Dynamic Programming for Fuzzy Systems
with Fuzzy Environment

J. F. BALDWIN AND B. W. PILSWORTH

Department of Engineering Mathematics, University of Bristol,
Bristol BS8 1TR, United Kingdom

Submitted by L. Zadeh

A dynamic programming functional equation for a multi-stage decision problem with fuzzy dynamics and environment is formulated and solved by a process of fuzzy interpolation. This is an extension of the Bellman–Zadeh model.

1. INTRODUCTION

Human intelligence is often superior to existing machine intelligence for solving multi-stage decision problems which are of such a complex nature that finding an algorithmic approach based on classical mathematics for their solution is not feasible. The chess problem is an example where the complexity arises because of the high state dimensionality and the large number of possible decisions which are possible at each stage of the game. The rules and goals of the game are well defined but it is this microscopic precision which prevents a mathematical solution being obtained since it gives rise to a computationally impossible analysis. Human intelligence uses a much less precise model replacing microscopic precisional description with a more macroscopic fuzzy one. Thus a different state space is used (one of much lower dimension) and state mappings become fuzzy. This introduction of simplified description, taking into account only what is really relevant, allows human intelligence to gain insights into the chess problem resulting in a high standard of decision making, even though the decision policies will necessarily be fuzzy in nature.

For other problems, for example, political decision making, economic planning, domestic and other human problem solving, the state mappings from stage to stage will only be known in an imprecise manner. We all feel that we are able to choose the best car route when going on holiday and this is arrived at using a very fuzzy decision analysis. Nevertheless, human intelligence is limited to relatively simple chains of argument and often the analysis of complex multi-stage decision problems is over-simplified. This
motivates the need for a dynamic programming method for systems with
dfuzzy state mappings, fuzzy constraints and fuzzy goals.

In this paper we derive a dynamic programming functional equation for a
multi-stage decision process in which the state mapping from one stage to
the next is defined by a fuzzy automata acting in a fuzzy environment where
both control constraints and state goals are fuzzy in the sense of Zadeh [1].
It is thus an extension of the dynamic programming formulation given by
Bellman and Zadeh [2] to the case of fuzzy dynamics. For this purpose a
modified objective criterion is used, namely, a "truth function" is defined
which, broadly speaking, represents the truth that the goals and constraints
are satisfied. This reduces to Bellman's treatment for the case of non-fuzzy
state mappings.

The resulting functional equation cannot, in general, be solved exactly
owing to the high dimensional state and an approximate method of solution
is given which uses a new concept of "fuzzy interpolation." This basically
explores the solution for a given set of reference fuzzy subsets of state space
to produce conditional fuzzy statements of the IF ... THEN form. These are
then used with the modus ponens composition rule of inference given by
Zadeh [3] to induce a solution for other fuzzy subsets of state space.
Reference control subsets are also used to construct a control policy at each
stage.

A future paper will discuss conditions whereby an exact solution to the
functional equation may be obtained. Further, the treatment here can be
extended to the infinite horizon case and a policy space-iteration method
introduced. In addition it is easy to extend this treatment to include a
probabilistic model.

2. Bellman's model

See Bellman [2].

Consider the finite deterministic automaton \( V = \{U, X, \delta\} \), where \( U, X \) are
finite sets called control and state spaces, respectively, and \( \delta: X \times U \rightarrow X \).
The state equation is

\[
x(t + 1) = \delta(x(t), u(t)), \quad t = 0, 1, \ldots, T - 1,
\]

where \( T \) is the final time and \( x(0) \in X \) is the initial state. A fuzzy control
constraint, \( \chi \), is a fuzzy subset of \( U \) defined by the membership function
\( \chi: U \rightarrow [0, 1] \) and a fuzzy state goal, \( \chi' \), is a fuzzy subset of \( X \) defined by the
membership function \( \chi': X \rightarrow [0, 1] \).

We suppose then the existence of fuzzy control constraints
\( \{\chi_0, \chi_1, \ldots, \chi_{T-1}\} \), where \( \chi_i \) is relevant to the control input \( U_i \) at time \( i \),
\( 0 < i < T - 1 \).
Suppose also that a fuzzy goal $x_T$ is imposed on the final state $x_T$.  
A fixed input sequence $\{u_0, u_1, ..., u_{T-1}\}$ corresponds to the fuzzy decision  
\[ \in U^T = U \times U \times \cdots \times U \]  
given by the fuzzy interaction equation:  
\[ x(u_0, u_1, ..., u_{T-1}) = x_0(u_0) \land x_1(u_1) \cdots x_{T-1}(u_{T-1}) \land x_{T}'(x_T), \]
where we have written $u_i = u(i)$ and $x_T = x(T)$, and further $x(T)$ is calculated for a given $x(0) \in X$ from the state equation.

A maximum decision, in the sense of Negoita and Ralescu [4] for this problem is thus given by $(\bar{u}_0, \bar{u}_1, ..., \bar{u}_{T-1}) \in U^T$ such that
\[ \chi(u_0, u_1, ..., u_{T-1}) = \max_{u_0, ..., u_{T-1} \in U} \chi(u_0, u_1, ..., u_{T-1}). \]
Let $S_k(x) = \max_{u_k, ..., u_{T-1} \in U} x(u_k, u_{k+1}, ..., u_{T-1})$ when system starts in state $x$ at time $k$ and optimal control sequence is used. Then
\[ S_k(x) = \max_{u_k, ..., u_{T-1} \in U} [\chi_k(u_k) \land \chi_{k+1}(u_{k+1}) \land \cdots \land \chi_{T-1}(u_{T-1}) \land \chi_{T}'(x_T)], \]
where $x_T$ is calculated using $x_T = \delta(x_{r-1}, u_{r-1})$; $r = k + 1, ..., T$, with $x_k = x$, so that
\[ S_k(x) = \max_{u_k \in U} [\chi_k(u_k) \land (\max_{u_{k+1} \cdots u_{T-1}} \chi_{k+1}(u_{k+1}) \land \cdots \land \chi_{T-1}(u_{T-1}) \land \chi_{T}'(x_T) | x_{k+1} = \delta(x, u_k))], \]
so that
\[ S_k(x) = \max_{u_k \in U} [\chi_k(u_k) \land S_{k+1}(\delta(x, u_k))], \quad \text{for} \quad k = 0, 1, ..., T-1, \]

with $S_T(x) = \chi_T'(x)$.

