Proceedings of the 1996 IEEE
International Conference on Robotics and Automation
Minneapolis, Minnesota - April 1996

# A Useful Metric For Modular Robot Motion Planning

Amit Pamecha Gregory Chirikjian
Department of Mechanical Engineering
Johns Hopkins University
Baltimore, MD 21218

## Abstract

In this paper we examine the problem of dynamic self-reconfiguration of a class of modular robotic systems referred to as *metamorphic* systems. A metamorphic robotic system is a collection of mechatronic modules, each of which has the ability to connect, disconnect, and climb over adjacent modules. We define a concept of distance between metamorphic robot configurations which satisfies the formal properties of a metric. This metric, called the optimal assignment metric, is then applied to the automatic self-reconfiguration of metamorphic systems from any initial to any final specified configuration. The technique of simulated annealing is used to drive the re-configuration process with the optimal assignment metric as the cost function. By driving the distance between the present and the goal configuration to zero, sequences of configurations are generated.

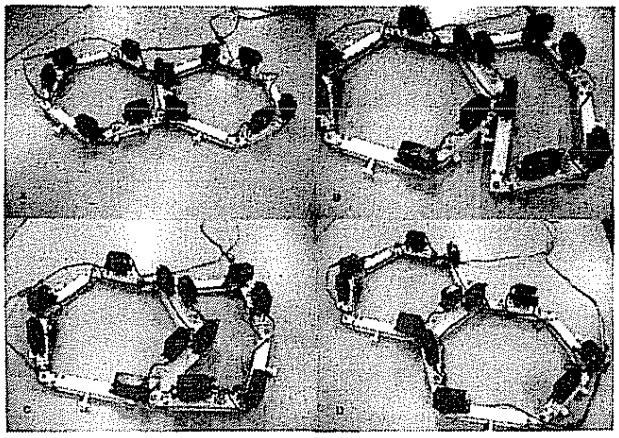

Figure 1: Hardware demonstration of a single 'move' in the reconfiguration process with two modules.

## 1 Introduction

A *metamorphic* robotic system [Ch94] is a collection of independently controlled mechatronic 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 morphology (i.e., a change in the relative location of modules within the collection) results from the locomotion of each module over its neighbors. Thus a metamorphic system has the ability to dynamically self-reconfigure. Other distinguishing features of metamorphic robots are described in [Ch94].

Figure 1 is an example of hexagonal metamorphic robot modules. In this figure a single "move" is made by one module over the other. For details of hardware that perform this kind of motion see [PKC95,MuKK95]. While the designs in those works have hexagonal symmetry, the method presented here is applicable to all polyhedral designs.

Potential applications of metamorphic systems composed of a large number of modules include : (1) obstacle avoidance in highly constrained and unstructured environments (such as a nuclear waste site); (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.

This paper addresses issues in the kinematics and motion planning of metamorphic systems with a fixed base, i.e., 'manipulators,' as opposed to 'mobile robots.' No distinction is made between 'motion planning' and 'self-reconfiguration' of these systems - these words are synonymous in the context of metamorphic systems. In Section 2, we review kinematics and motion planning issues pertaining to metamorphic robots. In Section 3, a definition of 'distance' between configurations is given and illustrated with an example. In section 4, the method of simulated annealing is used to drive the cost function based on this metric to zero. Section 5 discusses the results obtained from the implementation of the simulated annealing algorithm for two different energy functions.

## 2 Problem Formulation and Mathematical Background

In this section, we formulate the general problem of describing a metamorphic robot configuration, and characterize constraints on module motion.

One way to view space is as a collection of connected close-packed polyhedra (which is often referred to as a 'tessellation' of space), the centers and/or vertices of which form a regular lattice. In our problem,

elements of the lattice (individual polyhedral cells) are either filled with robotic modules or obstacles or remain empty. By denoting the origin as the vector $\vec{0} \in \mathbf{R^N}$ centered at the fixed base module, and defining unit vectors along any $N$ independent directions which contain at least two lattice points (module centers), every point in the lattice is given a unique set of coordinates with the above described unit vectors defining coordinate axes.

