A Fuzzy Logic Positioning System for an Articulated Robot Arm

Juan Martinez, John Bowles, and Patrick Mills
Electrical and Computer Engineering
University of South Carolina
Columbia, SC 29208

Abstract
Articulated robot arms offer maximum positioning flexibility but suffer from complex kinematics. In most applications, linear motion is desirable. Calculating the kinematic equations which govern an articulated arm is straightforward; however, it is generally difficult to calculate the inverse kinematic equations that are needed to position the arm in closed form. Using a fuzzy reasoning system, it is possible to accurately position an articulated arm without explicitly solving the inverse kinematic equations.

1. Introduction

Manipulator arms which mimic the human arm are called articulated arms. Articulated arms have several important advantages over other configurations: they can reach over or under objects, cover a large work space relative to the volume of arm, and have maximum flexibility. However, these advantages come at the cost of more complex kinematic models: difficulty in controlling linear motion, and difficulty in visualizing exactly how the arm should move from one point to another [1].

The kinematic equations which govern an articulated arm convert joint coordinates into 3-dimensional world coordinates. Using basic geometric relationships these equations are generally easy to express in closed form. But the inverse kinematic equations, which convert from world coordinates to joint coordinates, often cannot be obtained in closed form. In order to position the arm, the control system must be able to convert from the world coordinates to the corresponding joint coordinates. In effect, it must solve the inverse kinematic equations, sometimes called the arm equations.

In the cases where the arm equations cannot be found or where they are too complex to calculate in real time, heuristics must be used to estimate the joint coordinates based on information about the arm configuration. A fuzzy reasoning system provides an excellent platform for dealing with the ambiguity present in such heuristics.

2. System Implementation

The experimental system implementation consists of a personal computer, a Motorola 68HC11 microprocessor, [2, 3] and the articulated arm. The personal computer provides a graphical user interface which controls the desired arm position and motion. Position is specified in spherical coordinates which are provided as inputs to the fuzzy control system. The control system gives crisp joint coordinates as outputs. The joint coordinates are then sent to the HC11 which is responsible for actually positioning the arm. The arm consists of three servo motors (waist, shoulder, and elbow) and three solid links as shown in Figure 1. The HC11 outputs the time intensive wave form necessary to position the servos accurately.

It is important to note that the arm runs "open loop." There are no position sensors or other feedback mechanisms to aid in positioning the arm. Instructions are sent to the servo motors which then move the arm to its new position. The positioning accuracy depends on
the accuracy with which the joint coordinates are computed and the resolution of the servo motors (approximately .7 degrees) in moving the arm.

Figure 2 shows the articulated arm in the zero position. The direction of positive rotation for each joint is also shown. $\theta_1$ is the angle of rotation for the waist, $\theta_2$ for the shoulder, and $\theta_3$ for the elbow. Each servo motor can turn through an angle of about $190^\circ$. In the implementation, each servo (joint) is limited to an angle of rotation of $180^\circ$. Each joint is able to reach 256 discrete points evenly distributed across the axis of rotation. In our implementation, the arms are all of unit length ($L_1 = L_2 = L_3 = 1$) and the arm can reach points in the (approximately) half-sphere having the cross-sectional profile shown in Figure 3. If $L_3 < L_2$, the arm would not be able to reach back to the vertical axis, and the workspace envelop would consist of two concentric half-spheres.

3. Control System Kinematic Equations

For the articulated arm presented here, the kinematic equations, in Cartesian coordinates, are:

$$
x = \cos(\theta_1) [L_2 \cos(\theta_2) + L_3 \cos(\theta_2 + \theta_3)]$$
$$y = \sin(\theta_1) [L_2 \cos(\theta_2) + L_3 \cos(\theta_2 + \theta_3)]$$
$$z = L_2 \sin(\theta_2) + L_3 \sin(\theta_2 + \theta_3) + L_1$$

As can be seen from examination of these equations, derivation of the arm equations (i.e., solving for $\theta_1$, $\theta_2$, and $\theta_3$ in terms of $x$, $y$, and $z$) is a non-trivial endeavor. In some configurations, it may not be possible to solve the equations in closed form. In any case, solving for the joint coordinates using either a numeric procedure or a heuristic, if they are available, is a time consuming process.

To control the arm, we begin by observing (Figure 4) that the point $(x, y, z)$ must lie in the plane defined by the vertical axis and the arm. Thus, the waist angle, $\theta_1$ is determined by $\tan(\theta_1) = y/x$ or $\theta_1 = \arctan(y/x)$. Within the plane the point $(x, y, z)$ can be described by the spherical coordinates $\rho$ and $\varphi$, where $\rho$ is the distance from the origin to the end of the arm and $\varphi$ is the inclination angle from the horizontal to the vector $\rho$.

