한국정밀공학회지 제17권 제4호 (2000년 4월)
Journal of the Korean Society of Precision Engineering Vol. 17, No. 4, April 2000.

유전 알고리즘을 이용한 2진 로봇 머니퓰레이터의 역기구학적 해석

이인석*, 류길하*

An Inverse Kinematic Analysis of a Binary Robot Manipulator using Genetic Algorithms

Ihnseok Rhee* and Gilha Ryu*

ABSTRACT

2진 로봇 머니퓰레이터는 기하학적 형상이 가변트러스 구조로 되어 있으며 조인트의 구동원으로 사용되는 엑츄에이터는 2가지의 변위, 즉 최대 및 최소 변위만으로 동작한다. 따라서 작업영역은 연속적으로 주어지는 일반 로봇 머니퓰레이터와는 달리 유한 개의 위치 벡터의 집합 형태로 나타난다. 기존의 역기구학적 해석방법을 적용하기 어려운 2진 로봇 머니퓰레이터의 불연속적인 특성에 대해 새로운 작업영역과 역기구학 문제를 정의하였다. 유전 알고리즘을 사용하여 새로이 정의된 문제의 역기구학적 해석을 수행하였으며 유전 알고리즘이 2진 로봇 머니퓰레이터의 역기구학적 해석에 있어서 효과적이고 강건한 방법임을 보여주었다.

Key Words : binary robot manipulator(2진 로봇 머니퓰레이터), inverse kinematic analysis(역기구학적 해석), genetic algorithm(유전 알고리즘), workspace(작업영역)

1. Introduction

Traditional robot manipulators are consisted of the continuous-range-of-motion actuators such as DC motors or hydraulic actuators. In contrast to the traditional robot manipulators, the actuators of the binary parallel robot manipulators have two stable states only. The structure of the manipulators consists of variable geometry truss, stacking on top of each other, to form a long serial chain. The binary manipulators have several advantages over conventional six DOF manipulators because the additional degrees of freedom facilitate obstacle avoidance and allow tasks to be performed even if some of the actuators fail. Hyper redundant robots with very large degree of kinematic redundancy are analogous, in morphology and operation, to snakes, elephant trunks, and tentacles. Because the number of possible configurations of a binary robot manipulator grows exponentially with the number of actuators it is very difficult to solve an inverse kinematic problem. The kinematics and control of hyper redundant manipulators with continuous actuators have been studied by some researchers[1,2].

* 한국기술교육대학교 제어시스템공학과

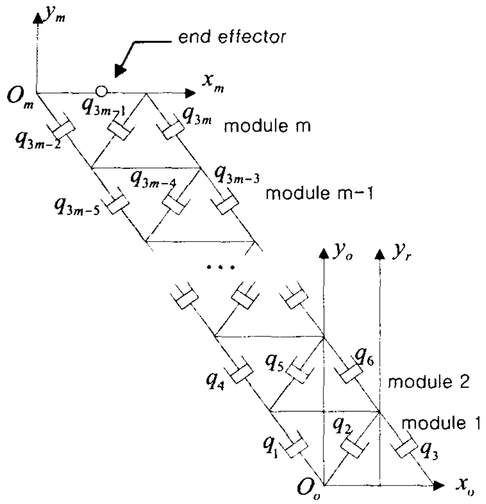

Fig. 1 Planar binary robot manipulator

An inverse kinematics of the binary manipulator was indirectly solved by using backbone curve[3,4]. The backbone curve, describing the shape of binary manipulator, is continuous. It can be generated by optimization scheme considering the curvature limit of the binary manipulator[4]. The workspace of the binary manipulator could not be described by conventional way because of its discrete nature. To describe the workspace of a binary manipulator, the point density was introduced[5]. Also an effective algorithm using the density map has been studied to compute the inverse kinematic solutions of the binary robot manipulators[6,7].

An inverse kinematic problem was solved by minimizing the error of the end effector position. A genetic algorithm was used as a minimization procedure. This algorithm was applied to obtain the inverse kinematic solutions with small error bound as well as the density map of the workspace.

