A Kinematic Analysis of a Binary Robot Manipulator using Genetic Algorithms

Gilha Ryu[1] and Ihnseok Rhee[1]

[1] Department of mechanical engineering, Korea University of Technology and Education, South Korea

ABSTRACT

A binary parallel robot manipulator uses actuators that have only two stable states being built by stacking variable geometry trusses on top of each other in a long serial chain. Discrete characteristics of the binary manipulator make it impossible to analyze an inverse kinematic problem in conventional ways. We therefore introduce new definitions of workspace and inverse kinematic solution, and then apply a genetic algorithm to the newly defined inverse kinematic problem. Numerical examples show that our genetic algorithm is very efficient to solve the inverse kinematic problem of binary robot manipulators.

Keywords: binary robot manipulator, inverse kinematic analysis, genetic algorithm, workspace

1. Introduction

The traditional robot manipulators are actuated with continuous-range-of-motion actuators such as DC motors or hydraulic actuators. Compare to a traditional robot manipulator, a binary parallel robot manipulator uses actuators which have only two stable states and its structure can be built by stacking variable geometry truss on top of each other in a long serial chain. They present an important alternative to 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 have a very large degree of kinematic redundancy, and 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]. An inverse kinematics of a binary manipulator was indirectly solved by using backbone curve[3][4]. A backbone curves is a continuous curve describing the shape of binary manipulator. It can be generated by optimization scheme considering a curvature limit of a binary manipulator[4] The workspace of binary manipulator is not described by conventional way because of its discrete nature Ref. [5] introduced point density to describe the workspace. Also an effective algorithm using density map has been studied to compute a inverse kinematic solution of a binary robot manipulator[6][7].

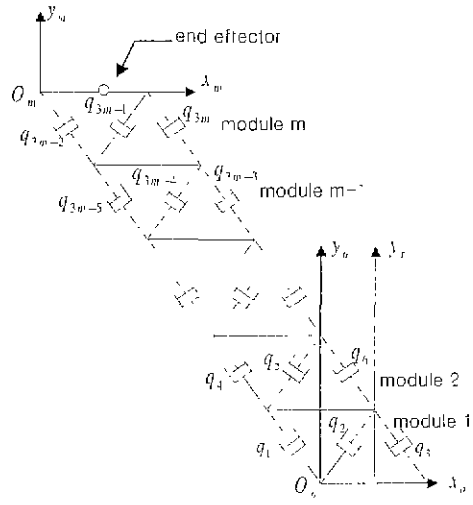


Fig. 1 Planar binary robot manipulator

In this paper, we solve an inverse kinematic problem by minimizing the end effector position error for given target position. A genetic algorithm is used as a minimization procedure. This algorithm is executed to obtain an inverse kinematic solution with a small error bound and density map of a workspace.

2. Kinematic Modeling

Fig. 1 represents a binary robot manipulator with m modules. Each actuator q_i has only two states, $l_{\min}$ and $l_{\max}$ so that the binary robot manipulator can reach 2^{3m} finite positions. The large number of possible positions of a binary robot manipulator prevents us from computing its inverse kinematic problem.

The basic truss module is composed of three linear binary actuators and two plates(upper and lower plate) as shown in Fig 2. For a kinematic analysis, we assign coordinate systems to each basic module. The origin of a moving coordinate system$\{i\}$ is located at the point D on upper plate and x-axis and y-axis are directed along and vertically to the upper plate respectively. The $\{0\}$ coordinate system is located on the base plate in the same way with its origin at A. The reference coordinate system $\{r\}$ is attached on the base plate and its origin is located at the center of base plate.

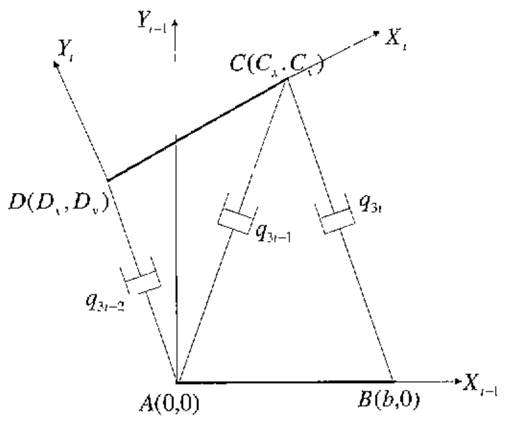


Fig. 2 Kinematic modeling of binary robot manipulator for i -th module

3. Kinematic Analysis

3.1 Forward kinematics

Two coordinate systems are attached on the upper and lower plate in a basic module mentioned before. The relationship between these two coordinate systems can be described by the following so called transformation matrix.

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

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

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

where

$$C_x = \frac{b^2 - q_{3i}^2 + q_{3i-1}^2}{2b}, \quad C_y = \sqrt{q_{3i-1}^2 - C_x^2}$$
$$D_x = \frac{D_y - k_1}{k_2}, \quad D_y = \frac{l_2 \pm \sqrt{l_2^2 - 4 l_1 l_3}}{2 l_1}$$
$$k_1 = \frac{q_{3i-1}^2 + q_{3i-2}^2 - b^2}{2 C_y}, \quad k_2 = -\frac{C_x}{C_y}$$
$$l_1 = \frac{1}{k_2^2} + 1, \ l_2 = \frac{2 k_1}{k_2^2}, \quad l_3 = \frac{k_1^2}{k_2^2} - q_{3i-2}^2$$

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.