This gives the kinematic equations in spherical coordinates:

$$\theta = \theta_1 = \arctan\left(\frac{y}{x}\right) \quad (1a)$$

$$\varphi = \arctan\left(\frac{z}{\sqrt{x^2 + y^2}}\right)$$
$$= \arctan\left[\frac{L_2 \sin(\theta_2) + L_3 \sin(\theta_2 + \theta_3) + L_1}{L_2 \cos(\theta_2) + L_3 \cos(\theta_2 + \theta_3)}\right] \quad (1b)$$

$$\rho = \sqrt{x^2 + y^2 + z^2} \quad (1c)$$
Figure 4. Spherical coordinate system for the articulated arm.

Here $\theta$ is the angle of rotation horizontally, $\phi$ is the angle of inclination, and $\rho$ is the distance from the origin. From equation (1a) the waist angle, $\theta_1$ is found trivially and the 3-dimensional positioning problem is reduced to a 2-dimensional problem. When the arm is in the vertical position ($\phi = \pi/2$) the denominator in equation (1b) is 0 and the expression for arc tan is undefined. Thus, for the two dimensional positioning problem we use the alternative formulation:

$$
\rho = \left[ (L_2\cos(\theta_2) + L_3\cos(\theta_2 + \theta_3))^2 + (L_2\sin(\theta_2) + L_3\sin(\theta_2 + \theta_3 + L_1))^2 \right]^{1/2}
$$

$$
\phi = \arccos \left( \frac{x^2 + y^2}{\rho} \right)
$$

$$
= \arccos \left( \frac{L_2\cos(\theta_2) + L_3\cos(\theta_2 + \theta_3)}{\rho} \right)
$$

Observe that $\rho$ has its maximum value of 3 when the arm is fully extended vertically ($\theta_2 = \pi/2, \theta_3 = 0$).

Again, finding $\theta_2$ and $\theta_3$ given $\rho$ and $\phi$, is generally difficult. However, we can easily compute $\rho$ and $\phi$ given $\theta_2$ and $\theta_3$. Figure 5a shows contours of constant $\theta_2$, constructed by evaluating the expressions in equation 2 for $\rho$ and $\phi$ while holding $\theta_2$ constant and rotating $\theta_3$ through its full range of values ($\theta_3 \in [0, \pi]$). The contours shown are for $\theta_2$ ranging from $0^\circ$ to $180^\circ$ in increments of $20^\circ$. Similarly, Figure 5b shows contours of constant $\theta_3$ from $0^\circ$ to $180^\circ$ in increments of $20^\circ$. Although no closed form solution exists, these contours represent the solution to the inverse kinematic equations for a given $\rho$ and $\phi$.

4. Fuzzy Positioning System

An alternative to explicitly solving the arm equations is to model the arm solution in Figure 5. Although the arm solution is highly non-linear, a good approximation can be obtained. The objective is to identify points in ($\rho, \phi$) where the arm solution is known and use those points as markers from which the fuzzy rule base for the positioning system can be developed.

The overall structure of the fuzzy positioning system is shown in Figure 6. Input variables, provided by the

![Figure 6. Fuzzy positioning system.](image)
user, give the location, in spherical coordinates, that the arm is to move to. These inputs are first fuzzified to represent the coordinates as members of a set of fuzzy input sets. The fuzzy inputs are then matched against the corresponding rules in the rule base and the results defuzzified to give the crisp outputs to the arm servo motors.

In the sequel, we illustrate the operation of the fuzzy positioning system for the inputs: \( \theta = 50^\circ, \varphi = 100^\circ \), and \( \rho = 2.6 \). As noted above, the waist angle \( \theta_1 \) is found immediately: \( \theta_1 = \theta = 50^\circ \).

### 4.1 Fuzzification

The first step in designing the rule base is to create membership sets which describe the inputs. In this case, there are 2 input variables: \( \rho \) and \( \varphi \). (\( \theta_1 = \theta \) is assumed to be already known.) The degree of accuracy of the model will be strongly influenced by these set descriptions.

The variable \( \varphi \) describes an arc of approximately 25° to 180° for the given arm configuration. Observe from Figure 5b that the contours of constant \( \theta_3 \) are closest together, indicating a rapid change with respect to \( \rho \), for large values of \( \rho \). This is shown in Figure 7 which shows \( \theta_3 \) versus \( \rho \) for a cross section of Figure 5b at \( \varphi = 100^\circ \).

To approximate this type of surface with triangular fuzzy sets (which gives a piecewise linear approximation to the surface), it is best to have fuzzy sets that are more closely spaced where the change in \( \theta_3 \) is varying and more widely spaced where \( \theta_3 \) is changing linearly. As shown in Figure 7 the rapid change in \( \rho \) starts at about \( \rho = 2.7 \) at \( \varphi = 165^\circ \), \( \rho \) starts to change rapidly at about \( \rho = 1.8 \). Hence we approximate \( \rho \) with the fuzzy sets shown in Figure 8a.