2. Kinematic Modeling

Schematic drawing of a binary robot manipulator with m modules is shown in fig. 1. The binary robot manipulator can reach 2^{3m} finite positions because the actuator q_i has two states, $l_{\min}$ and $l_{\max}$, only. The i-th truss module, composed of three linear binary actuators and two plates (upper and lower plate), is shown in fig 2. The coordinate systems are assigned to each basic module for the kinematic analysis. The $\{0\}$ coordinate system is located on the base plate with its origin at O_o in fig. 1 and x- and y-axes are tangential and normal to the upper plate respectively. The origin of a moving coordinate system $\{i\}$ is located at the point D on the upper plate of the i-th module. The reference coordinate system $\{r\}$ is attached on the base plate with its origin at the center of base plate as shown in fig. 1.

3. Kinematic Analysis

3.1 Forward Kinematics

The relationship between these two coordinate systems can be described by the transformation matrix as follow,

$$T_i^{i-1} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & x_0 \\ \sin(\phi) & \cos(\phi) & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

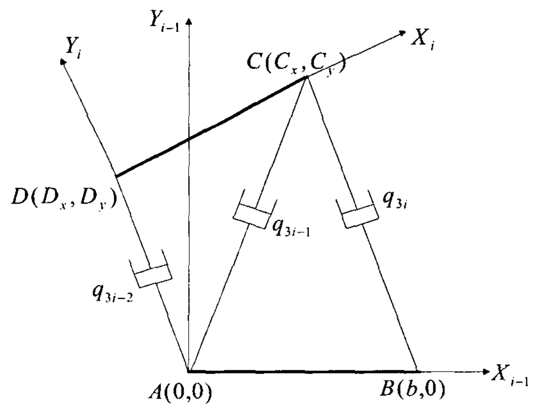

Fig. 2 Kinematic modeling of binary robot manipulator for the i-th module

where ϕ is the rotation angle between two coordinate systems and (x_0, y_0) is the position vector of origin of the coordinate system $\{i\}$ with respect to $\{i-1\}$. And (x_0, y_0) and ϕ are given as[4]

$$x_0 = D_x, y_0 = D_y, \ \phi = \tan^{-1}\left(\frac{C_y - D_y}{C_x - D_x}\right)$$

where

$$C_x = \frac{b^2 - q_{3i}^2 + q_{3i-1}^2}{2b}, \quad C_y = \sqrt{q_{3i-1}^2 - C_x^2}$$

$$D_x = \frac{D_y - k_1}{k_2}, \qquad D_y = \frac{l_2 \pm \sqrt{l_2^2 - 4 l_1 l_3}}{2 l_1}$$

$$k_1 = \frac{q_{3i-1}^2 + q_{3i-2}^2 - b^2}{2 C_y}, \quad k_2 = -\frac{C_x}{C_y}$$

$$l_1 = \frac{1}{k_2^2} + 1, \ l_2 = \frac{2 k_1}{k_2^2}, \ l_3 = \frac{k_1^2}{k_2^2} - q_{3i-2}^2$$

D_y should be chosen to satisfy $\overrightarrow{AC} \times \overrightarrow{AD} > 0$.

The coordinate of the end effector can be described using the following equation:

$$T_m^r = T_0^r T_1^0 T_2^1 T_3^2 \,..\, T_i^{i-1} \,..\, T_m^{m-1}.$$

2.2 Inverse Kinematics

The workspace in terms of conventional way is an area where robot manipulator can reach. The workspace of a binary manipulator is not given as a continuous area but a set of 2^{3m} points. This discrete characteristic makes it impossible to find an exact inverse kinematic solution in terms of conventional way for almost all target positions in plane. Therefore, new definitions of workspace and inverse kinematic solution need to be introduced for the binary manipulator.