The solution of this dynamic programming equation by backward iteration gives the maximum decision $(\bar{u}_0, \bar{u}_1, ..., \bar{u}_{T-1})$ and represents the solution of a multi-stage decision problem in a fuzzy environment but with deterministic state mappings.

We will extend this model to include fuzzy state mappings and thus make it applicable to a more general class of multi-stage decision process in which the system dynamics can only be described imprecisely or is conveniently described as such.
3. Fuzzy Mapping

A fuzzy mapping \( f: X \rightarrow Y \) is a fuzzy set on \( X \times Y \) with membership function \( \chi_f(x, y) \).

A fuzzy function \( f(x) \) is a fuzzy set on \( Y \) with membership function

\[ \chi_{f(x)}(y) = \chi_f(x, y). \]

Let \( A \) be a fuzzy subset on \( X \) defined by membership function \( \chi_A(x) \).

The fuzzy set \( f(A) \) on \( Y \) is a fuzzy mapping of a fuzzy set defined as

\[ \chi_{f(A)}(y) = \bigvee_{x \in X} (\chi_A(x) \land \chi_f(x, y)); \quad \text{all } y \in Y, \]

where \( \land \) stands for MIN and \( \lor \) for MAX.

4. A Fuzzy Decision Problem

Consider the finite fuzzy automaton \( V = \{U, X, \delta, F(U), F(X)\} \), where \( U, X \) are finite sets called control and state spaces respectively, and \( \delta: X \times U \rightarrow X \) and \( F(U), F(X) \) are the sets of fuzzy controls, states, respectively.

The state equation is given by

\[ \chi_{x(t+1)}(x(t+1)) = \chi_{h(v)} = \bigvee_{v \in V} (\chi_s(v) \land \chi_{\delta}(v, x(t+1))); \quad \text{all } x(t+1) \in X \text{ and } t = 0, 1, \ldots, T-1, \]

where \( V = X \times U; \ v = (x(t), u(t)); \ v \in V \) and \( V \) is a fuzzy set on \( V \) representing fuzzy state \( x(t) \) with fuzzy control \( u(t) \) having memberships function

\[ \chi_s(x, u) = \chi_{x(t)}(x) \land \chi_{u(t)}(u). \]

Suppose further the existence of fuzzy constraints \( \{\chi_0, \chi_1, \ldots, \chi_{T-1}\} \), \( \chi_i \in F(U) \), where \( \chi_i \) is imposed on the input \( u_t \), \( 0 \leq i \leq T - 1 \) and also that a fuzzy goal \( \chi_T \) is imposed on the final state \( x(T) \).

This is the optimisation problem dealt with above but modified to include fuzzy dynamics which replaces the deterministic state mappings.

5. One Stage Problem

As a special case, first consider a one stage decision problem, i.e., the fuzzy control \( u_0 \) is applied to the system in the fuzzy state \( x_0 \) with the constraint \( \chi_0 \) imposed on the control \( u_0 \) to produce a new state \( x_1 \) via the
fuzzy mapping $\delta$ with a goal constraint $\chi'_1$ imposed on $x_1$. Define a relational matrix $R$ such that

$$R(u, x) = \chi_0(u) \land \chi'_1(x); \quad u \in U; \ x \in X;$$

and this can be thought of as representing how well the control and goal constraints are satisfied. Thus for a fuzzy control $u_0$ and resulting fuzzy state $x_1$ we can form

$$T(u_0 R x_1) = u_0 \circ R \circ x_1,$$

where $\circ$ represents MAX-MIN composition as given by Zadeh [3] and Kaufmann [6]. This gives a measure of the truthness of "control and goal constraints satisfied" and this truth function has been discussed by Chang [5]. The decision criterion which we will use in this paper is to select the $u \in U^*$ which maximizes $T(u_0 R x_1)$ and this we will take as our optimal decision, where $U^*$ is the allowed set of fuzzy decisions. Thus

$$T(u_0 R x_1) = \bigvee_{x \in X} \left[ \left( \bigvee_{u \in U} (\chi'_1(x) \land \chi_0(u) \land \chi_{w_0}(u)) \right) \land \chi_{x_1}(x) \right]$$

and on using the distributivity property of $\land$ and $\lor$ over one another simplifies, as shown in Lemma 1 below, to

$$T(u_0 R x_1) = \left[ \bigvee_{x \in X} (\chi'_1(x) \land \chi_{x_1}(x)) \right] \land \left[ \bigvee_{u \in U} (\chi_0(u) \land \chi_{w_0}(u)) \right]$$

so that

$$T(u_0 R x_1) = (\chi'_1 \circ \chi_{x_1}) \land (\chi_0 \circ \chi_{w_0}).$$

**LEMMA 1.**

$$\bigvee_{i \in I} \left[ \bigvee_{j \in J} (A_i \land B_j \land C_j) \land D_i \right]$$

$$= \left[ \bigvee_i (A_i \land D_i) \right] \land \left[ \bigvee_j (B_j \land C_j) \right]$$

for $I = \{i_1, i_2, \ldots, i_m\}; \ J = \{j_1, j_2, \ldots, j_n\}$. 
Proof.

L.H.S. = \[ \bigvee_i \left( \left( A_i \land B_i \land C_{j_i} \right) \land \left( A_i \land B_{j_2} \land C_{j_2} \right) \lor \cdots \lor \left( A_i \land B_{j_n} \land C_{j_n} \right) \right) \lor D_i \]

= \[ \bigvee_i \left( \left( A_i \land \left( B_{j_1} \land C_{j_1} \right) \lor \left( B_{j_2} \land C_{j_2} \right) \right) \lor \left( \cdots \lor \left( A_i \land \left( \cdots \lor \left( A_i \land \left[ \left( B_{j_n} \land C_{j_n} \right) \right) \lor D_i \right) \right) \right) \right) \]

= \[ \bigvee_i \left( \left( A_i \land \left( B_{j_1} \land C_{j_1} \right) \lor \left( B_{j_2} \land C_{j_2} \right) \lor \cdots \lor \left( B_{j_n} \land C_{j_n} \right) \right) \lor D_i \right) \]

= \[ \left( A_i \land \left( \bigvee_j \left( B_j \land C_j \right) \right) \right) \lor \left( \left( A_i \land D_{i_1} \right) \lor \left( A_i \land D_{i_2} \right) \lor \cdots \lor \left( A_i \land D_{i_m} \right) \right) \]