In order to define distance between configurations, we will first need a concept of distance between modules. While the regular Euclidean metric is an acceptable choice, one that more accurately reflects the least possible number of moves required by a module to move between two points is defined as follows. First construct a *lattice connectivity graph*, i.e., 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 $\mathbf{R^N}$. We call this measure of distance a *lattice metric*, and denote it as $\delta_L(a, b)$, 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. Though this distance is a unique number, the number of equidistant paths may be very large.

The kinematic constraints governing the motion of one module over the surface of a collection of other modules are:

- Modules can only move into spaces which are not already occupied.
- Every module must remain connected to at least one other module, and at least one of the modules must stay connected to the fixed base from which the collection of modules originated.
- A single module may only move one lattice space per timestep, and it achieves this motion by deforming and mating faces to faces (or in the planar case edges to edges, as shown in Figure 1).
- Modules must observe limitations on their motions due to internal joint limits.

Under these constraints, the motion planning/self-reconfiguration problem becomes : *determination of the sequence of module motions from any given initial configuration to any given final configuration in a reasonable (preferably minimal) number of moves.*

This however leads to a computationally complex step of determining an optimal set of moves, i.e. the minimum number of moves required to completely reconfigure. To the best of our knowledge, there is no simple method of solving the above problem. The reason is that for any number of modules $n$, the number of connected configurations possible appears to be exponential in $n$. To find a optimal sequence of configurations leading from the initial configuration to the final configuration is akin to finding the shortest path in a graph consisting of such configurations as vertices. Because of the size of the graph, this requires exponential time to solve using standard graph search techniques.

As a result, we have to look for heuristics which can give a near optimal solution. Any such heuristic would require a distance measure between configurations so that the shortest path between configurations is picked. The metric discussed in this paper is one possible distance measure. The metric properties ensure that it's a well defined distance measure.

## 3 Defining Distance Between Configurations

In this section, we define a measure of distance between *configurations* of metamorphic systems as opposed to distance between modules as discussed before. Each configuration of $n$ modules is defined by the collection of $n$ connected lattice spaces which it fills. That is, we do not distinguish between different modules, and any permutation of labels has no effect on configuration since all modules are identical. Therefore, two configurations with the same shape and relative position in space are said to be the same. Metrics that define distance between configurations in this way are denoted $\delta_C$ for 'configuration metric.' Formally, a metric function satisfies the properties :

$$d(A,B) \geq 0 \quad \text{and} \quad d(A,B) = 0 \Longleftrightarrow A = B$$
$$d(A,B) = d(B,A) \tag{1}$$
$$d(A,B) + d(B,C) \geq d(A,C),$$

which are referred to as positive definiteness, symmetry, and the triangle inequality, respectively. The original set, together with a metric function defined on that set is called a *metric space*.

Let the present configuration of the robot be described by the set of modules $A$, where $a_i \in A$ represents a lattice point occupied by a module in the configuration $A$ for $i = 1, \ldots, n$. Let the new configuration be defined by the set $B$, where $b_j \in B$ for $j = 1, \ldots, n$ represents a lattice point in the new configuration. One possible configuration metric is the discrete metric

$$\delta_C^{(1)}(A,B) = \begin{cases} 1 & A \neq B \\ 0 & A = B \end{cases}$$

Another such metric is the *overlap* metric

$$\delta_C^{(2)}(A,B) = n - |A \cap B|$$

which gives the number of modules not overlapping between configurations $A$ and $B$.

In the following subsections we define and illustrate one particluar configuration metric called the *optimal assignment* metric, which is denoted as $\delta_C^{(3)}$. $\delta_C^{(3)}$ is obtained by using the lattice metric and concepts of optimal assignment. We will assign modules (or more precisely, the lattice spaces in which the modules reside)

in two configurations in such a way that the sum of the lattice distances between matched module spaces is minimized over all possible matchings.

