Fuzzy Lookup Table Controller (FLTC) via Center Average De-fuzzifier (CAD) with a New Efficient Heaviside Search Algorithm (HSA) for Lookup Table Implementation

Chi-Hsu Wang¹, Kar-Chun Hor² and Wei-Chuan Chen²
¹ Department of Electrical and Computer Engineering, National Chiao Tung University, 1001 University Road, Hsinchu 300, Taiwan, ROC.
² College of Electrical and Computer Engineering, National Chiao Tung University, 1001 University Road, Hsinchu 300, Taiwan, ROC.

E-mail: cwang@cn.nctu.edu.tw; s358389@gmail.com

Abstract. This paper proposes an alternate interpretation of the traditional lookup table controller via the Center Average De-fuzzier (CAD) in Fuzzy Set Theory with a set of Disjoint Block Pulse Membership Functions (DBPMFs). Further, a new Heaviside Search Algorithm (HSA) is also proposed to implement the Fuzzy Lookup Table Controller (FLTC) in a more efficient way. The computation complexity of our new HSA is $O(\sum R_i)$, where $R_i$ is the number of membership functions for $i^{th}$ input variable in FLTC. Whereas the average computational complexity in traditional Linear Search Algorithm (LSA) is $O((\prod R_i)/2)$. In comparison with LSA, the HSA also reduces the complexity of coding significantly in the same order as the comparison of computation complexities between HSA and LSA. The balancing of Inverted Pendulum System (IPS) is adopted as the benchmark to show the feasibility and efficiency of this new FLTC and HSA.

1. Introduction

It is well-known that Fuzzy Logical Control (FLC), with its broad feasibility, has been widely applied to lots of industrial and system control applications [1-4]. As an important real-time controller in the past few decades, the traditional Lookup Table Techniques (LUTs) [5-7] have been applied effectively in designing real-time controller with the avoidance of massive computation burden. The de-fuzzification process in FLC can also be time-consuming to inhibit its real-time applications.

In this paper, by using the Center Average De-fuzzifier (CAD) with the same Disjoint Block-Pulse Membership Functions (DBPMFs) in [7], we can show that the FLC with DBPMFs is also equivalent to the traditional LUT, so that a Fuzzy Lookup Table Controller (FLTC) can be generated to simplify the de-fuzzification process in FLC. Furthermore, a new efficient Heaviside Search Algorithm (HSA) is also proposed to the implementation of FLTC. It is easy to show that the average computational complexity of applying the traditional Linear Search Algorithm (LSA) to FLTC is $O(\prod R_i / 2)$, where $R_i$ is the number of membership functions for $i^{th}$ input variable. However, the computational complexity of HSA for the search of FLTC is only in the order of $O(\sum R_i)$. In order to show the feasibility of FLC with DBPMFs to the control of nonlinear systems, the balancing of Inverted Pendulum System (IPS) [8-10] is illustrated via the FLTC with HSA.
2. The Fuzzy Lookup Table Controller (FLTC)

A general Fuzzy Logic Controller (FLC) consists of four modules, i.e., fuzzification module, inference engine module, fuzzy rules base module and defuzzification module. In this paper, \( n \) is the total number of fuzzy input variables for FLC, \( m \) is the total number of output variables for FLC, and \( L \) is the total fuzzy rules number in the fuzzy rule base. The total fuzzy rules number can be found via the following equation

\[
L = \prod_{i=1}^{n} R_i
\]

where \( R_i \) is the number of membership functions for \( i^{th} \) input variable. The fuzzy rules in the fuzzy rule base can be inferred as

Rule \( j \): IF \( x_i \) is \( A_i' \) and ... and \( x_n \) is \( A_n' \)

THEN \( u_k \) is \( W_k' \) and ... and \( u_m \) is \( W_m' \)

where \( j \) is the fuzzy rule number \((j = 1, 2, \ldots, L)\), \( A_i' \) are the input membership functions for \( x_i \) in the \( j^{th} \) fuzzy rule, and \( W_k' \) are the output membership functions for \( u_k \) \((k = 1, 2, \ldots, m)\) in the \( j^{th} \) fuzzy rule.

