Fuzzy Logic Control for a Redundant Manipulator

- Resolved Motion Rate Control

Sung-Woo Kim and Ju-Jang Lee

Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
Taejon, 305-701, Korea

Abstract

The resolved motion rate control (RMRC) is converting to Joint space trajectory from given Cartesian space trajectory. The RMRC requires the inverse of Jacobian matrix. Since the Jacobian matrix of the redundant robot is generally not square, the pseudo-inverse must be introduced. However the pseudo-inverse is not easy to be implemented on a digital computer in real time as well as mathematically complex. In this paper, a simple fuzzy resolved motion rate control (FRMRC) that can replace the RMRC using pseudo-inverse of Jacobian is proposed. The proposed FRMRC with appropriate fuzzy rules, membership functions and reasoning method can solve the mapping problem between the spaces without complexity. The mapped Joint space trajectory is sufficiently accurate so that it can be directly used to control redundant manipulators. Simulation results verify the efficiency of the proposed idea.

1. Introduction

Conventional robot controllers are designed in Joint space. However the desired trajectory is usually given in Cartesian space, so it is necessary to convert between two spaces. This is generally known as *kinematics* when Joint space is converted to Cartesian space and *inverse kinematics* when Cartesian space to Joint space. Because of the high nonlinearity of a robot mechanism, the closed form solution of kinematics is very complex and highly coupled and the one of inverse kinematics is much more complex.

Instead of solving kinematics and inverse kinematics, conversion between two spaces can be done by using the differential relationship, so-called Jacobian. The Jacobian matrix determines the relationship between change of linear position and change of angular position. Using this property, one can find the current Cartesian position (or Joint position) by integrating the change of linear position (or that of angular position). These can be represented mathematically as follows.

Generally, the kinematic relationship between Cartesian space and Joint space is given by

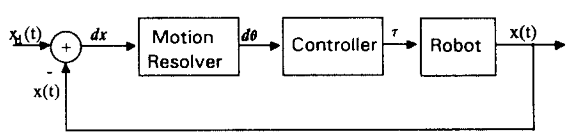
$$x = f(\theta) \tag{1}$$

Their differential relationships are given as follows:

$$dx = J(\theta) \cdot d\theta \tag{2}$$

where $J = \partial f / \partial \theta \in R^{m \times n}$ is the Jacobian matrix.

Computing dx by solving the linear equation of (2) for given θ and x is proposed by Whitney(1969) as the *resolved motion rate control*[1]. The block diagram of robot manipulator control containing the motion resolver is shown Fig. 1.



x(t) : current position in Cartesian space
$x_d(t)$: desired trajectory

Fig. 1. Robot controller containing the motion resolver

Conventional robot manipulators have square Jacobian matrix. Thus the solution of the linear equation of (2) is given by multiplying the inverse Jacobian matrix to both side of (2). However the redundant manipulators have more degree of freedom than the conventional manipulators, so their Jacobian is not a square one. Therefore the pseudo-inverse of J is introduced.

The general solution of (2) using the pseudo-inverse is obtained as follows.

$$d\theta = J^{\#}(\theta) \cdot dx + [I_n - J^{\#}(\theta) J(\theta)] y \qquad (3)$$

where $J^{\#}(\theta) \in R^{n\times m}$ is the pseudo-inverse of $J(\theta)$, $y \in R^n$ is an arbitrary vector and $I_n \in R^{n\times n}$ is and identity matrix. If the exact solution does not exist, (3) covers all least-square solutions that minimize $\|dx - J(\theta) d\theta\|$.

Even though the general solution of (2) is given by (3), finding out the pseudo-inverse of Jacobian matrix is not easy. It needs much more calculation than the conventional matrix inverse. Thus the resolved motion rate control using (3) has difficulties to implement on a digital computer in real time. To overcome the above drawbacks, a simple algorithm is proposed to replace the pseudo-inverse of the Jacobian matrix. That is, by solving (2) using the fuzzy logic, one can build simple and fast resolved motion rate controller.