= \[ \bigvee_i \left( A_i \land D_i \right) \right) \lor \left( \bigvee_j \left( B_j \land C_j \right) \right) = \text{R.H.S.} \quad \text{Q.E.D.} \]

6. Example

\[ X = \{x_1, x_2, x_3\}; \quad U = \{u_1, u_2\} \]

\[ x_1 \quad x_2 \quad x_3 \]

\[ \delta: \begin{array}{ccc}
  x_1 u_1 & 1 & 0.7 & 0.3 \\
  x_1 u_2 & 0.7 & 1 & 0.3 \\
  x_2 u_1 & 0.7 & 1 & 0.3 \\
  x_2 u_2 & 1 & 0.7 & 0.3 \\
  x_3 u_1 & 0.2 & 0.8 & 1 \\
  x_3 u_2 & 1 & 0.8 & 0.2 \\
\end{array} \]

\[ x_0: \begin{bmatrix} 0.3 \\ 0.8 \end{bmatrix}, \]

\[ x_1 \quad x_2 \quad x_3 \]

\[ x_1': \begin{bmatrix} 0.5 & 0.9 & 0.2 \end{bmatrix}. \]

\[ x_0 = 0.6/x_1 + 0.1/x_2 + 0.8/x_3, \]

\[ U^* = \{(0.2, 0.9); (0.7, 0.5)\}. \]

The \( x_1 \) resulting from \( u_0 = 0.2/u_1 + 0.9/u_2 \) is given by

\[ x_1(0.2, 0.9) = 0.8/x_1 + 0.8/x_2 + 0.3/x_3. \]
Similarly for $u_0 = 0.7/u_1 + 0.5/u_2$, $x_1$ is given by

$$x_1(0.7, 0.5) = 0.6/x_1 + 0.7/x_2 + 0.7/x_3,$$

then

$$T(u_0, R x_1) = (0.5 \ 0.9 \ 0.2) \circ (0.8 \ 0.8 \ 0.3)$$

$$\wedge (0.3 \ 0.8) \circ (0.2 \ 0.9)$$

$$= 0.8 \text{ for } u_0 = 0.2/u_1 + 0.9/u_2.$$

Similarly, $T(u_0, R x_1) = 0.5$ for $u_0 = 0.7/u_1 + 0.5/u_2$ giving $u_0 = 0.2/u_1 + 0.9/u_2$ as optimal.

### 7. Multi-Stage Problem

The one stage development discussed above can be extended to the multi-stage case.

Let $\bar{U} = U_0 \times A_1 \times \cdots \times U_{T-1} = U \times U \cdots \times U = U^T$ and also

$$R \subseteq \bar{U} \times X,$$

where $R(u_0, u_1, \ldots, u_{T-1}, x_T) = \chi_0(u_0) \wedge \chi_1(u_1) \wedge \cdots \wedge \chi_{T-1}(u_{T-1}) \wedge \chi_T(x_T)$. If a fuzzy control sequence $(u_0, u_1, \ldots, u_{T-1}) = u$, say, is used giving rise, via mapping $\delta$, to final state $x_T$, then from Lemma 2 below

$$T(u R x_T) = (\chi_0 \circ \chi_{u_0}) \wedge (\chi_1 \circ \chi_{u_1}) \wedge \cdots \wedge (\chi_{T-1} \circ \chi_{u_{T-1}}) \wedge (\chi'_T \circ x_T).$$

The optimal decision sequence is then defined as that sequence $u \in \bar{U}^*$ which maximizes $T(u R x_T)$, where $\bar{U}^*$ is the allowed set of fuzzy sequences.

**Lemma 2.**

$$T(u R x_T) = (\chi_0 \circ \chi_{u_0}) \wedge (\chi_1 \circ \chi_{u_1}) \wedge \cdots \wedge (\chi_{T-1} \circ \chi_{u_{T-1}}) \wedge (\chi'_T \circ x_T).$$

**Proof.** The decision "goals and constraints satisfied" for the case of non-fuzzy $u_i$ and $x_T$ can be expressed by the truth function

$$T(u_0, u_1, \ldots, u_{T-1}, R, x_T) = \chi_0(u_0) \wedge \chi_1(u_1) \wedge \cdots \wedge \chi_{T-1}(u_{T-1}) \wedge \chi'_T(x_T).$$
Also for fuzzy \( u_0 \) and non-fuzzy \( u_1, u_2, \ldots, u_{T-1} \) and \( x_T \), then

\[
T(u_0, u_1, \ldots, u_{T-1} R x_T)
= \bigvee_{u_0 \in U} \left[ T(u_0, u_1 \cdots u_{T-1} R x_T) \land \chi_{u_0}(u_0) \right]
= \left\{ \bigvee_{u_0 \in U} \left[ \chi_0(u_0) \land \chi_{u_0}(u_0) \right] \right\} \land \chi_1(u_1) \land \chi_2(u_2) \land \cdots \land \chi_{T-1}(u_{T-1}) \land \chi_T'(x_T)
\]

using distributivity property of \( \land \) over \( \lor \) as in Lemma 1. Similarly

\[
T(u_0, u_1, u_2, \ldots, u_{T-1} R x_T)
= \bigvee_{u_1 \in U} \left[ T(u_0, u_1, \ldots, u_{T-1} R x_T) \land \chi_{u_1}(u_1) \right]
= (\chi_0 \circ \chi_{u_0}) \land \left\{ \bigvee_{u_1 \in U} \left[ \chi_1(u_1) \land \chi_{u_1}(u_1) \right] \right\} \land \chi_2(u_2) \land \cdots \land \chi_{T-1}(u_{T-1}) \land \chi_T'(x_T)
\]

so that

\[
T(u_0, u_1, u_2, \ldots, u_{T-1} R x_T)
= (\chi_0 \circ \chi_{u_0}) \land (\chi_1 \circ \chi_{u_1}) \land (\chi_2(u_2) \land \cdots \land (\chi_{T-1}(u_{T-1}) \land \chi_T'(x_T)
\]

and thus by extension

\[
T(u_0, u_1, \ldots, u_{T-1} R x_T)
= (\chi_0 \circ \chi_{u_0}) \land (\chi_1 \circ \chi_{u_1}) \land (\cdots \land (\chi_{T-1}(u_{T-1}) \circ \chi_{x_T}).
Q.E.D.
\]