We adopt product inference engine and the center average de-fuzzifier (CAD) as the de-fuzzification method, as shown in the following equation:

\[
u_k = \frac{\sum_{i=1}^{n} \gamma_i^k (\prod_{i=1}^{n} A_i'(x_i))}{\sum_{j=1}^{L} (\prod_{i=1}^{n} A_i'(x_i))}, \quad (k = 1, \ldots, m)\]

(1)

where \( \gamma_i^k \) is the center value of the \( k^{th} \) output MF in the \( i^{th} \) fuzzy rule, \( S_j \) is the product of the MFs’ mappings \((A_i'(x_i), i = 1, 2, \ldots, n)\) for all input variables \((x_1, \ldots, x_n)\) in the \( j^{th} \) fuzzy rule, and \( u_k \) is the \( k^{th} \) crisp output of the FLC \((k = 1, 2, \ldots, m)\). It is noted that the conventional FLC is, in general, an online floating point computation process controller, which is time consuming for real time applications. Therefore, the Lookup Table (LUT) method is a useful tool to overcome this obstacle [7]. Hence, a special Fuzzy Lookup Table controller (FLTC) with DBPMFs will be developed and discussed in this section, in order to minimize the online computing resources of a controller. The DBPMF for the \( i^{th} \) MF for the \( x_i \) input of FLC can be defined as:

\[
A_i'(x) = \begin{cases} 
1, & \frac{\sigma_i'^2}{2} + c_i'^2 < x < \frac{\sigma_i'^2}{2} + c_i'^2 \\
0, & \text{otherwise}
\end{cases}
\]

(2)

where \( \sigma_i' \) and \( c_i' \) are the \( i^{th} \) variance (width) and mean (center) of the DBPMF in \( x_i \). The disjoint properties of DBPMFs will be used to construct the FLTC, which is shown as follows:

\[
A_i'(x)A'_j(x) = \begin{cases} 
A_i'(x), & l = k \\
0, & l \neq k
\end{cases}
\]

(3)

where \( l \) and \( k = 1, 2, \ldots, R_i \). The output of the center average de-fuzzifier in (3) can be simplified to a certain center value of an output MFs via the following Theorem 1.

**Theorem 1:** For the FLC with DBPMFs as the MFs, the FLC will be simplified to a Fuzzy Lookup Table Controller (FLTC), where every entry \( \{u_k | k = 1, \ldots, m\} \) in FLTC is a center value of a certain output membership function (MF). The \( u_k \) corresponds to a certain activated fuzzy rule number \( i \) is equal to

\[
u_k = \gamma_i^k
\]

(4)

where \( i \) is a certain activated fuzzy rule number and is decided by the inference process and \( \gamma_i^k \) is a center value of a certain output MFs corresponded to a certain fuzzy rule.

**Proof:** Assume that the FLC has \( n \) input variables, \( k \) outputs, each input variable has \( R_i \) DBPMFs and
has total \( L \) fuzzy rules, therefore, we can rewrite (3) as:

\[
u_k = \frac{\sum_{j=1}^{L} \lambda_j \mathbf{S}_j}{\sum_{j=1}^{L} \lambda_j}
\]

(8)

Since all the DBPMFs are disjoint with each other, there are only single \( i^{th} \) rule activated at each time, and due to the fact that the mapping value for every DBPMFs is equal to 1, therefore we have

\[
\begin{align*}
S_i &= 1, \quad l = i \\
S_i &= 0, \quad l \neq i
\end{align*}
\]

so we have

\[
\sum_{j=1}^{L} S_j = S_i = 1
\]

(10)

Thus, we can rewrite (8) as

\[
u_k = \frac{\sum_{j=1}^{L} \lambda_j \mathbf{S}_j}{\sum_{j=1}^{L} \lambda_j}
\]

(11)

Hence, we have the following result

\[
u_k = \mathbf{S}_i
\]

(12)

where \( i \) is the index of the activated fuzzy rule.