In this paper, a fuzzy resolved motion rate controller (FRMRC) which converts Joint space to Cartesian space is proposed. It will be shown that conversion between two spaces can be done without solving inverse kinematics nor pseudo-inverse of Jacobian matrix. Moreover, by choosing the fuzzy variables and rules as small as possible, its structure is simplified. First the FRMRC for 1-link robot manipulator is derived, and then extended to multi-link redundant manipulators hierarchically. Finally the simulation results will verify the proposed algorithm.

2. Fuzzy Resolved Motion Rate Controller

The basic structure of fuzzy resolved motion rate controller (FRMRC) for redundant manipulators is like that the inverse Jacobian is replaced by the fuzzy reasoning. In other words, instead of calculating the pseudo-inverse of Jacobian matrix, we define fuzzy rules, membership functions, fuzzy reasoning method, and universe of discourse [Fig. 2].

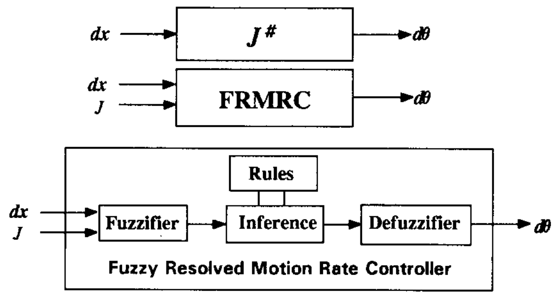


Fig. 2. Basic Structure of FRMRC

A. One-link Manipulator

To show how the rules and the membership functions can be determined, consider 1-link robot manipulator first [Fig. 3].

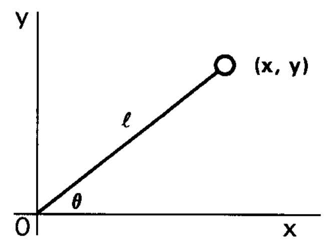


Fig. 3. 1-link robot manipulator

From the kinematics of the manipulator, the tip position (x,y) is given by

$$\begin{aligned} x &= l\cos\theta \\ y &= l\sin\theta \end{aligned} \qquad (4)$$

The Jacobian matrix is obtained by differentiating both sides of (4) w.r.t. time :

$$\begin{aligned} dx &= -l\sin\theta \cdot d\theta \\ dy &= l\cos\theta \cdot d\theta \end{aligned} \qquad (5)$$

So the Jaocibian matrix is $J = [-l\sin\theta \quad l\cos\theta]^T$.

The problem is to calculate $d\theta$ from the given dx, dy and θ.

$$\begin{aligned} (dx, \ -l\sin\theta) &\to d\theta \\ (dy, \ \ l\cos\theta) &\to d\theta \end{aligned} \qquad (6)$$

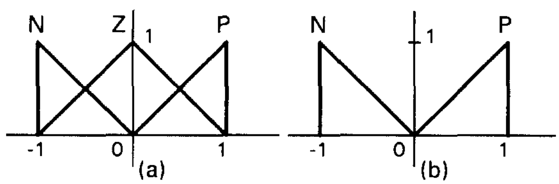
As mentioned before, this solution can be find by pseudo-inverse of the Jacobian matrix, but in this paper we solve this problem by fuzzy logic.

Consider the case $(dx, -l\sin\theta) \to d\theta$ and define fuzzy variables, $dx, d\theta, -l\sin\theta$ as follow.

$$
\begin{aligned}
dx &= \{N, Z, P\} \\
d\theta &= \{N, Z, P\} \\
-l\sin\theta &= \{N, P\}
\end{aligned}
\tag{7}
$$

where N, P and Z denote *Negative, Positive,* and *Zero*, respectively.

The membership functions corresponding the fuzzy variables are below.