We define the workspace of the binary manipulator, $W(\delta) \subset \boldsymbol{R}^2$, as a set of points where end effector can reach within certain bound δ, that is,

$$W(\delta) = \{ \boldsymbol{x} \mid \| \boldsymbol{x} - \boldsymbol{x}_e(\boldsymbol{q}) \| \le \delta \text{ for some } \boldsymbol{q}s \}$$

where $\|\cdot\|$, $\boldsymbol{x}_e$, and $\boldsymbol{q} = \{q_1, q_2, \cdots, q_{3m}\}$ are Euclidian norm, end effector position and a joint variable vector respectively. For a target point $\bar{\boldsymbol{x}} \in W(\delta)$, we can find a $\boldsymbol{q}$ such that

$$\| \bar{\boldsymbol{x}} - \boldsymbol{x}_e(\boldsymbol{q}) \| \le \delta \qquad (1)$$

It is clear that $\boldsymbol{q}$ satisfying equation (1) is not unique for a given target point. Therefore, an inverse kinematic solution for a binary manipulator should be described by a set of $\boldsymbol{q}$s satisfying equation (1). The inverse kinematic solution for a target point $\bar{\boldsymbol{x}}$ is characterized by triplet $\{\underline{\boldsymbol{q}}, \rho, \delta\}$ where $\underline{\boldsymbol{q}}$ is a joint vector such that

$$\| \bar{\boldsymbol{x}} - \boldsymbol{x}_e(\underline{\boldsymbol{q}}) \| \le \| \bar{\boldsymbol{x}} - \boldsymbol{x}_e(\boldsymbol{q}) \| \qquad (2)$$

for all $\boldsymbol{q}$ satisfying equation (1) and ρ denotes density defined as

$$\rho = n / 2^{3m}$$

where n is the number of joint vector satisfying equation (1). Obviously, $\underline{\boldsymbol{q}}$ is the joint vector producing minimum error and ρ represents the degree of smoothness, the ability of the manipulator to move smoothly in the neighborhood of a target point, as well as the probability that an arbitrary joint vector drives the end effector within a ball given by equation (1).

To obtain an inverse kinematic solution triplet for a target point $\bar{\boldsymbol{x}}$, we first find a joint vector $\boldsymbol{q}^*$ minimizing cost function J given as

$$J(\boldsymbol{q}) = \| \bar{\boldsymbol{x}} - \boldsymbol{x}_e(\boldsymbol{q}) \|^2 \qquad (3)$$

Then $\bar{\boldsymbol{x}} \in W(\delta)$ and $\underline{\boldsymbol{q}} = \boldsymbol{q}^*$ if and only if $\boldsymbol{q}^*$ satisfies equation (1). However, it is difficult to apply optimization schemes such as the nonlinear programming or the gradient method to minimization of equation (3) since q_i has two states, $l_{\min}$ and $l_{\max}$, only.

4. Genetic Algorithm

Genetic algorithm is an optimization technique based on the mechanics of natural genetics. In genetic algorithm, the parameter set of the optimization problem is usually coded as a finite length binary string. Genetic algorithm, due to its binary nature, can be easily applied to the inverse kinematic problem of a binary manipulator.

Let v be a binary string representation of joint variables of a binary manipulator, that is,

$$v = \langle b_{3m}\, b_{3m-1} \cdots b_2\, b_1 \rangle \qquad (4)$$

where

$$b_i = \begin{cases} 0 & \text{for } q_i = l_{\min} \\ 1 & \text{for } q_i = l_{\max} \end{cases}$$

At each iteration t of a genetic algorithm, the population $P(t)$ is composed of potential solutions

$$P(t) = \{ v_1^t, v_2^t, \ldots, v_{n_p}^t \} \qquad (5)$$

where n_p is the population size which remains constant over entire iteration and individual v_i is coded as equation (4). We adopt simple genetic algorithm described in ref. [8] as a basic algorithm. In simple genetic algorithm, a population of binary coded potential solutions is constructed and undergoes three genetic operators, selection, crossover and mutation at each iteration, so called generation. Fig. 3 shows the procedures. Selection is a reproduction process of population based on the fitness of individual. We use the following formula to calculate the fitness of an individual v, the binary representation of joint variable vector $\boldsymbol{q}$.

```
procedure genetic algorithm begin  t = 0
    initialize P(t)
    evaluate P(t)
    while (not termination-condition) do
    begin
        t ← t+1
        select P(t) from P(t-1)
        recombine P(t)
        evaluate P(t)
    end
end
```