Q.E.D

In order to show the feasibility of FLTC (Theorem 1) for real control purpose, the popular nonlinear IPS will be balanced by the FLTC in the next section.

3. Balancing the IPS via FLTC

This section shows the feasibility of FLTC for balancing the IPS. The mass of the cart is \( m_c \), the pivot point mass at the upper end of the pendulum pole is \( m \), \( l \) is half the length of the pendulum pole, \( g \) is the gravity acceleration, \( \theta \) is the angle of the pole from the origin point and \( u \) is the control force applied to the cart. We define the first state variable \( x_1 \) equals to \( \theta \) (\( x_1 = \theta \)) and the second state variable \( x_2 \) equals to the angle velocity of the pole \( \dot{\theta} \). The dynamic equation of IPS can be expressed as:

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= g \sin x_1 - \frac{m_l x_2^2 \cos x_1 \sin x_1}{m + m_l} + \frac{\cos x_1}{l} \frac{m}{m + m_l} u
\end{align*}
\]

(13)

where the parameters are \( g = 9.8 \text{ m/s}^2 \), \( m_p = 1.0 \text{ kg} \), \( m = 0.1 \text{ kg} \) and \( l = 0.5 \text{ m} \). The control objective is to balance the pole around the origin position by controlling the cart moving forward \( (u > 0) \) or backward \( (u < 0) \) via FLTC. The following Example 1 shows the balancing of the IPS via a 5 by 5 FLTC.

Example 1: The 5 by 5 FLTC to balance the Inverted Pendulum System.

In this example, the 5 by 5 FLTC consists of two inputs \( (x_1 = e \) and \( x_2 = \Delta e) \), and each has 5 DBPMFs \( (R_1 = R_2 = 5) \). The control intervals for \( x_1 \) and \( x_2 \) are partitioned into five subintervals or five DBPMFs by using \( \eta_{i1} = -0.1047, \eta_{i2} = -0.106, \eta_{i3} = 0.1047 \) and \( \eta_{i4} = 0.106 \) for \( x_1 \), \( \eta_{i5} = -0.04 \), \( \eta_{i6} = -10^{-5} \), \( \eta_{i7} = 10^{-5} \) and \( \eta_{i8} = 0.04 \) for \( x_2 \), which are shown in Figure 1a and 1b.

![Figure 1](image)

**Figure 1.** (a) DBPMFs for input variable \( x_1 = e \). (b) DBPMFs for input variable \( x_2 = \Delta e \).

These subintervals also serve as the DBPMFs for the input \( x_1 \) and \( x_2 \). The boundary values of every subinterval are designed to match the system behaviour of the IPS. Moreover, 7 Gaussian MFs are
adopted for output MFs, where the center of each output Gaussian MFs is defined as \{NB = -5, NM = -3, NS = -1, ZE = 0, PS = 1, PM = 3, PB = 5\}. Therefore, we can further construct a 5 by 5 FLTC (by Theorem 1) with 7 output MFs, which is shown in Figure 2a.

| $\Delta c$ | NB | N | ZE | P | PB |
|-----------|----|---|----|---|----|
| NB        | NB (0) | NM (0) | NM (0) | NM (0) | NM (0) |
| N         | NB (0) | NM (0) | NM (0) | NM (0) | NM (0) |
| ZE        | NM (0) | NS (0) | NS (0) | PS (0) | PM (0) |
| P         | PS (0) | PS (0) | PM (0) | PB (0) | PB (0) |
| PB        | PM (0) | PM (0) | PM (0) | PB (0) | PB (0) |

(a)

Figure 2. (a) The FLTC for 5 DBPMFs. (b) The control response of IPS using conventional FLC (black dotted line) and the 5 by 5 FLTC (blue solid line).

