

Optical Fuzzy Logic for Low-Power Satellite Controller

Elizabeth Golden
Computer Science Dept.
University of Idaho at Idaho Falls
1776 Science Center Drive, Suite 306,
Idaho Falls, ID 83402
Email: egolden@poky.net

Milos Manic, Ph.D
Department of Computer Science.
University of Idaho at Idaho Falls
1776 Science Center Drive, Suite 306,
Idaho Falls, ID 83402
Email: misko@ieee.edu

Abstract—Power consumption is often a major advantage of optics. When access to power is limited, such as for a satellite in geosynchronous orbit, power is often not only important, but critically important. Satellites rely on power for motion control, controlling yaw, pitch, and roll. An optical Fuzzy controller would overcome the problem of limited power. This paper explores the possibility of optical implementation of fuzzy logic for low-power optical fuzzy controllers. Recent advances in optical logic have suggested ways to overcome the problems that have plagued that field for over 40 years. In this paper the authors overviews recent advances in the optical implementation of Boolean logic and explores whether these or similar technologies might feasibly be applied to optical implementation of fuzzy logic. Specifically, the authors examines whether fuzzy logic might be productively implemented in an interferometric network in which weighting is accomplished by optical phase shifting of mutually coherent beams of light. This paper produces a optical fuzzy OR and lays the foundation for other fuzzy operators based on an interferometric network.

I. INTRODUCTION

A satellite in geosynchronous orbit around the earth is controlled through adjustments of its yaw, pitch, and roll [1]. A controller for satellites has limited power onboard and is subject to radiation, and noise. For electronic controllers both these represent problems, but for optical controllers these problems can be easily overcome. Photons are fundamentally different from electrons and are not susceptible to noise, heat, and radiation as electrons are.

Fuzzy logic can be most readily applied to systems like those of the satellite where information is inherently fuzzy. Some applications of fuzzy logic are pattern recognition of brain structures [2], fuzzy-logic schedulers [3] voter turnout [4], software agent bidding strategies [5], financial prediction/control [6] and many others. There are also many examples of optical fuzzy logic systems [7] - [13]. All these fuzzy logic systems tend to be spatial, nonlinear, and have problems with calculability.

Rethinking the way optics is used in optical logic implementations has led to a new logical paradigm. Can this new logical paradigm be applied to optical fuzzy logic? Can the lessons learned in Optical logic be applied to Fuzzy Controllers? This paper explores the techniques used in optical logic and applies them to the field of Fuzzy Logic.

II. FUZZY LOGIC

Fuzzy logic, introduced in 1965 by Lotfi A. Zadeh [14], takes Boolean logic with its discreet domain $\{0,1\}$ into a fuzzy domain $[0,1]$. The fuzzy domain best represents the real world where nothing is black and white. For example, let's take a look at temperature. Hot may be considered anything over 95 degrees Fahrenheit, but what about 94 degrees is that not hot also, what about 93 degrees? Where do we draw the line between warm and hot? In Boolean we have a crisp set which has hot as anything over 95 degrees and as such we have 94 degrees as warm. In fuzzy logic we can say that 94 degrees belongs to hot with a certain degree and also belongs to warm with a certain degree. Figure (1) shows this relationship, a temperature of 88 belongs to warm with a membership degree of 0.36 as indicated by the lower dotted line of figure 1 and to normal with a membership degree of 0.46 as indicated by the upper dotted line of figure 1.

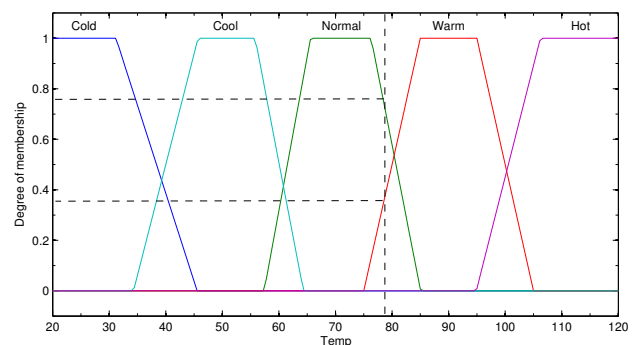


Fig. 1. Fuzzy Temperature

III. FUZZY LOGIC OPERATIONS

Boolean logic has as its main logical operators :

- Conjunction : AND (Table I)
- Disjunction : OR (Table II)
- Complement : NOT (Table III)

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

TABLE I
CONJUNCTION (AND) TRUTH TABLE

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

TABLE II
DISJUNCTION (OR) TRUTH TABLE

A	NOT A
0	1
1	0
0	1
1	0

TABLE III
COMPLEMENT (NOT) TRUTH TABLE

A	1 - A
0.3	0.7
0.8	0.2
0.2	0.8

TABLE VI
FUZZY COMPLEMENT (1 - A) TRUTH TABLE

- Monotonicity : $b \leq d$ implies $u(a,b) \leq u(a, d)$;
- Commutativity : $u(a,b) = u(b,a)$;
- Associativity : $u(a, u(b, d)) = u(u(a,b), d)$.