8. Dynamic Programming Solution of Multi-Stage Problem

We now consider the solution of the multi-stage fuzzy decision problem of the last section: Let

\[
S_k(x_k) = \max_{u_k, \ldots, u_{T-1} \in U_k} T(u_k, \ldots, u_{T-1} R x_T)
\]
when system starts at stage \( k \) in fuzzy state \( x_k \) and \( x_T \) is the final fuzzy state obtained via mapping \( \delta \) applied at each stage where \( U_k^T \) is the set of allowed fuzzy sequences of form \( (u_k, \ldots, u_{r-1}) \). Hence

\[
S_k(x_k) = \max_{u_k \in U_k} \left[ (x_k \circ x_{u_k}) \wedge (x_{k+1} \circ x_{u_{k+1}}) \wedge \cdots \wedge (x_{T-1} \circ x_{u_{T-1}}) \wedge (x_T \circ x_T) \right]
\]

so that

\[
S_k(x_k) = \max_{u_k \in U_k} \left[ (x_k \circ x_{u_k}) \wedge S_{k+1}(x_{k+1}) \right]
\]

for \( k = T, 1, T-2, \ldots, 1, 0 \), with

\[
S_T(x_T) = (x_T \circ x_T)
\]

as boundary condition, where \( x_{k+1} \) is the state resulting from using fuzzy control \( u_k \) when system is in state \( x_k \), i.e., \( x_{k+1} \) is a fuzzy subset on \( X \), namely,

\[
x_{k+1}(x_{k+1}) = \bigvee_{(x_k, u_k) \in X \times U} \{x_k(x_k) \wedge x_{u_k}(u_k) \wedge \delta(x_k, u_k; x_{k+1}) \}
\]

\[
= (x_k(x_k) \wedge x_{u_k}(u_k)) \circ \delta(x_k, u_k; x_{k+1}).
\]

It should be noted that for a deterministic mapping \( \delta \) replacing \( \delta \) and non-fuzzy \( x_0 \) then Bellman's equation of Section 2 is obtained from the functional equation of this section.

It can be seen that the dynamic programming functional equation for this problem with fuzzy mapping and initial state in addition to fuzzy environment has associated with it the well known "curse of dimensionality" of dynamic programming. We will describe in the next section a fuzzification scheme for the solution of this D.P. functional equation. Briefly this corresponds to finding a relational matrix for \( S_{k+1}(x_{k+1}) \) using such statements as

If \( x_{k+1} \) is large then \( S_{k+1}(x_{k+1}) \) is small,
reasoning involves such a process as this when dealing with large amounts of factual information.

The allowed set of fuzzy decisions $U^*$ can, of course, consist of only non-fuzzy decisions. We have allowed the inclusion of fuzzy decisions, since these are meaningful if a human operator is the decision maker in the sense of actually carrying our orders.

9. Fuzzy Interpolation for D.P. Functional Equation

Consider that the following statements are known for $S_{k+1}(x)$

If $x = x_i'$ then $S_{k+1}(x) = S_{k+1}(x_i')$ else.

If $x = x_i''$ then $S_{k+1}(x) = S_{k+1}(x_i'')$ else.

... ... ... ... ... ... ... ...

If $x = x_n'$ then $S_{k+1}(x) = S_{k+1}(x_n')$.

Note that it is assumed that the determination of $S_{k+1}(x)$ for a given $x$ is fuzzy and the value of $S$ is a fuzzy subset $\subseteq [0, 1]$. We can form a relational matrix $R(k + 1)$ representing these conditional fuzzy statements concerning the behaviour of $S$ as defined by Zadeh [3]:

$$R(k + 1) \subseteq X \times [0, 1].$$

For any given subset of $X$, namely, $x'$, we thus can obtain an induced $S \subseteq [0, 1]$ given by

$$S_{k+1}(x') = x' \circ R(k + 1)$$

and thus we will use in the functional equation when calculating $S_k(x)$. Thus, for a given $x_k$ and $u_k$ we calculate

$$b(x_k, u_k) = [(\chi_k \circ \chi_{u_k}) \wedge S_{k+1}(x_{k+1})] = [(\chi_k \circ \chi_{u_k}) \wedge (x_{k+1} \circ R(k + 1))].$$

This may be written as

$$b = [a \wedge s], \quad \text{where } a \in [0, 1] \text{ and } S \subseteq [0, 1] \text{ defined by } \chi_s.$$

This is interpreted as follows:

<table>
<thead>
<tr>
<th>$x_k(x)$</th>
<th>$u_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x &gt; a$</td>
<td>$0$</td>
</tr>
<tr>
<td>$x = a$</td>
<td>$\sup_{y \geq a} \chi_s(y)$</td>
</tr>
<tr>
<td>$x &lt; a$</td>
<td>$\chi_s(x)$</td>
</tr>
</tbody>
</table>
For example,

\[ 0.2 \land (0.1/0.9 + 0.2/0.7 + 0.3/0.6) = 0.1/0.9 + 0.2/0.7 \]

or

\[(ii) \quad \chi_b(x) = 0 \quad \text{if} \quad x > a \]
\[= \sum_{y \in a} \chi_b(y) \quad \text{if} \quad x = a \]
\[= \chi_b(x) \quad \text{if} \quad x < a, \]

where \[\sum_{y_1, y_2} (\chi_b(y_i)) = \chi_b(y_1) + \chi_b(y_2) = \chi_b(y_1) + \chi_b(y_2) - \chi_b(y_1) \cdot \chi_b(y_2).\]

For example,

\[ 0.2 \land (0.1/0.9 + 0.2/0.7 + 0.3/0.6) = 0.1/0.9 + (0.2/0.7 + 0.2/0.6) \]
\[= 0.1/0.9 + 0.2/0.68. \]

The set of fuzzy states \(x_1', x_2', \ldots, x_k'\) are termed the reference set of states. One can also define a set of fuzzy controls \(u_1', u_2', \ldots, u_k'\) for stage \(k\) and by a series of calculations as above using these reference controls one can produce the following conditional statements:

For a given \(x_k\)

If \(u_k = u_k'\), then \(b(x_k, u_k) = b_1(x_k, u_k')\) else.

If \(u_k = u_k'\), then \(b(x_k, u_k) = b_2(x_k, u_k')\) else.

If \(\ldots\) else.

If \(u_k = u_k'\), then \(b(x_k, u_k) = b_m(x_k, u_k')\).