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

3.2 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 a binary manipulator.

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

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

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

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

for all $\boldsymbol{q}$'s 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 how smoothly the binary manipulator is able to move in the neighbourhood of a target point as well as the probability that an end effector due to arbitrary joint vector lands 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

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

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 nonlinear programming or gradient method to minimization of equation (3) since q_i has only 2 states, $l_{\min}$ and $l_{\max}$.

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. Due to the binary nature, genetic algorithm is easily applied to solve the inverse kinematic problem of a binary manipulator

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

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

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

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

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. Figure 3 shows the procedures.

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

Fig. 3 A simple genetic algorithm

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 $\boldsymbol{q}$.

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

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 operator, crossover and mutation. We mate selected individuals randomly. For each pair, crossover takes place with crossover probability p_c. Crossover operation replaces a pair $\langle \overline{b}_{3m} \cdots \overline{b}_{c+1}\ \overline{b}_c \cdots \overline{b}_1 \rangle$, $\langle \underline{b}_{3m} \cdots \underline{b}_{c+1}\ \underline{b}_c \cdots \underline{b}_1 \rangle$ with a pair of offspring $\langle \overline{b}_{3m} \cdots \overline{b}_{c+1}\ \underline{b}_c \cdots \underline{b}_1 \rangle$, $\langle \overline{b}_{3m} \cdots \overline{b}_{c+1}\ \underline{b}_c \cdots \underline{b}_1 \rangle$

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 changing a 1 to a 0 and vice versa. Every bit of indivisual have 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 $\boldsymbol{q}^*$. 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 in the case that several target positions is involved, n for each target position gives us a knowledge about relative density.

5. Numerical Examples

In this section, examples are presented to show how useful genetic algorithms are to solve an inverse kinematic analysis for a binary robot manipulator. As an example, we consider a binary robot with $m = 10$, $l_{\max} = 7$, $l_{\min} = 5$ and $b = 5$. For genetic algorithms, n_p, w, p_c and p_m are chosen as 30, 5, 0.6 and 0.0333 respectively. Figure 4 shows a relative density map for $\delta = 2.5$. Target positions are located at the centers of square and equally spaced by 5 horizontally and vertically. We evolved 5000 generations for each target position to obtain the minimum error solution $\boldsymbol{q}^*$. In figure 4 Dark zone means high density and thin zone means low one. The higher density a target point has, the more accurate minimum error solution in terms of probability we can find.

Table 1 shows the minimum error solutions for some target points. n at 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 $\boldsymbol{q}^*$. Figure 5 shows the configurations of binary manipulator for each cases. The mark '△' represents target point. In case (a) for which the target point is out of work space, minimum error solution results large error. For the targets point of case (c) and (d) which have relatively high density, minimum error solutions derive the end effector to target point very accurately.

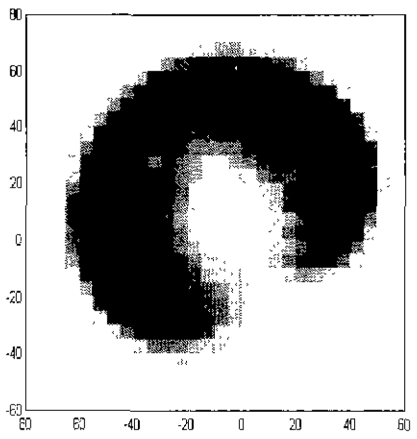
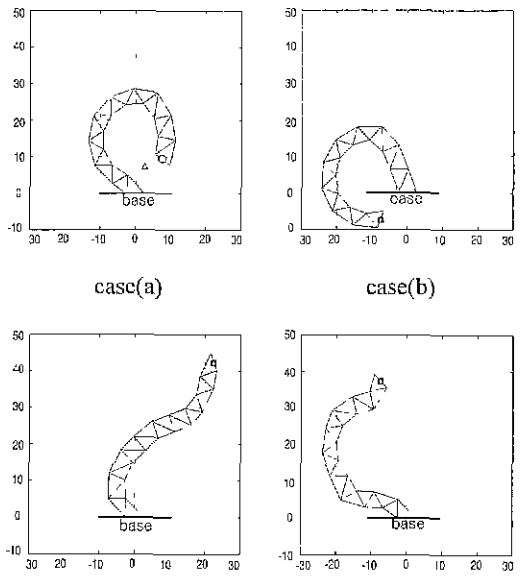


Fig. 4 A density map of a workspace

Table 1. Results of an 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



case(a) case(b)

case(c) case(d)

Fig. 5 Configurations of a binary manipulator

6. Conclusion

A genetic algorithm has been proposed as a method of solving an inverse kinematic analysis of binary manipulators. Numerical examples show that the genetic algorithm is very efficient for the inverse kinematic problem of binary manipulators. The genetic algorithm used in this paper gives a joint vector producing very small errors for a given a target point in workspace as well as relative density of workspace.

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. 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.