### 3.1 Evaluating the Optimal Assignment Metric $\delta_C^{(3)}$

The optimal assignment metric $\delta_C^{(3)}(A,B)$ between two configurations $A$ and $B$ is given by an optimal assignment of each element $a_i$ in $A$ to an element $b_j$ in $B$, $f : A \rightarrow B$, such that the sum of the distances between modules (as defined by the lattice metric) for the assignment is minimized. Equivalently, this can be represented as finding a minimum weight matching in a *bipartite graph*.

We now review the optimal assignment problem and an algorithm for solving it.

Let $m_{ij}$ be a variable which is 1 if module $a_i$ in the present configuration maps to module $b_j$ in the new configuration and 0 otherwise. $d_{ij} = \delta_L(a_i, b_j)$ is the lattice distance between module $a_i$ and $b_j$. An arbitrary assignment will have an associated cost function:

$$f(A,B) = \sum_{\substack{1\le i\le n\\ 1\le j\le n}} d_{ij}m_{ij} \tag{2}$$

with the constraints

$$\sum_{i=1}^{n} m_{ij} = 1 \quad \text{for all } j = 1,..,n \quad \text{and} \tag{3}$$

$$\sum_{j=1}^{n} m_{ij} = 1 \quad \text{for all } i = 1,..,n.$$

The constraints ensure that the assignment or mapping is a bijection. We define

$$\delta_C^{(3)}(A,B) = \min_{\Pi_n} f(A,B), \tag{4}$$

where $\Pi_n$ is the set of all possible assignments and is equivalent to the set of permutations of module labels.

Several algorithms are available for solving this optimal assignment problem. The method illustrated in the next section is the Hungarian algorithm for optimal assignment [Ku55],[Ro84].

### 3.2 Example

For an illustration of the optimal assignment algorithm, consider the following example. Figure 2(a) shows the present configuration of a six module metamorphic robot, Figure 2(b) shows the new configuration and Figure 2(c) shows a superposition of the modules in the two configurations.

Construct an $n \times n$ matrix $\mathbf{D}$, with elements $d_{ij} = \delta_L(a_i, b_j)$ as shown in Equation 5.

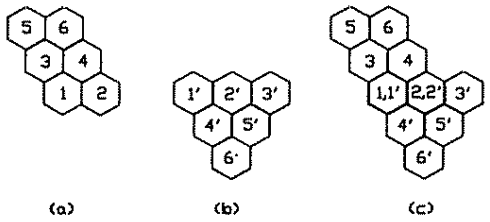


Figure 2: (a)Present Configuration (b)New Configuration (c)Module Labeling

$$\mathbf{D} = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1' & 2' & 3' & 4' & 5' & 6' \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 1 & 2 & 2\\ 1 & 0 & 1 & 1 & 1 & 2\\ 1 & 2 & 3 & 2 & 3 & 3\\ 1 & 1 & 2 & 2 & 2 & 3\\ 2 & 3 & 4 & 3 & 4 & 4\\ 2 & 2 & 3 & 3 & 3 & 4 \end{pmatrix} \end{array} \tag{5}$$

Observe that if we subtract a constant $k_q$ from the $q^{th}$ row of $\mathbf{D}$, giving rise to a new matrix $\mathbf{D'}$ with elements $d'_{ij}$, then

$$\sum_{\substack{1\le i\le n\\ 1\le j\le n}} d'_{ij}m_{ij} = \sum_{\substack{1\le j\le n\\ 1\le j\le n}} d_{ij}m_{ij} - k_q \sum_{1\le j\le n} m_{qj}$$

$$= \sum_{\substack{1\le i\le n\\ 1\le j\le n}} d_{ij}m_{ij} - k_q \tag{6}$$

using (3). Thus, an assignment $m_{ij}$ that minimizes (1) also minimizes ( 6) and vice versa. The same result is obtained if a constant $l_p$ is subtracted from the $p^{th}$ column. This gives us a method of finding the optimal assignment.