A fuzzy relational matrix \(R'(x_k)\) can now be produced representing these fuzzy conditional statements concerning the behaviour of \(b(x_k, u_k)\). A method must now be found for determining the \(u_k \in U^*\) which gives the best \(b(x_k, u_k)\) for a given \(x_k\) where best is interpreted as optimal in a maximizing sense defined above. In this paper we use the following method:

Assume \(U^* = U\).

Let \(k'-\text{large}\) be a fuzzy subset of \([0, 1]\) such that

\[ \chi_{k'-\text{large}}(x) \left\{ \begin{array}{ll}
0 & \text{for} \quad x \leq k' \\
> 0 & \text{for} \quad x > k'
\end{array} \right. \]

with \(\chi_{k'-\text{large}}(y_1) > \chi_{k'-\text{large}}(y_2)\) for \(y_1 > y_2 > k'\) and \(\chi_{k'-\text{large}}(1) = 1. \)
These are test fuzzy subsets and should be chosen carefully such that \( x_{k_1} \)-\( \text{large} \) is preferred to \( x_{k_2} \)-\( \text{large} \) if \( k_1 > k_2 \). Form the induced fuzzy subsets
\[
u(k') = R'(x_k) \circ x_{k'-\text{large}} \quad \text{for} \quad k = 1, 0.9, 0.8, \ldots
\]
and accept \( u_k \) (optimal) = \( u(k') \) corresponding to the largest \( k' \) such that
\[
\operatorname{SUP}_x x_{u(k')} (x) \geq m
\]
where \( m \) is a parameter which should be carefully chosen. It should be noted that
\[
T(uR'(x(k))) x_{k'-\text{large}}
\]
is maximum when \( u = R'(x_k) \circ x_{k'-\text{large}} \). One can then write
\[
u_k(x_k) = R'(x_k) \circ x_{k'-\text{large, must satisfied}}
\]
and
\[
S_k(x_k) = u_k \circ R'(x_k).
\]
In the notation used above we have assumed the control reference set to depend on the stage \( k \) although for all examples in this paper the same reference set is taken for each stage. It should also be noted that the test fuzzy subsets influence the determination of \( R(k) \).

The above calculation can be done for different \( x_k \) and so we can determine a set of conditional fuzzy statements for a control policy at stage \( k \) as follows:
\[
\text{If } x_k = x_{k_1} \text{ then } u_k(x_k) = u_{k_1}(x_{k_1}) \text{ else.}
\]
\[
\ldots \quad \text{else.}
\]
\[
\text{If } x_k = x_{k_n} \text{ then } u_k(x_k) = u_{k_n}(x_{k_n}),
\]
and hence we can form a relation \( R_k(k) \) representing the fuzzy control policy for stage \( k \). In a similar manner \( R(k) \) is calculated using the conditional statements:
\[
\text{If } x_k = x_{k_1} \text{ then } S_k(x_k) = S_{k_1}(x_{k_1}) \text{ else.}
\]
\[
\ldots \quad \text{else.}
\]
\[
\text{If } x_k = x_{k_n} \text{ then } S_k(x_k) = S_{k_n}(x_{k_n}),
\]
obtained from above.

Once again we have notated the dependence of the state reference set on \( k \) but in this paper the same reference set is used for all stages. We now move to the preceding stage and repeat and continue until the stage calculations are performed in the usual Dynamic Programming Backward recurrence
manner. The method is initiated in the usual way using the boundary condition

Thus a sequence of control policies $R_u(T-1), R_u(T-2), \ldots, R_u(1), R_u(0)$ is formed.

The solution to a specified problem can now be determined using forward recursion by inducing a control using the stage control policy and the state. For example, if $x_0$ is the initial state, then

$$u_0 = x_0 \circ R_u(0).$$

This procedure is then followed stage by stage.

Feedback enters this model through the observation of the state from stage to stage and this observation can, of course, be fuzzy. The observed fuzzy state need not coincide with the calculated new state using the mapping $\delta$ since $\delta$ may only represent the dynamics in an approximate way and will not necessarily simulate the process of observation.

10. Demonstration Example

Consider, as a simple example, an automata having three element state set $X$ and two element control set $U$, thus:

$$X = \{x_1, x_2, x_3\}, \quad U = \{u_1, u_2\},$$

and suppose the system dynamics be governed by a fuzzy mapping $\delta: X \times U \rightarrow X$ as follows:

$$\delta = (X \times U) \begin{bmatrix} x_1 u_1 & 1 \alpha \beta \\ x_1 u_2 & 1 \alpha \beta \\ x_2 u_1 & \alpha 1 \alpha \\ x_2 u_2 & 1 \alpha \beta \\ x_3 u_1 & \beta \alpha 1 \\ x_3 u_2 & 1 \alpha \beta \end{bmatrix} ; \quad \alpha > \beta; \alpha, \beta \in [0, 1],$$

where $\alpha, \beta$ are system parameters which express the degree of fuzziness of the mapping $\delta$.

Consider the $T=3$ stage problem and suppose that the goal, $x_3$, and the constraints $x_i, i = 0, 1, 2$, are defined as follows:
\[
\begin{align*}
\chi_3 &= (1 \ 0.4 \ 0.1); \\
\chi_2 &= (1 \ 1), \quad \chi_1 = (0.8 \ 0.7); \quad \chi_0 = (1 \ 0.5),
\end{align*}
\]
where
\[
\chi_3 = (1 \ 0.4 \ 0.1) \rightarrow \chi_3 = (x_1/1, x_2/0.4, x_3/0.1)
\]
and
\[
\chi_1 = (0.8 \ 0.7) \rightarrow \chi_1 = (u_1/0.8, u_2/0.7), \text{ etc.}
\]

11. Sample of Detailed Calculations

Based on the simple system description of Section 10 above, the following sample of detailed calculations show the kind of results which occur when fuzzy interpolation is employed. For simplicity, only three reference fuzzy states, \( x_i, i = 1, ..., 3 \), and two reference fuzzy controls, \( u_j, j = 1, 2 \), are defined as follows:

\[
\begin{align*}
\chi_{x_1} &= (1 \ 0.4 \ 0.1), \quad \chi_{u_1} = (1 \ 0.2), \\
\chi_{x_2} &= (0.4 \ 1 \ 0.4), \quad \chi_{u_2} = (0.2 \ 1), \\
\chi_{x_3} &= (0.1 \ 0.4 \ 1),
\end{align*}
\]

and the parameters for the mapping \( \delta \) are defined as
\[
\alpha = 0.7, \quad \beta = 0.3.
\]