(a) Membership functions of dx and $d\theta$
(b) Membership functions of J

Fig 4. Membership functions

Now define the fuzzy rules. From the relationships of (5), define 5 rules.

Rule 1 : If $-l\sin\theta$ is *Negative* and dx is *Negative*, then $d\theta$ is *Positive.*
Rule 2 : If $-l\sin\theta$ is *Positive* and dx is *Negative*, then $d\theta$ is *Negative.*
Rule 3 : If $-l\sin\theta$ is *Negative* and dx is *Positive*, then $d\theta$ is *Negative.*
Rule 4 : If $-l\sin\theta$ is *Positive* and dx is *Positive*, then $d\theta$ is *Positive.*
Rule 5 : If dx is *Zero,* then $d\theta$ is *Zero.*

The case $(dy, l\cos\theta) \to d\theta$ can be considered as the same manner. The corresponding fuzzy variables, membership functions and fuzzy rules are defined in analogy with (7).

$$
\begin{aligned}
dy &= \{N, Z, P\} \\
d\theta &= \{N, Z, P\} \\
l\cos\theta &= \{N, P\}
\end{aligned}
\tag{8}
$$

The two cases are rearranged in unified manner by defining new variables dr and J. When dealing the case $(dx, -l\sin\theta) \to d\theta$, dr and J is replaced by dx and $-l\sin\theta$, respectively and when $(dy, l\cos\theta) \to d\theta$, dr and J is by dy and $l\cos\theta$, respectively. Table 1 shows the results.

Table 1. Rule Table (5 rules)

$J \setminus dr$	N	Z	P
N	P	Z	N
P	N	Z	P

Now, calculating $d\theta$ from the above two cases. That is, $d\theta$ can be obtained by concerning the two cases $(dx, -l\sin\theta) \to d\theta$ and $(dy, l\cos\theta) \to d\theta$ properly. To satisfy these two conditions at the same time, we consider fuzzy OR logic, i.e.

$$(dx, -l\sin\theta) \text{ OR } (dy, l\cos\theta) \to d\theta$$

There are a lot of candidates of fuzzy OR(x, y) logic, for example, Zadhe OR = MAX(x, y) and Lukasiewicz OR = MIN(x+y, 1). In this paper, we prefer Zadhe OR.

B. Kinematic Redundant Manipulator

Based upon the results of 1-link manipulator, consider the 3-link redundant robot having kinematic redundancy.

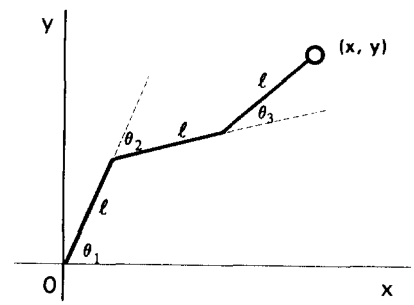


Fig. 4. 3-DOF redundant manipulator

Let C and S denote $\cos\theta$ and $\sin\theta$, respectively. From kinematics,

$$
\begin{aligned}
x &= lC_1 + lC_{12} + lC_{123} \\
y &= lS_1 + lS_{12} + lS_{123}
\end{aligned}
\tag{10}
$$

The differential relationships are,

$$
\begin{aligned}
dx &= J_{11}d\theta_1 + J_{12}d\theta_2 + J_{13}d\theta_3 \\
dy &= J_{21}d\theta_1 + J_{22}d\theta_2 + J_{23}d\theta_3
\end{aligned}
\tag{11}
$$

Thus Jacobian matrix, J is

$$J = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \end{bmatrix} \tag{12}$$

where

$$J_{11} = -lS_1 - lS_{12} - lS_{123} \qquad J_{21} = lC_1 + lC_{12} + lC_{123}$$
$$J_{12} = -lS_{12} - lS_{123} \qquad J_{22} = lC_{12} + lC_{123}$$
$$J_{13} = -lS_{123} \qquad J_{23} = lC_{123}$$