Performing row operations (subtracting $k_q$, the minimum element of each row, from each row respectively) we get the matrix in Equation 7. Similarly performing the column operations, we get the *reduced matrix* in Equation 8.

$$\mathbf{D}' = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1' & 2' & 3' & 4' & 5' & 6' \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 1 & 2 & 2\\ 1 & 0 & 1 & 1 & 1 & 2\\ 0 & 1 & 2 & 1 & 2 & 2\\ 0 & 0 & 1 & 1 & 1 & 2\\ 0 & 1 & 2 & 1 & 2 & 2\\ 0 & 0 & 1 & 1 & 1 & 2 \end{pmatrix} \end{array} \tag{7}$$

$$\overline{\mathbf{D}} = \begin{pmatrix} \boxed{0} & 1 & 1 & 0 & 1 & 0\\ 1 & \boxed{0} & 0 & 0 & 0 & 0\\ 0 & 1 & 1 & \boxed{0} & 1 & 0\\ 0 & 0 & \boxed{0} & 0 & 0 & 0\\ 0 & 1 & 1 & 0 & 1 & \boxed{0}\\ 0 & 0 & 0 & 0 & \boxed{0} & 0 \end{pmatrix} \tag{8}$$

The problem can then be solved by finding an *independent set* [1] of $n$ $0's$ in the *reduced matrix* since they correspond to the least cost assignment. The optimal assignment is given by taking $m_{ij}$ equal to 1 for the corresponding $0's$. If the number of independent $0's$ is equal to $n$ then the solution is simply the assignment $m_{ij}$ corresponding to the above $0's$. Otherwise we successively modify the *reduced matrix* to form a new *modified matrix* $\hat{\mathbf{D}}$ where there are $n$ independent $0's$. One method to do this is to find out the minimum number of lines (one line refers to one complete row or column) which cover all the $0's$ in $\bar{\mathbf{D}}$. Let $p$ be the smallest uncovered element. Modify the *reduced matrix* by subtracting $p$ from all the uncovered elements and adding $p$ to each twice covered element by the lines (i.e. each element which lies at the intersection of two lines). This is the *modified matrix* $\hat{\mathbf{D}}$. It is easy to show that the new *modified matrix* has been obtained from the preceding one by adding or subtracting a constant from different rows or columns. Thus, the problem remains the same.

For the present case the *reduced matrix* in Equation 8 contains several combinations of six independent $0's$ and so there is no need for forming a *modified matrix*. One such combination of six independent $0's$ which solves the problem is shown in Equation 8 by the boxed elements. The value of $\delta_C^{(3)}(A,B)$ is obtained by the sum of the elements in $\mathbf{D}$ corresponding to any $n$ independent 0's in $\overline{\mathbf{D}}$.

Choosing the boxed solution above, the distance between configurations ($\delta_C^{(3)}(A,B)$) is given by,

$$d_{11'} + d_{22'} + d_{34'} + d_{43'} + d_{56'} + d_{65'} = 0 + 0 + 2 + 2 + 4 + 3 = 11 \quad (9)$$

The minimal value is achieved by assigning modules with the subscripts in the above expression.

An interesting property of the optimal assignment approach is that it only needs to consider pairings of *nonoverlapping* modules, thus reducing the size of the matrices generated using the techniques of Section 3. In cases where there is substantial overlap, this can save a lot of computational effort. The proof of these properties is given in [PEC95].

# 4 The Method of Simulated Annealing

One of the most powerful and popular methods for solving classes of problems which are intractable by standard graph searching methods is the method of simulated annealing.

In this section we use the metrics discussed in the previous sections in a motion planning/ reconfiguration algorithm based on simulated annealing. It is easy to observe that a pure greedy approach will often get stuck in local minima and will not yield a solution, much less a good one.