At stage \( k \), the above reference fuzzy states and controls induce fuzzy states at stage \( k + 1 \) as follows:

\[
\begin{align*}
\chi_{x_1(k+1)}(x_1, u_1) &= (1 \ 0.7 \ 0.4), \\
\chi_{x_1(k+1)}(x_1, u_2) &= (1 \ 0.7 \ 0.3), \\
\chi_{x_2(k+1)}(x_2, u_1) &= (0.7 \ 1 \ 0.7), \\
\chi_{x_2(k+1)}(x_2, u_2) &= (1 \ 0.7 \ 0.3), \\
\chi_{x_3(k+1)}(x_3, u_1) &= (0.4 \ 0.7 \ 1), \\
\chi_{x_3(k+1)}(x_3, u_2) &= (1 \ 0.7 \ 0.3).
\end{align*}
\]

The boundary condition on the dynamic programming recurrence equation gives the following end-stage decisions:

\[
S_1(x_1) = 1; \quad S_3(x_2) = 0.4; \quad S_3(x_3) = 0.4.
\]
The conditional statements: "If \( x_1 \) then \( S_3(x_2) \) else..." give rise to a relation \( R(3) \subset X \times \{0, 1\} \) as follows:

\[
R(3) = (X) \begin{bmatrix}
0.4 & 1 \\
0.4 & 1 \\
1 & 0.4 \\
1 & 0.1
\end{bmatrix}.
\]

Hence, for example, at stage 2:
- \( x_2 \) and \( u_1 \) induce \( S_3(x_2)(x_1, u_1) = \{0.4/1, 1/0.7\} \),
- \( x_2 \) and \( u_2 \) induce \( S_3(x_2)(x_1, u_2) = \{0.4/0.7, 1/1\} \).

Then, since control-constraint compositions are both unity (i.e., \( x_2 \circ x_{u_1} = x_2 \circ x_{u_2} = 1 \)):

\[
S_2(x_2) = \text{MAX} \left\{ u_1 : [1 \wedge \{0.4/1, 1/0.7\}], u_2 : [1 \wedge \{0.4/0.7, 1/1\}] \right\} = \text{MAX} \left\{ u_1 : \{0.4/1, 1/0.7\}, u_2 : \{0.4/0.7, 1/1\} \right\}.
\]

The above braces imply conditional statements: "If \( u_1 \) then \( S_2(u_1) \) else if \( u_2 \) then \( S_2(u_2) \)," which give rise to relation \( R_2'(x_2) \subset U \times \{0, 1\} \) as follows:

\[
R_2'(x_2) = (U) \begin{bmatrix}
0.4 & 1 \\
0.7 & 1
\end{bmatrix}.
\]

Test function \( x_{k_{\text{large}}} = \{1/1\} \) obtains a control with maximum truth for \( M = 1 \). Thus:

\[
\chi_{u}(2)(x_2) \text{ optimal } = (0.7 \quad 1)
\]
and

\[
S_2(x_2) = \{0.4/0.7, 1/1\}.
\]

Similar calculations are made for all three reference fuzzy states at stages 2, 1 then 0. For example, the conditional statements: "If \( x_1 \) then \( S_4(x_1) \) else..." give rise to a relation \( R(1) \subset X \times \{0, 1\} \) as follows:
so that at stage 0:

\[ x_1 \text{ and } u_1 \text{ induce } S_1(x(1)(x_1, u_1)) = \{0.4/0.7, 0.7/0.2, 0.8/1\}, \]

\[ x_1 \text{ and } u_2 \text{ induce } S_1(x(1)(x_1, u_2)) = \{0.4/0.7, 0.7/0.2, 0.8/1\}. \]

Then, since \( x_0 \circ x_1 = 1 \) and \( x_0 \circ x_2 = 0.5 \):

\[
S_0(x_1) = \text{MAX} \left\{ u_1 : \left[ 1 \land (0.4/0.7, 0.7/0.2, 0.8/1) \right], u_2 : \left[ 0.5 \land (0.4/0.7, 0.7/0.2, 0.8/1) \right] \right\},
\]

\[
= \text{MAX} \left\{ u_1 : (0.4/0.7, 0.7/0.2, 0.8/1), u_2 : (0.4/0.7, 0.5/1) \right\}.
\]

The braces imply conditional statements: "If \( u_1 \) then \( S_0(u_1) \) else if \( u_2 \) then \( S_0(u_2) \)," which give rise to a relation \( R_0(x_1) \subseteq V \times [0, 1] \) thus:

\[
([0, 1]) \\
0.4 \quad 0.5 \quad 0.7 \quad 0.8
\]

\[
R_0(x_1) = (U) u_1 \left[ \begin{array}{ccc}
0.7 & 0.2 & 0.2 \\
0.7 & 1 & 0.2 & 0.2
\end{array} \right].
\]

The test-function \( x_0 \) optimal = \( (1 \quad 0.2) \)

and

\[
S_0(x_1) = \{0.4/0.7, 0.5/0.2, 0.7/0.2, 0.8/1\}.
\]

The backward induction solution process gives conditional statements: "If \( x_1 \) then \( u(x_1) \) optimal else..." for each stage and these give rise to fuzzy relations \( R_u(k) \subseteq X \times U, k = 0, 1, 2 \) as follows:

\[
R_u(0) = x_2 \left[ \begin{array}{c}
1 \\
1 \\
1
\end{array} \right], \quad R_u(1) = \left[ \begin{array}{c}
1 \\
1 \\
1
\end{array} \right], \quad R_u(2) = \left[ \begin{array}{c}
1 \\
0.7 \\
0.4
\end{array} \right].
\]
The optimal fuzzy control policy may then be derived for any initial state $x_0$. For example, the solution with $x_{k_0} = (0 \ 0.2 \ 1)$ is:

<table>
<thead>
<tr>
<th>STAGE</th>
<th>STATE (induced by control)</th>
<th>OPTIMAL CONTROL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$(0 \ 0.2 \ 1)$</td>
<td>$(1 \ 0.2)$</td>
</tr>
<tr>
<td>1</td>
<td>$(0.3 \ 0.7 \ 1)$</td>
<td>$(1 \ 0.2)$</td>
</tr>
<tr>
<td>2</td>
<td>$(0.7 \ 0.7 \ 1)$</td>
<td>$(0.7 \ 1)$</td>
</tr>
<tr>
<td>3</td>
<td>$(1 \ 0.7 \ 0.7)$</td>
<td></td>
</tr>
</tbody>
</table>

12. Effects of Changing System Parameters

Using the simple system description of Section 10, the following results can be derived, which show how the solution table fuzzy relations $R_u(k): X \to U$, $k = 0, 1, 2$, are affected by changes in:

(i) The degree of fuzziness of the mapping $\delta$, via parameters $\alpha$ and $\beta$.
(ii) The degree of fuzziness of the reference state sets and reference control sets.