Fig. 3 A simple genetic algorithm

$$f(v) = J_{\max} - J(\boldsymbol{q})$$

where f denotes fitness and $J_{\max}$ the maximum cost in the last w generations which is called the scaling window. Individuals of new population are selected by spinning a roulette wheel with slots sized according to fitness of current population.

The selected population is recombined with two genetic operators, crossover and mutation. We mate selected individuals randomly. For each pair, crossover takes place with crossover probability p_c. Crossover operation replaces a pair $\langle \bar{b}_{3m} \cdots \bar{b}_{c+1}\, \bar{b}_c \cdots \bar{b}_1 \rangle \langle \underline{b}_{3m} \cdots \underline{b}_{c+1}\, \underline{b}_c \cdots \underline{b}_1 \rangle$ with a pair of offspring $\langle \bar{b}_{3m} \cdots \bar{b}_{c+1}\, \underline{b}_c \cdots \underline{b}_1 \rangle$ $\langle \bar{b}_{3m} \cdots \bar{b}_{c+1}\, \underline{b}_c \cdots \underline{b}_1 \rangle$ where c is the crossover point which is also chosen randomly. Mutation is random alternation of a string position. In binary code, it means that the mutated bit is interchanged between 1 and 0. Every bit of individual has a chance to be mutated with probability p_m.

In addition to basic three operators, we use the elitist strategy in which the current best individual survives intact to next generation.

As generation elapses, the best individual of each generation converges to the binary representation of q^*. We accumulate the number of distinct individuals satisfying equation (1) at each generation. The accumulated number converges to n. If n is not zero, the given target point is in $W(\delta)$. Practically, it is almost impossible to obtain n exactly since too many generations are required. However the ns for several target positions give us a knowledge about relative density.

5. Numerical Examples

In this section, examples are presented to demonstrate the effectiveness of the genetic algorithms to obtain inverse kinematic solutions for the binary robot manipulator. As an example, we consider a binary robot with $m=10$, $l_{max}=7$, $l_{min}=5$ and $b=5$. For the genetic algorithm, n_p, w, p_c, and p_m are chosen as 30, 5, 0.6, and 0.0333 respectively. Relative density map for $\delta=2.5$ is shown in fig. 4. Target positions are located at the center of the squares and equally spaced by 5 horizontally and vertically. 5000 generations are evolved for each target position to obtain the minimum error solution q^*. In fig. 4, dark zone represents high density area and thin zone for low value. The higher density a target point has the more accurate minimum error solution in terms of probability one can get.

The minimum error solutions for some target points are shown in table 1. The count n at the third column is the total number of distinct individuals satisfying equation (1) for whole 5000 generations. Octal number format is used to describe the binary string representation of q^*. Fig. 5 shows the configurations of binary manipulator for each case. The mark '△' and 'o' represent the target point and the end effector position respectively. In case (a), when the target point is out of the workspace, minimum error solution results in large error. For the target points of case (c) and (d) with higher density, minimum error solutions derive the end effector to the target points more accurately. The density for the case (c) is higher than (d), but the error is larger. Higher density implies that more accurate minimum error solution could be obtained statistically not deterministically.

6. Conclusions

New definitions of the workspace and the point density for the inverse kinematic problem were introduced. The point density represents the ability of the manipulator to move smoothly in the neighborhood of a target point, as well as the probability that an arbitrary joint vector drives the end effector within a ball. A genetic algorithm was proposed as a method to solve an inverse kinematic problem of a binary manipulator. The adopted genetic algorithm produces the minimum error solution as well as relative density of the workspace for a given target point. Efficiency of the genetic algorithm was demonstrated by numerical examples.

