Mech Design 117 (1995), 470–478.
21. T. Fukuda and S. Nakagawa, Dynamically reconfigurable robotic systems, Proc 1988 IEEE Conf on Robotics and Automation, 1988, pp. 1581–1586.
22. S. Hackwood and J. Wang, The engineering of cellular robotic systems, IEEE Int Symp on Intelligent Control, Arlington, VA, August 24–26, 1988.
23. S. Hackwood and G. Benni, Self-organization of sensors for swarm intelligence, 1992 IEEE Conf on Robotics and Automation, 1992, pp. 819–829.
24. G.J. Hamlin and A.C. Sanderson, TETROBOT: A modular approach to parallel robotics, IEEE Robot Automat Mag 4:(1) (1997), 42–50.
25. W. Henning, F. Hickman, and H. Choset, Motion planning for serpentine robots, Proc ASCE Robotics 98, Albuquerque, NM, 1998.
26. S. Hirose and Y. Umetani, An active cord mechanism with oblique swivel joints and its control, Ro Man Sy 4 (1981), 327–340.
27. S. Hirose and A. Morishima, Design and control of a mobile robot with an articulated body, Int J Robot Res 9:(2) (1990), 99–114.
28. K. Hosokawa, et al., Self-organizing collective robots with morphogenesis in a vertical plane, Proc IEEE Int Conf on Robotics and Automation, Leuven, Belgium, May 1998, pp. 2858–2863.
29. K. Kazerounian and J. Rastegar, Object norms: A class of coordinate and metric independent norms for displacement, Flexible Mech Dyn Anal ASME DE-47 (1992), 271–275.
30. L. Kelmar and P.K. Khosla, Automatic generation of kinematics for a reconfigurable modular manipulator system, Proc 1988 IEEE Conf on Robotics and Automation, 1988, pp. 663–668.
31. H. Kobayashi and S. Ohtake, A local feedback law for hyper-redundant manipulators, Adv Robotics 9:(3) (1995), 245–254.
32. S. Kokaji, A fractal mechanism and a decentralized

control method, Proc U.S.- Japan Symp Flexible Automation, 1129/1134, 1988.
33. A. Sh. Koliskor, The 1-coordinate approach to the industrial robot design, 5th IFAC Symp, Suzdal, USSR, April 22–25, 1986, pp. 108–115.
34. K. Kotay, D. Rus, M. Vona, and C. McGray, The self-reconfigurable robotic molecule, Proc IEEE Int Conf on Robotics and Automation, Leuven, Belgium, 1998, pp. 424–431.
35. A.B. Kyatkin and G.S. Chirikjian, Regularized solutions on a nonlinear convolution equation on the Euclidean group, Acta Appplicandae Mathematicae 53 (1998), 89–123.
36. A.B. Kyatkin and G.S. Chirikjian, Synthesis of binary manipulators using the Fourier transform on the Euclidean group, ASME J Mech Design 121 (1999), 9–14.
37. J. Lenarčič, On hyper-redundant multiple-link robots Lab Robot Automat 8:(1) (1996), 11–16.
38. J. Lenarčič and V. Parenti-Castelli (Editors), Recent advances in robot kinematics, Kluwer, Boston, 1996.
39. R. Malczyk and A. Morecki, Elastic manipulator of the elephant trunk type, Biocybernet Bio Eng 7 (1987).
40. J.M.R. Martinez and J. Duffy, On the metrics of rigid body displacement for infinite and finite bodies, ASME J Mech Design 117 (1995), 41–47.
41. H. Mochiyama, Shape control of manipulators with hyper degrees of freedom, Ph.D. dissertation, JAIST, Japan, March 1998.
42. S. Murata, H. Kurokawa, and S. Kokaji, Self-assembling machine, Proc IEEE Int Conf on Robotics and Automation, 1994, pp. 441–448.
43. S. Murata, et al., A 3-D self-reconfigurable structure, Proc IEEE Int Conf on Robotics and Automation (ICRA '98), 1998, pp. 432–439.
44. F. Naccarato and P.C. Hughes, Inverse kinematics of variable-geometry truss manipulators, J Robot Syst 8:(2) (1991), 249–266.
45. J. Ostrowski and J. Burdick, The geometric mechanics of undulatory robotic locomotion, Int J Robot Res 17:(7) (1998), 683–701.
46. A. Pamecha, I. Ebert-Uphoff, and G.S. Chirikjian, Useful metrics for modular robot motion planning, IEEE Trans Robot Automat 13:(4) (1997), 531–545.
47. F.C. Park, Distance metrics on the rigid-body motions with applications to mechanism design, ASME J Mech Design 117 (1995), 48–54.
48. D.L. Pieper, The kinematics of manipulators under computer control, Ph.D. dissertation, Stanford University, October, 1968.
49. D. Resnik and V. Lumelsky, Sensor-based motion planning in three dimensions for a highly redundant snake robot, Adv Robotics 9:(3) (1995), 255–280.
50. B. Roth, J. Rastegar, and V. Scheinman, On the design of computer controlled manipulators, 1st CISM-IFTMM Symp on Theory and Practice of Robots and Manipulators, 1973, pp. 93–113.
51. D. Rus, Self-reconfiguring robots, IEEE Intell Syst 13:(4) (1998), 2–5.
52. R.J. Salerno, C.F. Reinholtz, and H.H. Robertshaw, Shape control of high degree-of-freedom variable geometry trusses, Proc Workshop on Computational Aspects Control Flexible Systems, Part 2, Williamsburg, VA, July 12–14, 1988.
53. K. Suzumori, S. Iikura, and H. Tanaka, Development of flexible microactuator and its applications to robotic mechanisms, 1991 IEEE Conf on Robotics and Automation, Sacramento, CA, April 1991.
54. S. Tavakkoli and S.G. Dhande, Shape synthesis and optimization using intrinsic geometry, Proc ASME Design Conf, Chicago, Il, September 16–19, 1990.
55. K. Tomita, et al., "Reconfiguration method for a distributed mechanical system," Distributed Autonomous Robot Syst 2, H. Asama, et al. (Editors), Springer, 1996.
56. K.J. Waldron and P.-H. Yang, "Parallel arrays of binary actuators," Advances in Robot Kinematics: Analysis and Control, J. Lenarčič and M.L. Husty (Editors), Kluwer, 1998, pp. 17–26.
57. J.F. Wilson and U. Mahajan, The mechanics and positioning of highly flexible manipulator limbs, ASME J Mech Trans Automat Design 111 (1989).
58. M. Yim, A reconfigurable modular robot with many modes of locomotion, Proc 1993 JSME Int Conf on Advanced Mechatronics, Tokyo August 1993, pp. 283–288.
59. M. Yim, J. Lamping, E. Mao, and J.G. Chase, Rhombic dodecahedron shape for self-assembling robots, Xerox PARC SPL TechReport P9710777, 1997.
60. E. Yoshida, et al., Distributed formation control for a modular mechanical system, Proc IEEE/RSJ Intelligent Robot and Systems (IROS '97), 1997, pp. 1091–1097.