In addition, the truth relation $R(0): X \to [0, 1]$ is formed at the initial stage which gives a measure of how well satisfied for truth is any optimal policy.

(iii) The effect of changing the reference state fuzzy sets on this truth relation is then demonstrated.

(i) Four parameter pairs $(\alpha, \beta)$ are considered, thus:

(a) $(\alpha, \beta) = (0, 0)$,
(b) $(\alpha, \beta) = (0.3, 0.1)$,
(c) $(\alpha, \beta) = (0.5, 0.2)$,
(d) $(\alpha, \beta) = (0.7, 0.3)$.

Using deterministic reference sets, i.e.:

$X_{x_1} = (1 \ 0 \ 0)$; $X_{x_1} = (1 \ 0)$;
$X_{x_2} = (0 \ 1 \ 0)$; $X_{x_2} = (0 \ 1)$;
$X_{x_3} = (0 \ 0 \ 1)$;

the relations $R_u$ are as follows:

$R_u(0) = R_u(1) = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$; $R_u(2) = \begin{bmatrix} 1 & 1 \\ \alpha & 1 \\ \beta & 1 \end{bmatrix}$. 
Using the following 10 reference state fuzzy sets and 7 reference control fuzzy sets:

\begin{align*}
\chi_{x_1} &= (1 \quad 0.3 \quad 0); \quad \chi_{u_1} = (1 \quad 0.1); \\
\chi_{x_2} &= (0.9 \quad 0.5 \quad 0.1); \quad \chi_{u_2} = (0.9 \quad 0.3); \\
\chi_{x_3} &= (0.8 \quad 0.6 \quad 0.3); \quad \chi_{u_3} = (0.8 \quad 0.5); \\
\chi_{x_4} &= (0.7 \quad 0.7 \quad 0.4); \quad \chi_{u_4} = (0.7 \quad 0.7); \\
\chi_{x_5} &= (0.6 \quad 0.9 \quad 0.5); \quad \chi_{u_5} = (0.5 \quad 0.8); \\
\chi_{x_6} &= (0.5 \quad 0.9 \quad 0.6); \quad \chi_{u_6} = (0.3 \quad 0.9); \\
\chi_{x_7} &= (0.4 \quad 0.7 \quad 0.7); \quad \chi_{u_7} = (0.1 \quad 1); \\
\chi_{x_8} &= (0.3 \quad 0.6 \quad 0.8); \\
\chi_{x_9} &= (0.1 \quad 0.5 \quad 0.9); \\
\chi_{x_{10}} &= (0 \quad 0.3 \quad 1),
\end{align*}

the relations \( R_u \) are as follows:

\[
R_u(0) = R_u(1) = \begin{bmatrix} 1 & 0.5 \\ 0.9 & 0.5 \\ 1 & 0.5 \end{bmatrix},
\]

but for parameter pairs (a), (b) and (c):

\[
R_u(2) = \begin{bmatrix} 1 & 1 \\ 0.7 & 0.9 \\ 0.5 & 1 \end{bmatrix}
\]

and for parameter pair (d):

\[
R_u(2) = \begin{bmatrix} 1 & 1 \\ 0.7 & 0.9 \\ 0.7 & 1 \end{bmatrix}
\]

The results show the tendency for fuzzy reference sets to make the solution less sensitive to changes in the degree of fuzziness of the mapping \( \delta \).

(ii) The degree of fuzziness of the reference sets may conveniently be expressed parametrically, as follows:

\[
\begin{align*}
\chi_{x_1} &= (1 \quad \lambda \quad \mu); \quad \chi_{u_1} = (1 \quad \nu); \\
\chi_{x_2} &= (\lambda \quad 1 \quad \lambda); \quad \chi_{x_2} = (\nu \quad 1); \\
\chi_{x_3} &= (\mu \quad \lambda \quad 1);
\end{align*}
\]

where \( \lambda, \mu, \nu \in [0, 1] \), and \( \lambda \geq \mu \).
Taking \((\alpha, \beta) = (0.7, 0.3)\), three parameter triples \((\lambda, \mu, \nu)\) are considered, thus:

(a) \((\lambda, \mu, \nu) = (0, 0, 0)\),
(b) \((\lambda, \mu, \nu) = (0.4, 0.1, 0.2)\),
(c) \((\lambda, \mu, \nu) = (0.6, 0.3, 0.4)\),

and the following relations \(R_u\) are obtained:

\[
R_u(0) = R_u(1) = \begin{bmatrix} 1 & \nu \\ 1 & \nu \\ 1 & \nu \end{bmatrix},
\]

for parameter triple (a):

\[
R_u(2) = \begin{bmatrix} 1 & 1 \\ \alpha & 1 \\ \beta & 1 \end{bmatrix},
\]

and for parameter triples (b) and (c):

\[
R_u(2) = \begin{bmatrix} 1 & 1 \\ \alpha & 1 \\ \lambda & 1 \end{bmatrix}.
\]

(iii) The truth relation \(R(0): X \rightarrow [0, 1]\) is formed from the fuzzy conditional statements: "If \(x_i\) then \(S_i(x_i)\) else ... else if \(x_3\) then \(S_3(x_3)\)."

Continuing the parametric description of (ii) above, with \((\alpha, \beta) = (0.7, 0.3)\) as above, but taking \(\nu = 0.2\), it is found, for each set of reference states, that all the rows of \(R(0)\) are the same and the following results are obtained:

(a) \((\lambda, \mu) = (0, 0) \rightarrow R^{row}(0) = \{0.1/0.7, 0.4/0.7, 0.5/0.2, 0.7/0.2, 0.8/1\} ,
(b) \((\lambda, \mu) = (0.3, 0.1) \rightarrow R^{row}(0) = \{0.3/0.7, 0.4/0.7, 0.5/0.2, 0.7/0.2, 0.8/1\} ,
(c) \((\lambda, \mu) = (0.4, 0.1) \rightarrow R^{row}(0) = \{0.4/0.7, 0.5/0.2, 0.7/0.2, 0.8/1\} ,
(d) \((\lambda, \mu) = (0.5, 0.1) \rightarrow R^{row}(0) = \{0.4/0.7, 0.5/0.7, 0.7/0.2, 0.8/1\} ,
(e) \((\lambda, \mu) = (0.7, 0.3) \rightarrow R^{row}(0) = \{0.4/0.7, 0.5/0.2, 0.7/0.7, 0.8/1\}.