[1] An independent set of $0's$ in a matrix is a set of $0's$, no two of which are in the same row or same column.

## 4.1 Simulated Annealing

Simulated Annealing is an algorithmic approach to solving optimization problems especially in cases where the global extremum is hidden among several local extrema [KiGV83]. The basic idea behind this algorithm comes from an analogy with simulating the annealing of solids [MeRRTT53] and slow cooling of liquids, i.e. the way metals or crystals cool and anneal to achieve the minimum energy state.

The basic simulated annealing algorithm considers the objective function to be minimized as the energy of the system. Starting from an initial state with energy $E$, the system is perturbed to a neighboring state and the change in energy, $\Delta E$, computed. If $\Delta E$ is negative, i.e. the energy is less in the new state, then the new state is accepted. If $\Delta E$ is positive, then the new state is accepted with a probability usually taken as $e^{-\Delta E/T}$, where $T$ is a control parameter corresponding to temperature in the analogous case of thermodynamic cooling. Besides the *energy function* and the *control parameter* $T$, a *cooling schedule* is required, i.e. a scheme for changing $T$ as the algorithm proceeds, usually taken as $T_{i+1} = \beta \times T_i$ where $\beta < 1$ is a constant. Initially $T$ is set to a high value and after a certain number of steps ($k_c$) at each value of $T$, its value is decreased by the factor $\beta$. Finally a *stopping criterion* is required to end the algorithm.

## 4.2 Energy Functions for Simulated Annealing

In the application of simulated annealing to *metamorphic robot* motion planning/ reconfiguration, energy (or cost) functions which reflect the differences between configurations are important. Using measures of distance that formally satisfy the definition of a metric guarantee that we have a well defined stopping criterion, i.e, $\delta_C(A_g, A_k) = 0$ or $k = M_{max}$, which ever comes first. $A_k$ is the $k^{th}$ configuration in a sequence of configurations, $A_g$ is the goal configuration, and $M_{max}$ is the maximum allowable moves. Furthermore, the triangle inequality is important because without it the optimal cost reducing path generated from configuration to configuration could involve large excursions. This could occur without the triangle inequality because the evaluation of cost between two configurations may be larger than the sum of distances between each configuration and an intermediate one.

In this paper, two types of metric based cost functions of the form

$$E = \delta_C(C, F)$$

are used with simulated annealing where $C$ denotes the current configuration of the robot and $F$ denotes the final configuration desired. First, the difference of the number of modules and the number of overlapping modules of any two configurations with the same number of modules (defined as overlap metric in Section 3) is considered. Evaluating this with the present and final configurations yields an energy function of the form:

$$E = \delta_C^{(2)}(C, F) = n - \mid C \cap F \mid$$

where $n$ is the number of modules.
Second, the metric defined using concepts of optimal assignment developed in Sections 3 and 4 applied to the current and final configurations is an energy function :

$$E = \delta_C^3(C, F)$$

### 4.3 Reconfiguration Algorithm based on Simulated Annealing

In our implementation of simulated annealing, the change in energy associated with the move from any current configuration to all possible neighboring configurations is computed at each step in the algorithm. A neighboring configuration of $C$ (denoted $C_N$) is a connected configuration for which $\delta_C^{(i)}(C, C_N) = 1$ for $i = 2, 3$ . If a move or moves consistent with the motion constraints results in a neighboring configuration with reduced energy, i.e. $E(C_N, F) < E(C, F)$, then one of these moves is selected randomly. If none of the moves result in a decrease in energy, then a normalized probability is obtained for each move based on the probability function

$$p_i = \frac{e^{-\Delta E_i/T}}{\sum_{i=1}^{k_m} e^{-\Delta E_i/T}} \qquad (10)$$

where $k_m$ is the number of all possible moves. The next move is selected based on the above probability.
The algorithm can then be described as