We compare the performance of balancing the IPS via 5 by 5 FLTC and the conventional FLC found in [8], where the result is shown in Figure 2b. The initial angle of the IPS is 8 degrees, and the performance of the 5 by 5 FLTC (blue solid line) is better than conventional FLC (black dotted line), where 5 by 5 FLTC takes only 1 second to balance the IPS around the origin and the conventional FLC takes 7 seconds to fully balance the IPS.

4. The Heaviside Search Algorithm (HSA) for FLTC

This section proposes an efficient search algorithm based on the Heaviside function for FLTC to search the corresponding output value. The Heaviside function is expressed as follows:

$$H(\eta) = \begin{cases} 
0, & \eta < 0 \\
1, & \eta \geq 0
\end{cases}$$ (14)

The Heaviside index for FLTC is expressed as:

$$\Phi(x_i(t)) = 1 + \sum_{j=2}^{N} H(x_i(t) - \eta_j'); \quad (i = 1, \ldots, n)$$ (15)

where $H(\cdot)$ represents the Heaviside function in (14), $x_i(t)$ is the input signal from input $x_i$ at a certain time $t$, $j$ is the index of DBPMFs and $\eta_j'$ is the boundary value for $x_i$ in its $j^{th}$ DBPMF $A_j'$. Note that $\eta_j' < \eta_{j+1}'$. The Heaviside index $\Phi(x_i(t))$ in (15) returns the accumulative index of the corresponding DBPMF $A_j'$. This Heaviside search algorithm can then be performed to find the Heaviside indices for all input values \{ $x_i(t)$, $x_j(t)$, $x_k(t)$ \}, and the corresponding control output can be obtained as:

$$u_k = FLTC[\Phi(x_1(t)), x_2(t), \ldots, x_n(t)]$$ (16)

where $FLTC$ is the $n$ dimension fuzzy lookup table controller and $u_k$ is the control output in the $k^{th}$ ($k = 1, \ldots, m$) output. The following Algorithm 1 illustrates the Heaviside search algorithm (HSA) for FLTC.
Algorithm 1: Heaviside search algorithm (HSA) for FLTC

Input:

- $R_i$: The number of DBPMFs in each input variable $x_i$.
- $x_i(t)$: Input value from input variable $x_i$, $i=1, ..., n$.

Output:

- $\Phi(x_i(t))$: The Heaviside indices for $x_i(t)$.

Step 1: Define the DBPMFs intervals for all inputs.

For $i = 1$ to $n$
  For $j = 2$ to $R_i$
    Set $\eta'_j$
  End
End

Step 2: Find the all the Heaviside indices $\Phi(x_i(t))$ for every input variables.

For $i = 1$ to $n$
  For $j = 2$ to $R_i$
    If $(x_i(t) - \eta'_j) \geq 0$
      $\Phi(x_i(t)) = \Phi(x_i(t)) + 1$
    End
  End
End

Step 3: The output is

$$u_k = FLTC[\Phi(x_1(t)), \Phi(x_2(t)), ..., \Phi(x_n(t))]$$

Step 4: Stop.

5. The comparison of computational and coding complexities between HSA and LSA

This section compares the computational complexities between the HSA and traditional LSA. Assume a FLC has $n$ inputs and $m$ outputs, each input has $R_i$ DBPMFs. Therefore, a FLTC with the dimension equals to $\prod_{i=1}^{n} R_i$ can be constructed. The LSA is roughly shown as follows:

$$
\begin{align*}
\text{if rule 1 then } u_{i,m} &= \pi_{i,m}^1 \\
\text{else if rule 2 then } u_{i,m} &= \pi_{i,m}^2 \\
&\vdots \\
\text{else if rule } L \text{ then } u_{i,m} &= \pi_{i,m}^L
\end{align*}
$$