Each of the \(R^{row}(0)\) is in fact the fuzzy truth of the optimal policy for any normal initial fuzzy state \(x_0\) (where normal means that peak membership is unity). The fuzzy truths suggest descriptions of the form: "Approximately \(S_1\) OR approximately \(S_2\)." For example, in (a), \(R^{row}(0)\) may be interpreted as: "approximately between 0.1 and 0.4 OR approximately 0.8," and in (c).
R_{0}^{row}(0) may be interpreted as: "approximately 0.4 OR approximately 0.8." Clearly (c), (d) and (e) are preferable to (a) and (b) because of their higher truth values overall. But which of (c), (d) and (e) is to be preferred is not so obvious.

It is found that a very poor decision is obtained (i.e., one with a very low truth) if the end-stage truth relation R(3), defined by the boundary condition, is too fuzzy. For example, by appropriate choice of reference states at the end-stage only, the following relation R(3) can be obtained:

\[
x_1 \begin{bmatrix} 0 & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 & 1 \\
0 & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 & 1 \\
0 & 0.1 & 0.2 & 0.3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 
\end{bmatrix}
\]

If, thereafter, the same parameters as defined in (c) are computed, the following truth relation R(0) is obtained:

\[
x_1 \begin{bmatrix} 0 & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 \\
0 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 1 \\
0 & 0.7 & 0.7 & 0.7 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 
\end{bmatrix}
\]

which has an intolerably high membership level for low truth values. This result is to be expected due to the effect of propagation of fuzziness which is a feature of any multi-stage decision with fuzzy dynamics.

13. DISCUSSION OF REFERENCE CONTROL SET

It is not immediately obvious why it should be necessary or desirable to choose a set of fuzzy reference controls rather than deterministic ones; but the following exact solution of a single stage example shows how a fuzzy control can derive a higher truth value than any deterministic control.

Consider a three element state set X and a three element control set U, thus:

\[ X = \{x_1, x_2, x_3\}, \quad U = \{u_1, u_2, u_3\}; \]

and let the system dynamics be governed by a fuzzy mapping \( \delta: X \times U \to X \) as follows:
Consider only the single stage problem and suppose that the goal, $\chi'_1$, and the constraint, $\chi_0$, are defined as follows:

$$\chi'_1 = (0.1 \ 0.5 \ 0.9); \quad \chi_0 = (0.4 \ 1 \ 0.6).$$

Let the initial state $x_0$ be given by $x_{x_0} = (1 \ 0.5 \ 0.1)$ and let the truth value associated with fuzzy control $u^*$, where $x_{u^*} = (1 \ 0.8 \ 0.3)$, be compared with the three deterministic controls.

From Section 5, the expression for truth is:

$$T(uR\chi_i) = (\chi'_i \circ \chi_{x_i}) \land (\chi_0 \circ \chi_u),$$

where $x_i = \delta(x_n, u)$:

$$\chi_u = \{u_1/1\} = (1 \ 0 \ 0) \Rightarrow \chi_{x_i} = (0.2 \ 0.7 \ 1)$$
$$\Rightarrow T = (0.1 \ 0.5 \ 0.9) \circ (0.2 \ 0.9 \ 1)$$
$$\land (0.4 \ 0.6 \ 0) \circ (1 \ 0 \ 0),$$

$$\chi_u = \{u_2/1\} = (0 \ 1 \ 0) \Rightarrow \chi_{x_i} = (0.4 \ 0.8 \ 0.7)$$
$$\Rightarrow T = (0.1 \ 0.5 \ 0.9) \circ (0.4 \ 0.8 \ 0.7)$$
$$\land (0.4 \ 1 \ 0.6) \circ (0 \ 1 \ 0),$$

$$\chi_u = \{u_3/1\} = (0 \ 0 \ 1) \Rightarrow \chi_{x_i} = (0.9 \ 0.1 \ 0)$$
$$\Rightarrow T = (0.1 \ 0.5 \ 0.9) \circ (0.9 \ 0.1 \ 0)$$
$$\land (0.4 \ 1 \ 0.6) \circ (0 \ 0 \ 1),$$

$$\chi_u = x_{u^*} = (1 \ 0.8 \ 0.3) \Rightarrow \chi_{x_i} = (0.4 \ 0.7 \ 1)$$
$$\Rightarrow T = (0.1 \ 0.5 \ 0.9) \circ (0.4 \ 0.7 \ 1)$$
$$\land (0.4 \ 1 \ 0.6) \circ (1 \ 0.8 \ 0.3).$$
Hence,

If \( x_u = (1\ 0\ 0) \) then \( T = 0.9 \land 0.4 = 0.4, \)
If \( x_u = (0\ 1\ 0) \) then \( T = 0.7 \land 1 = 0.7, \)
If \( x_u = (0\ 0\ 1) \) then \( T = 0.5 \land 0.6 = 0.5, \)
If \( x_u = (1\ 0.8\ 0.3) \) then \( T = 0.9 \land 0.8 = 0.8, \)

so that \( u^* \) has a higher truth associated with it than any deterministic control.

14. CONCLUSIONS

A dynamic programming functional equation is formulated for a multi-stage decision process in which both the environment and the system dynamics are fuzzy. The functional equation is a generalization of that derived by Bellman and Zadeh [2] to which it simplifies when the system dynamics are non-fuzzy.

The problem of high dimensional state, inherent in the formulation, is treated by employing a new concept of "fuzzy interpolation," which uses reference state and reference control fuzzy sets to explore the solution over the desired range.

The reference fuzzy sets need to be chosen carefully to suit each individual problem and it is found, for example, that deterministic reference sets are not necessarily the best. However, it is demonstrated that the essential character of the solution may not be affected much by the choice of reference sets, and only the degree of fuzziness of the control policy may be altered.

It is felt, that this approach to multi-stage decision problems perhaps models human decision behaviour better than traditional decision theory methods.

REFERENCES