From (11) dx can be considered as superposition of $J_{1i}d\theta_i$ and dy can be of $J_{2i}d\theta_i$, where i=1,2,3. We can determine $d\theta_i$ by splitting (11) into 3 sub-parts. First, $d\theta_1$ is calculated from (13), the sub-equation of (11).

$$\begin{aligned} dx &= J_{11}d\theta_1 \\ dy &= J_{21}d\theta_1 \\ (dx, J_{11}) \text{ OR } &(dy, J_{21}) \rightarrow d\theta_1 \end{aligned} \tag{13}$$

Next, $d\theta_2$ is calculated from (14), the second sub-equation of (11).

$$\begin{aligned} dx' &= dx - J_{11}d\theta_1 = J_{12}d\theta_2 \\ dy' &= dy - J_{21}d\theta_1 = J_{22}d\theta_2 \\ (dx', J_{12}) \text{ OR } &(dy', J_{22}) \rightarrow d\theta_2 \end{aligned} \tag{14}$$

Finally, $d\theta_3$ is calculated from (15), the third sub-equation of (11).

$$\begin{aligned} dx'' &= dx' - J_{12}d\theta_2 = J_{13}d\theta_3 \\ dy'' &= dy' - J_{22}d\theta_2 = J_{23}d\theta_3 \\ (dx'', J_{13}) \text{ OR } &(dy'', J_{23}) \rightarrow d\theta_3 \end{aligned} \tag{15}$$

By splitting (11) into (13) to (15), we use the fuzzy rules, membership functions and reasoning method of 1-link manipulator derived in previous section without modification. Generally, even though the manipulator has n degree-of-freedom, we can use the same rule table and membership functions as 1-link manipulator. This algorithm can be generalized from the first link to the end link as considering the sub-equations hierarchically.

• **Hierarchical FRMRC algorithm**

$dx = x_d(t) - x(t)$
$dy = y_d(t) - y(t)$
Do while $1 \le k \le n$
  (dx, J_{1k}) OR $(dy, J_{2k}) \rightarrow d\theta_k$
  $dx = dx - J_{1k} \times d\theta_k$
  $dy = dy - J_{2k} \times d\theta_k$
EndDo

The hierarchical FRMRC is shown in Fig. 5 .