**Reconfiguration Using Simulated Annealing**
Assign $T = T_{initial}$
While { (Final Configuration not reached) and (moves made < moves allowed)}
After every $k$ moves, let $T = \beta \times T$
Find the energy $E$ of the current config. using the given cost function
Find out all possible moves of all modules.
For each possible move $i$
Find the change in energy $\Delta E_i$ if that move is taken
If there is a move/ moves for which $\Delta E$ is negative
Pick any one of those moves
Else If $\Delta E$ is positive for all moves
Assign a probability
$p_i = \frac{e^{-\Delta E_i/T}}{\sum_{i=1}^{k_m} e^{-\Delta E_i/T}}$ to each move
Pick a move based on the assigned probabilities.

## 5 Results

We ran twenty trials of the simulated annealing algorithm for two sets of initial and final configurations (Figures 3 and 5), for each of eight different initial values of the temperature parameter $T$. The *annealing schedule* consisted of 10 moves at each value of $T$ followed by a decrease in the value of $T$ by a factor of 0.8, i.e. $T_{i+1} = 0.8 * T_i$. The algorithm stopped if the final configuration was reached or if 300 moves had taken place. The results for the two cases are shown in figures 4, and 6. Note that average number of moves made were considered in these figures because the minimum number of moves do not present a true picture. The minimum moves made were close to those obtained by hand.

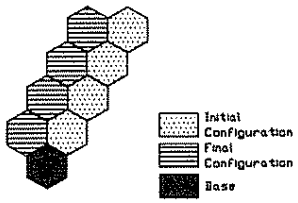


Figure 3: Reconfiguration involving two serial structures parallel to each other

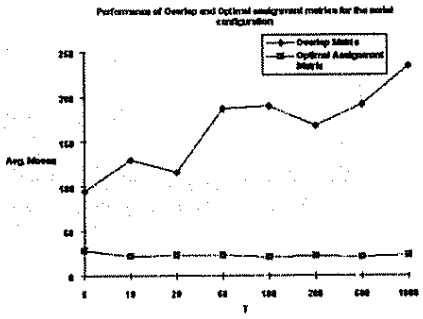

Figure 4: Results for the serial configuration

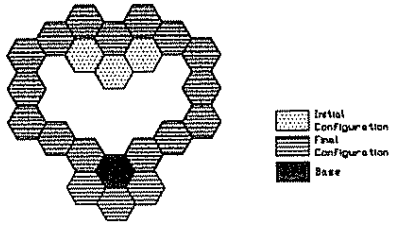


Figure 5: Reconfiguration involving breaking a loop structure

Two very different sets of configurations were chosen in order to ascertain the behavior of the simulated annealing algorithm with two different energy functions. Figure 4 shows the result for configurations corresponding to figure 3. The initial and final configurations in this case are two serial structures parallel to each other. Because of the motion constraints the modules cannot simply move into the lattice spaces corresponding to the final configuration but instead

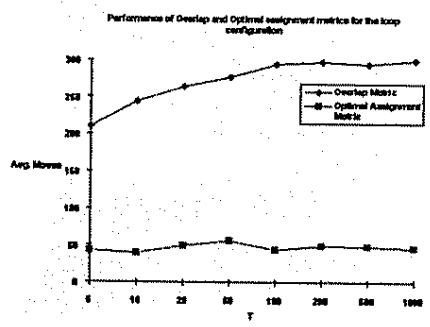

Figure 6: Results for the loop configuration

have to move over each other to attain the final configuration. Hill climbing is involved in this case and the two energy functions lead to different behaviors.

The results in figure 6 correspond to the configurations in fig. 5. In this case the overlapping modules form a loop. The non-overlapping modules corresponding to the initial configuration lie inside the loop while those corresponding to the final configuration lie outside the loop. Since both energy functions can be locally minimized by preserving the overlap, a definite hill climbing is involved.