The above (17) shows the fact that we need to prepare a big loop which consists of $L = \prod_{i=1}^{n} R_i$ if-then-else commands in the big loop, which is a cumbersome job for programmers. It is obvious that the computational complexity of LSA is in the order of $O(\prod_{i=1}^{n} R_i / 2)$. On the other hand, the HSA for a FLTC with $n$ inputs is shown as follows:

$$
\begin{align*}
\Phi(x_1(t)) &= 1 + H_z(x_1(t) - \eta^1_1) + \ldots + H_{s_k}(x_1(t) - \eta^k_1), \\
\Phi(x_2(t)) &= 1 + H_z(x_2(t) - \eta^1_2) + \ldots + H_{s_k}(x_2(t) - \eta^k_2), \\
&\vdots \\
\Phi(x_n(t)) &= 1 + H_z(x_n(t) - \eta^1_n) + \ldots + H_{s_k}(x_n(t) - \eta^k_n)
\end{align*}
$$

where $k = 1, 2, ..., m$. The illustration for HSA is shown in Figure 3 and it is obvious that the coding of HSA requires only $n$ small loops to find their respective indices, that only consists of $\sum_{i=1}^{n} R_i$ if-then-else commands, which is much simpler than that of LSA.
Figure 3. The illustration of FLT using Heaviside searching algorithm.

Figure 4. The comparison of computational complexities between HSA and LSA with three DBPMFs for each input.

Note that the number of if-then-else commands in coding is identical to the order of computational complexity. According to (18) and Figure 3, the computational complexity for Heaviside search algorithm is in the order of \( O(\sum R_i) \). The following experiment compares the computational complexities of both HSA and LSA by increasing the number of input from 3 to 7 with fixed 3 DBPMFs for every input \( x_i \) (\( R_i = 3 \)), which is shown in Figure 4. It is obvious to find that the computational complexity of HSA is significant lower than that of LSA while increasing the number of inputs. The table searching time of HSA is generally faster than LSA with an approximate ratio of \( \left( \prod_{i} R_i \right) / \left( 2^m \sum R_i \right) \). It is worth mentioning that the coding of HSA is also much simpler compared with traditional LSA, especially for the FLTC with large amount of inputs or DBPMFs.

6. Conclusion
This paper proposes a new interpretation of de-fuzzification via Center Average De-fuzzification when the input MFs are the Disjoint Block Pulse Membership Functions (DBPMFs). This will result in a Fuzzy Lookup Table Controller (FLTC). We also propose a new efficient Heaviside Search Algorithm (HSA) for a much more efficient table search performance. In comparison with the traditional Linear Search Algorithm (LSA) for table search, the search performance of HSA is not only superior, but the coding simplicity is also obtained. The balancing of Inverted Pendulum System (IPS) is adopted as the benchmark to show the feasibility of FLTC. The control performance of adopting a simple 5 by 5 FLTC is much better than that of the traditional Fuzzy Logic Controller (FLC). The comparison of computational complexities for HSA and LSA is also performed. For more inputs, the performance of HSA will be dramatically much better than that of LSA.

Acknowledgments
This work was supported by the Ministry of Science and Technology (MOST) of Taiwan (No. 106-2221-E-009 -062 -).
[6] Hwang C L 2008 Microprocessor-based fuzzy decentralized control of 2-D piezo-driven systems IEEE Trans. Ind. Electron. 55(3) 1411-20

[7] Wang C H and Wen J S 2008 On the equivalence of a table lookup (TL) technique and fuzzy neural network (FNN) with block pulse membership functions (BPMFs) and its application to water injection control of an automobile IEEE Trans. Syst., Man, Cybern. C, Applicat. Rev. 38(4) 574-80

[8] Wang L X 1997 A Course in Fuzzy Systems and Control (Upper Saddle River, NJ: Prentice Hall)

[9] Jung S, Cho H T and Hsia T C 2007 Neural network control for position tracking of a two-axis inverted pendulum system: Experimental studies IEEE Trans. Neural Netw. 18(4) 1042-48

[10] Huang C H, Wang W J and Chiu C H 2011 Design and implementation of fuzzy control on a two-wheel inverted pendulum IEEE Trans. Ind. Electron. 58(7) 2988-3001