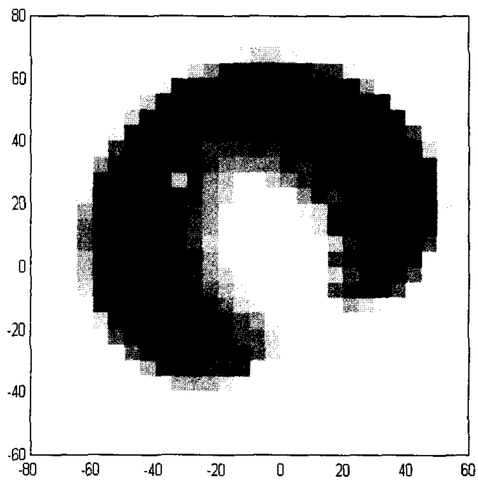

Fig. 4 Density map of the workspace ($\delta=2.5$)

Table 1 Results of the inverse kinematic analysis

case	target point	count n	error	joint state (q^*)
a	(2.5,7.5)	0	5.339	3331111114_8
b	(-7.5,-7.5)	76	0.347	4444444667_8
c	(22.5,42.5)	2293	0.078	1402232335_8
d	(-7.5, 37.5)	748	0.036	4230113144_8

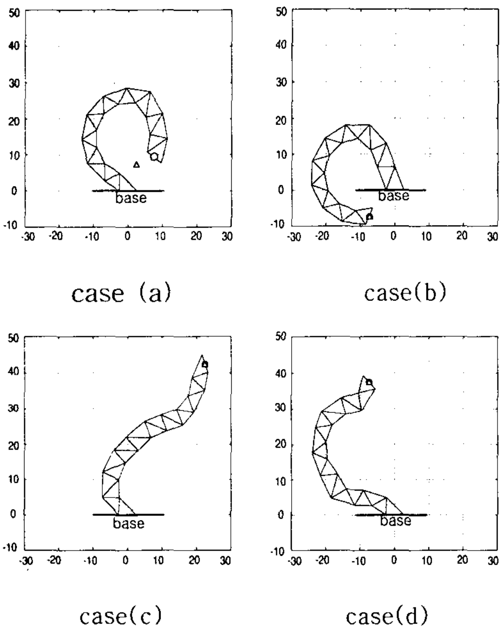

Fig. 5 Configurations of the binary manipulator

References

1. B. Padmanabhan, V. Arun and C. F. Reinholtz, "Closed-Form Inverse Kinematic Analysis of Variable-Geometry Truss Manipulators," Trans. of the ASME J. of Mechanical. Design, Vol. 114, Sep., pp. 438-443, 1992.
2. M. Subramaniam and S. N. Kramer, "The Inverse Kinematic Solution of the Tetrahedron Based Variable-Geometry Truss Manipulator," Trans. of the ASME J. of Mechanical Design, Vol. 114, Sep., pp. 433-437, 1997.
3. G. S. Chirikjian, "Inverse Kinematics of Binary Manipulators Using a Continuum Model," J. of Intelligent and Robotic System, Vol. 19, pp. 5-22, 1997.
4. G. Ryu and I. Rhee, "A Study on the Inverse Kinematic Analysis of a Binary Robot Manipulator using Backbone Curve," J. of the KSPE, Vol. 16, No. 3, pp. 174-179, 1999.
5. I. Ebert-Uphoff and G. S. Chirikjan, "Efficient Workspace Generation for Binary Manipulators with Many Actuators," J. of Robotic Systems, Vol. 12, No. 6, pp. 383-400, 1995.
6. I. Ebert-Uphoff and G. S. Chirikjian, "Inverse Kinematics of Discretely Actuated Hyper-Redundant Manipulators using Workspace Densities," ICAR '96.
7. G. S. Chirikjian and Imme Ebert-Uphoff, "Numerical Convolution on the Euclidean Group with Applications to Workspace Generation," IEEE Trans. on Robotics and Automation, Vol. 14, No. 1, 1998.
8. Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 3rd Ed. Springer-Verlag, 1994.