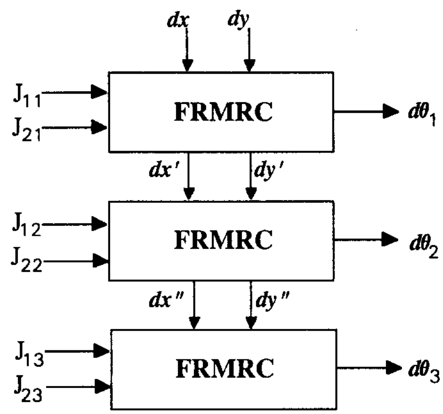


Fig. 5. Hierarchical Fuzzy Resolved Motion Rate Controller

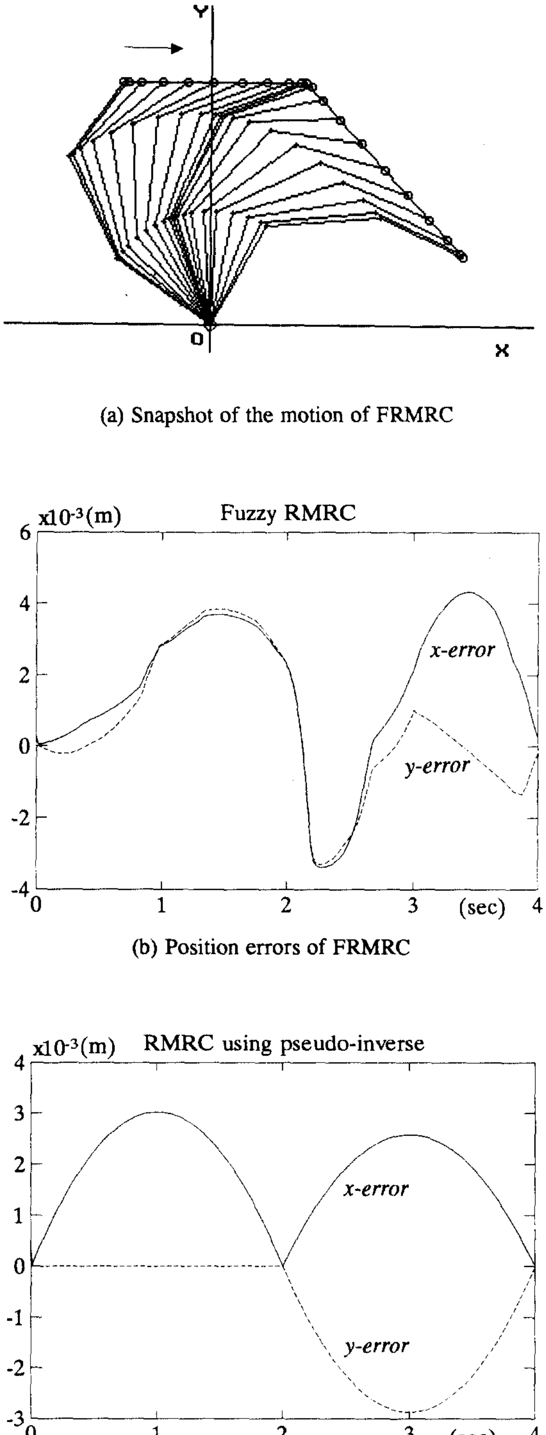
3. Simulation and Results

As derived previous section, simulate the proposed algorithm for a planar 3-DOF redundant manipulator. The link lengths l_1,l_2 and l_3 are 0.5, 0.5 and 0.4 meters, respectively.

The concerning motion trajectories are linear like Fig. 6 and circular like Fig. 7. For the linear motion, the desired Cartesian space trajectory is given by cubic polynomial.

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{16}$$
$$x(0) = x_0 \qquad \dot{x}(0) = \dot{x}_0$$
$$x(t_f) = x_f \qquad \dot{x}(t_f) = \dot{x}_f$$

The FRMRC and the conventional RMRC using pseudo-inverse of Jacobian matrix are compared. In Fig. 6 and Fig. 7, the snapshots of the motion of the manipulator are for the case of FRMRC. The simulation results show the position errors of x *and* y for both FRMRC and RMRC. We see that the errors are approximately same and both have zero error at the final goal. In Fig. 7, circular trajectory of 3-cycle is represented. According to the results, we see the error does not blow up as the manipulator tracks cyclic trajectory.