Also observe in Figure 5b, for circles of constant \( \rho \), the value of \( \theta_3 \) for \( \varphi \) near 90° changes relatively little with respect to changes in \( \varphi \). Near the ends of the range, \( \theta_3 \) changes rather quickly for changes in \( \varphi \). Thus we chose to represent \( \varphi \) with fuzzy sets that are more closely spaced at the extremes than near the 90° point. These
sets are shown in Figure 8b.

Similarly, observe from Figure 5a that for values of $\phi$ greater than approximately 80° the contours of constant $\theta_2$ are approximately evenly spaced for constant $\rho$. For $\phi < 80°$ the rate of change of $\theta_2$ is somewhat less (the contours are spaced farther apart). Thus, we chose to represent $\theta_2$ with the fuzzy sets shown in Figure 9b. (More can be used for greater accuracy.) Although the effect is not as pronounced as in Figure 5b, we again observed that $\theta_2$ changes most rapidly for large and small values of $\rho$. Thus, again we use fuzzy sets that are closely spaced for small and large values of $\rho$ and relatively widely spaced for $\rho$ in the middle of its range to best approximate this curve behavior. These sets are shown in Figure 9a.

Example: Matching the values given for $\phi$ and $\rho$ against the fuzzy sets in Figures 8 and 9 we find the corresponding fuzzy inputs:

For $\theta_2$ (Fig. 9):

$\phi = 100° \Rightarrow 0.33$ in M and 0.66 in ML;
$\rho = 2.6 \Rightarrow 0.6$ in L and 0.4 in VL

(3a)

4.2 Fuzzy Rule Base

Figure 10 shows the fuzzy rule base in the form of a Fuzzy Associative Memory (FAM). Potentially, 90 rules are needed to control $\theta_2$ and 108 rules for $\theta_3$; however, due to physical constraints only 40 rules are actually required for $\theta_2$ and 42 rules for $\theta_3$. Observe that the rule output is a crisp number rather than a linguistic term referring to a fuzzy set (often singleton valued). This simplifies the rules and enables the rule base to be constructed directly from Figure 5. For example, rule at the FAM location (L,M) in Figure 10a is read as:

\[
\text{if } \rho = \text{Medium and } \rho = \text{Large, then } \theta_2 = 49°
\]

is derived by examining Figure 5a for $\phi = 90°$ and $\rho = 2.5$. Note that these are the respective nominal values (when viewed as fuzzy numbers) of the fuzzy sets M for $\phi$ in Figure 9a and L for $\rho$ in Figure 9b. If linguistic terms and singleton fuzzy output sets had been used, tuning the

\[
\begin{array}{|c|c|c|c|c|c|c|c|c|}
\hline
\theta_2 & \text{VS} & S & MS & M & ML & L & VL & XL & XVL & MAX \\
\hline
\text{XL} & & & & & & & & 90 & & \\
VL & 35 & 61 & 89 & 120 & & & & & & \\
L & 10 & 20 & 49 & 75 & 100 & 135 & & & & \\
M & 10 & 35 & 65 & 95 & 125 & 160 & & & & \\
S & 25 & 40 & 60 & 90 & 120 & 160 & & & & \\
VS & 10 & 60 & 90 & 120 & 145 & 165 & & & & \\
XS & 0 & 115 & 120 & 135 & 150 & 165 & 180 & & & \\
XVS & & 145 & 145 & 150 & 160 & 170 & & & & \\
XXS & & & & & & & & & & \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|c|c|c|c|}
\hline
\theta_3 & XVS & XS & VS & S & MS & M & ML & L & VL & XL & XVL & MAX \\
\hline
\text{XL} & & & & & & & & 0 & & & & \\
VL & 20 & 50 & 58 & 45 & 30 & & & & & & \\
L & 35 & 60 & 80 & 83 & 74 & 70 & 40 & & & & \\
M & 100 & 105 & 100 & 90 & 70 & 30 & & & & & \\
S & & 125 & 115 & 105 & 90 & 70 & & & & & \\
VS & & 155 & 150 & 135 & 115 & 100 & 80 & & & & \\
XS & & 180 & 160 & 150 & 140 & 120 & 110 & 90 & & & \\
XVS & & 155 & 150 & 145 & 130 & 115 & & & & & \\
XXS & & & & & & & & & & & & \\
\hline
\end{array}
\]

Figure 10. Fuzzy Associative Memory (FAM) for $\theta_2$ and $\theta_3$ as functions of $\phi$ and $\rho$. 255
controller to get the most accurate positioning of the arm would have resulted in the linguistic output for the previous rule being assigned the numerical value $45^\circ$. Hence, by using the contours in Figure 5 to construct the rule base directly, considerable tuning of the controller and adjusting of the fuzzy sets was avoided.