As can be seen, the energy function corresponding to the optimal assignment metric yields better results than the overlap metric in all cases. When optimal assignment is used the moves made are usually those which reduce the distance between an empty lattice space in the final configuration and a module in the present configuration. In the case of overlap metric, unless there is a move which increases the overlap both good and bad moves are equally likely.

Another observation was that the initial temperature had no noticeable effect when the optimal assignment metric is used as the energy function. This is because if there is a move possible which reduces energy, simulated annealing will always choose that and in that case the value of the ratio $\Delta E/T$ does not influence the result. For example, in both cases $\Delta E_i$ is always negative for some move $i$, until a local minima is reached and such minima are few in the complete reconfiguration of the robot from the initial to the final configuration. Hence the above behavior. In the case when the overlap metric is used as an energy/cost function there are a large number of local minima and plateaus, i.e there's no move which decreases energy. As a result when $T$ is large the value of the ratio $\Delta E/T$ is approximately the same for all moves (Observe that $|\Delta E| < 1$ in all cases for one single move). This results in an approximately equal probability for all moves and so a bad move is as likely as a good move. This affects the average number of moves required to reconfigure.

An undesirable feature which was observed for the metrics considered was that they produced 'branching', i.e. a number of branches of modules can come out of the initial configuration, all pointing towards the final configuration. Since the configuration has to remain connected at all times, reconfiguration in this case involves overcoming 'deep' local minima.

Even though simulated annealing is a very powerful technique, it has the uncertainties associated with a randomized approach. As a result, it is best suited for performing a number of off line simulations and then using the best one out of those to reconfigure the robot instead of real time application. A pure greedy algorithm would not have worked at all for the sets of configurations shown in figures 3 and 5.

## 6 Conclusions

In this paper we define a useful metric which is one of many possible measures of distance between configurations of a metamorphic system. We then illustrate how this metric is applied to the motion planning/self-reconfiguration of metamorphic robotic systems. The method of simulated annealing was used with this metric as the energy function for two sets of initial and final configurations (one simply connected and one containing a loop). It was shown that the perfomance of simulated annealing using the metric developed in this paper performs better than with another cost function which seeks to maximize the number of overlapping modules.

## 7 References


[Ch94] Chirikjian, G S., "Kinematics of a Metamorphic Robotic System," *Proceedings of the 1994 IEEE Int. Conf. on Robotics and Automation*, San Diego, CA, May 1994, pp.449-455.

[ChPE96] Chirikjian, G.S.,Pamecha, A., Ebert-Uphoff, I., "Evaluating Efficiency of Self-Reconfiguration in a Class of Modular Robots," *Journal of Robotic Systems*, 1996 (To appear).

[KiGV83] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., "Optimization by Simulated Annealing," *Science*, Vol. 220, No. 4598, pp. 671-679, May 13, 1983.

[Ku55] Kuhn, H.W., "The Hungarian Methods for the Assignment Problem," *Naval Res. Logist. Quart.*, Vol 2, 1955, pp. 83-97.

[MeRRTT53] Metropolis, N., Rosenbluth, A, Rosenbluth, M., Teller A., Teller E., " Equation of State Calculation by Fast Computing Machines ", *Journal of Chemical Physics*, vol.21, pp 1087-1092, 1953.

[MuKK95] Murata, S., Kurokawa, H., Kokaji, S., 'Self-Organizing Machine," *Video Proceedings, 1995 IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, May 1995.

[PaC95] Pamecha, A., Chirikjian, G.S., "A Metamorphic Robotic System: Simulation and Connector Mechanism Demonstration" *Video Proceedings, 1995 IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, May 1995.

[PEC95] Pamecha, A., Ebert-Uphoff, I., Chirikjian, G.S., "Useful Metrics for Modular Robot Motion Planning", *J.H.U. Tech. Report RMS-5-95-1*, 1995.

[PKC95] Pamecha, A., Kohaya, Y., Chirikjian, G S., "Design of Error Tolerant Coupling Mechanisms for Metamorphic Robots," *Third IASTED Conference on Robotics and Manufacturing*, Cancun, Mexico, June 1995, pp 7-13.