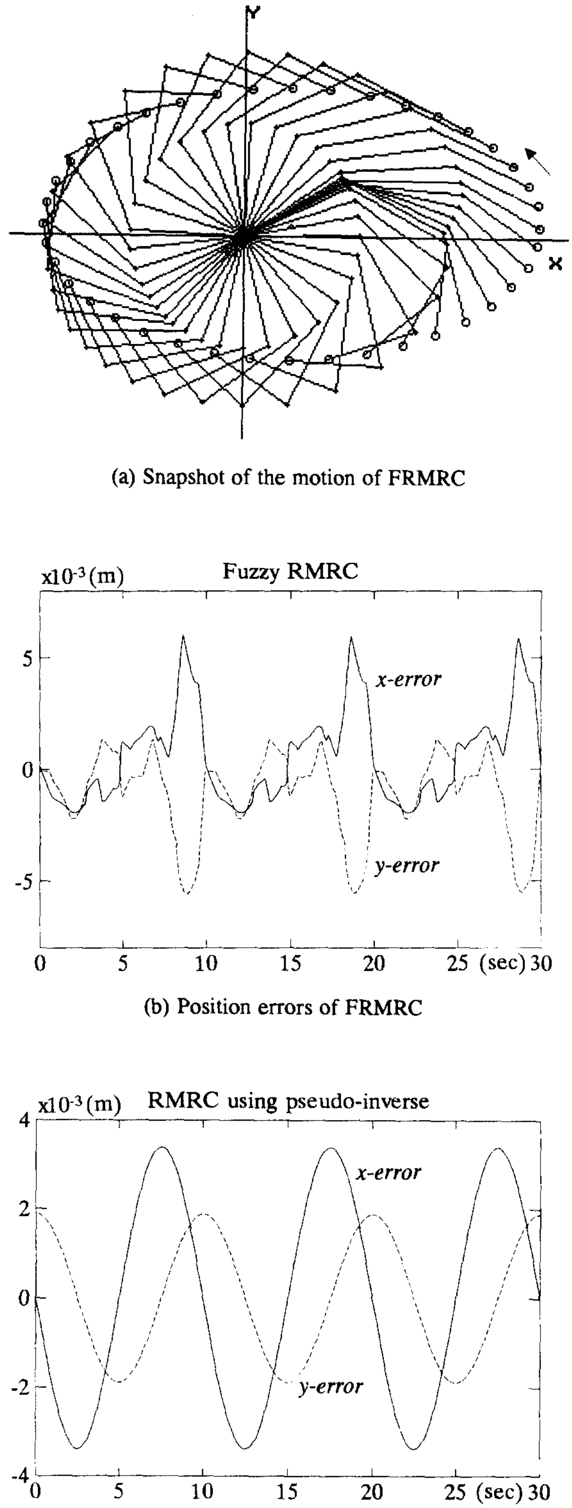


(a) Snapshot of the motion of FRMRC

(b) Position errors of FRMRC

(c) Position errors of RMRC

Fig. 6. FRMRC and RMRC for linear motion



(a) Snapshot of the motion of FRMRC

(b) Position errors of FRMRC

(c) Position errors of RMRC

Fig. 7. FRMRC and RMRC for circular motion

4. Conclusions

In this paper, We have proposed a simple resolved motion rate control algorithm using fuzzy logic and fuzzy inference, and have showed that this can replace the pseudo-inverse of the Jacobian matrix. The advantages' of this fuzzy motion resolving algorithm are summarized below.

- Complexity and computation burdens are excluded, since it is sufficient to use only 5 rules and 2 (or 3) membership functions for each fuzzy variable.
- This algorithm can be applied not only a 3-DOF manipulator but also multi-DOF robots using the hierarchical FRMRC. Therefore even though the manipulator has n degree-of-freedom, we can use the same rule table and membership functions as 1-link case.
- If the fuzzy reasoning method is realized by look-up table, then the FRMRC can be implemented in real time by a digital computer.

The proposed fuzzy motion resolving algorithm is not more accurate than that of using the pseudo-inverse, but we can see these two methods have same order of error. This inaccuracy can be overwhelmed by the above advantages, simplicity and rapidness. Further studies are to use the proposed FRMRC to control redundant robots and to extend to avoiding obstacles.

References

1. D. E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses," *IEEE Trans. Man, Machine Sys.* MMS-10 (2), pp47-53

2. Y. Nakamura, *Advanced Robotics -- Redundancy and Optimization,* Addison-Wesley, 1991

3. A. Nedungadi, "A Fuzzy Robot Controller -- Hardware Implementation," *IEEE International Conf. on Fuzzy Sys., 1992,* pp1325-1331

4. M. Sugeno, *Industrial Application of Fuzzy Control,* Elesevier Science Publishers, 1985

5. C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller -- Part I/II, " *IEEE Trans. on Sys. Man and Cyb.* Vol. 20, No. 2, pp404-435, March/April, 1990.