The "degree of truth" of the rule output is the minimum of the degrees of truth of the rule antecedents. For example, with the given inputs, the output of the above rule will have the degree of truth $\min(0.33, 0.6) = 0.33$.

Example: For the given inputs, equation (3), the following rules are activated; the degree of truth of the output is also shown:

For $\theta_2$:

\[
\begin{align*}
\text{If } \phi = \text{Medium} \text{ and } \rho = \text{Large}, & \quad \text{then } \theta_2 = 49^\circ; \\
\text{degree of truth} & = 0.33 \\
\text{If } \phi = \text{Medium} \text{ and } \rho = \text{VeryLarge}, & \quad \text{then } \theta_2 = 61^\circ; \\
\text{degree of truth} & = 0.33 \\
\text{If } \phi = \text{MLarge} \text{ and } \rho = \text{Large}, & \quad \text{then } \theta_2 = 75^\circ; \\
\text{degree of truth} & = 0.6 \\
\text{If } \phi = \text{MLarge} \text{ and } \rho = \text{VeryLarge}, & \quad \text{then } \theta_2 = 89^\circ; \\
\text{degree of truth} & = 0.4 
\end{align*}
\]

For $\theta_3$:

\[
\begin{align*}
\text{If } \phi = \text{Medium} \text{ and } \rho = \text{Large}, & \quad \text{then } \theta_3 = 83^\circ; \\
\text{degree of truth} & = 0.4 \\
\text{If } \phi = \text{Medium} \text{ and } \rho = \text{VeryLarge}, & \quad \text{then } \theta_3 = 58^\circ; \\
\text{degree of truth} & = 0.4 \\
\text{If } \phi = \text{MLarge} \text{ and } \rho = \text{Large}, & \quad \text{then } \theta_3 = 74^\circ; \\
\text{degree of truth} & = 0.5 \\
\text{If } \phi = \text{MLarge} \text{ and } \rho = \text{VeryLarge}, & \quad \text{then } \theta_3 = 45^\circ; \\
\text{degree of truth} & = 0.4 
\end{align*}
\]

4.3 Defuzzification

The fuzzy outputs are defuzzified using the "weighted averages" technique commonly used for singleton output sets (This is the same as the center of gravity technique applied to singleton fuzzy sets [4, p. 336]):

\[
Z = \frac{\sum_{i=1}^{n} \mu_i z_i}{\sum_{i=1}^{n} \mu_i} \tag{5}
\]

where $z_i$ is the output of the ith rule and $\mu_i$ is its degree of truth. The resulting crisp output matches the control surface exactly at each of the marker points and provides a linear interpolation of the control surface between these points.

Example: The outputs in equation (4), defuzzified according to equation (5) yield:

\[
\begin{align*}
\theta_2 & = 70.42^\circ \\
\theta_3 & = 66.5^\circ 
\end{align*}
\]

Hence, for the input: $\theta = 50^\circ$, $\phi = 100^\circ$, and $\rho = 2.6$ the solution to the inverse kinematic equations, found by the fuzzy positioning system is:

\[
\begin{align*}
\theta_1 & = 50^\circ \\
\theta_2 & = 70.42^\circ \\
\theta_3 & = 66.5^\circ 
\end{align*}
\]

To verify the accuracy of the system, we can substitute these values into equations (1a), (1b), and (1c), yielding:

\[
\begin{align*}
\theta & = 50^\circ \\
\phi & = 98.6^\circ \\
\rho & = 2.65 
\end{align*}
\]

The fuzzy positioning system provides an effective solution for controlling the articulated arm. The accuracy of the result depends critically on the marker point values in the FAM, Figures 10a and 10b, and only marginally on the definition of the input fuzzy sets. At these points the solution should be exact. For other points the solution is found by interpolating between the values at the marker points. If the fuzzy membership functions are changed, the accuracy of the interpolation may vary, but generally not by a great deal; if the values at the marker points are in error, gross errors in the positioning of the arm may occur.

5. Conclusion

Articulated arms allow maximum flexibility approaching that of the human arm. The major problem with articulated arms in the past has been the difficulty in calculating the solution to the inverse kinematic equations. However, the fuzzy positioning system developed here shows that a simple representation can be used in some applications.

The ability to control linear arm motion without solving for the inverse kinematic equations allows more complex articulated arms to be utilized. In a test model, the fuzzy positioning system proved to be extremely fast, flexible, and easy to maintain. Changes in the arm configuration require changes in the rule base, but such changes are easily made off-line.
References