[Ro84] Roberts, F., *Applied Combinatorics*, Prentice Hall, Englewood Cliffs, New Jersey, 1984

Proceedings of the 1996 IEEE
International Conference on Robotics and Automation
Minneapolis, Minnesota - April 1996

# Polynomial Motion Generation of Manipulators with Technological Constraints

Patrick PLEDEL
Laboratoire d'Automatique de Nantes, URA 823
Ecole Centrale de Nantes/Université de Nantes
1 rue de la Noè, 44072 Nantes, France

Yasmina BESTAOUI
Laboratoire d'Automatique de Nantes, URA 823
Ecole Centrale de Nantes/Université de Nantes,
1 rue de la Noè, 44072 Nantes, France

## Abstract

*Motion generation gives the joint positions, speeds and accelerations of manipulators, at every moment. We assume that the motion are polynomial trajectories, that ensure joint acceleration continuity. Usual kinematical constraints, obtained with approximations, are not always sufficient, then we will focus on actuators constraints on voltages and currents.*

## 1 Introduction

The minimum time motion generation has been solved in a number of ways, following the usual approach, i.e. taking as the feasible limits purely kinematic constraints on velocity and acceleration [2], [5], [6]. Conventional motion generation in joint space uses a constant bound on the acceleration. This bound must represent the global least upper bound of all operating accelerations so as to enable the manipulator to move under any operating conditions. It implies that the full capabilities of the manipulator cannot be utilized if the conventional approach is taken. The efficiency of the robotic system can be increased by considering the characteristics of the robot dynamics at the motion generation stage. [6] had applied the classical approach of point to point minimum time control to robot arms, where only a linear approximate model was used. [1] has presented a trajectory generation based on optimal control formulation. Assuming that joint torques are constrained and using the Hamiltonian formulation of the dynamic model, a minimum time cost criterion was considered. [3] have shown that most often the structure of the minimum time control requires that at least one of the actuators is always in saturation whereas the others adjust their torques so that some constraints on motion are not violated while enabling the arm to reach its final desired destination.

Several other methods were presented for the resolution of the via points motion problem. [7] used the fact that velocity and acceleration should be as close as possible to their bounds to achieve a time optimal motion. Constant velocity intervals are connected with constant acceleration ones (quadratic arcs). This results in a trajectory of all joints that move close to the given points with hopefully sufficient accuracy. [10] suggested an improvement of the algorithm with respect to the minimum task execution time and accuracy. [8] considered the optimal motion generation problem subject to various actuator constraints while the motion is constrained to an arbitrary path.

Although the obtained results are very important theoretically, practically they are not applicable directly to an industrial robot. From an user view point, it would be preferable to have a somewhat suboptimal but simpler solution to implement. For this purpose, we have chosen, a priori, a polynomial trajectory and we find parameters of the trajectory, for a $C^2$ minimal time motion. In this paper, using the formal calculus software MAPLE, we will show that the simple expressions previously obtained [5], [6], can be numerically extended when we include different actuator constraints.

The remainder of this paper is divided into six sections. While the models and the proposed problem are formulated in the second and third section, the resolution method is stated in the fourth paragraph. Some simulation results are given in the fifth paragraph and some conclusions added in the last section.

## 2 Models

### 2.1 Manipulator Model

The manipulator is assumed to be made of rigid links. For an n joints manipulator, the dynamic model can be expressed as :

$$\Gamma = A(q)\ddot{q} + \dot{q}^T B(q)\dot{q} + F(q)\dot{q} + G(q) \qquad (1)$$

where the vectors q, $\dot{q}$ and $\ddot{q}$ are respectively the joint position, velocity and acceleration, the vector $\Gamma$ is the joint input torque, G is the gravitational force vector, B is the n x n x n Coriolis and Centrifugal force matrix, F