IV. DIRECTED LOGIC: AN OVERVIEW

Any optical switch needs a controlling mechanism to switch light from one path to another. A simple example would be a Fredkin gate [17].

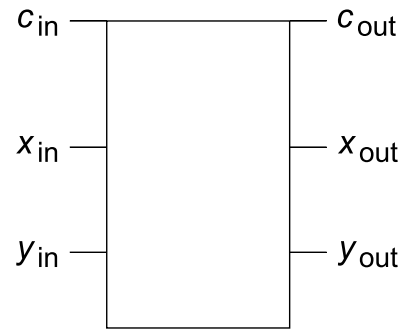


Fig. 2. Fredkin Element

The operators in fuzzy logic as suggested by Zadeh [15] are:

- Conjunction : minimum (A, B)(Table IV)
- Disjunction : maximum (A, B)(Table V)
- Complement : $1 - A$ (Table VI)

A	B	Min
0.2	0.3	0.2
0.2	0.8	0.2
0.7	0.3	0.3
0.7	0.8	0.7

TABLE IV
FUZZY CONJUNCTION (MIN) TRUTH TABLE

A	B	Max
0.2	0.3	0.3
0.2	0.8	0.8
0.7	0.3	0.7
0.7	0.8	0.8

TABLE V
FUZZY DISJUNCTION (MAX) TRUTH TABLE

Although Min and Max are the most frequently used logical operators in fuzzy logic many more are used such as: sum, bounded sum, winner takes all, weighted average for disjunction and product and bounded difference for intersection [16]. Any fuzzy operators must satisfy the following requirements [16], for any $a,b,d \in \{x|x \in C, |x| \leq 1\}$:

- Boundary Conditions : $u(a,0) = a$;

The Fredkin gate (Figure 2) is similar to many other 2 X 2 switches in that the all three inputs, control, X, and Y, are treated as logic inputs. This has been one of the problems that optical logic has faced over the last 45 years. The control, in optical switches, is often of a different type than that of the remaining two logical inputs such as electrical or light with a different polarization. An example of this problem can be seen in several implementations of Fredkin [18] that have the control in the electrical domain, while the remaining two logic inputs are in the optical domain. In [19] all three inputs are in the optical domain with the control input having a different wavelength to the remaining two logic inputs. This has caused problem with cascading logical terms, the output of the logical operation would need to be converted to the same domain as the control for the next logical operation. Directed logic still uses the Fredkin gate, but in a restricted form where the control and logic inputs are never interchanged. In order to still be able to cascade logic functions directed logic [20] overcame this problem by acknowledging that photons were fundamentally different from electrons and by rethinking the way the logic was done. Instead of cascading by chaining one logic operation after another, nesting rather than chaining was proposed. This nesting negated the problems of differing control and input domains. It is analogous to infix notation rather than suffix notation.

Directed logic circuits are networks of the restricted Fredkin elements. Directed logic performs distributed parallel computation on a function and its negation, no single Fredkin element performs a logical operation. The circuit as a whole performs the logical function, this will be even more obvious when cascading is introduced into directed logic. The term directed logic actually refers to the nature of the architecture. Light is "directed" by the logical inputs switching the light to an output such as OR or NOR. The operation of each element is independent of the operation of the other elements in the circuit. Computation of the logical function is performed only by the circuit as a whole by directing the light to the correct output. No single element within the circuit as a whole can be identified as a sub-function. For a simple logical function a circuit computes both the logical function and its inverse. Thus, any circuit that computes AND (Figure 3) also computes NAND.

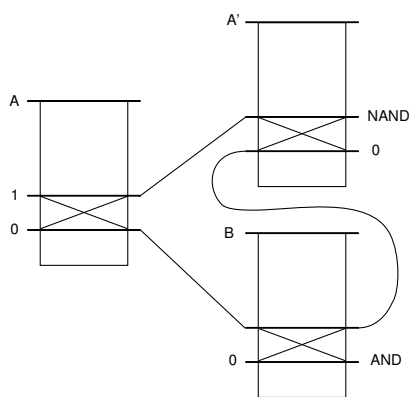


Fig. 3. AND/NAND circuit

Directed logic is Boolean logic system where a "False", Boolean 0, passes light directly through the switch without switching. The pass is the identity operation, its output is the same as its input. A "True", a Boolean 1 input, switches light between paths. The switch negates its input. The switch and the pass take only argument and are thus monadic. A switching of input (1,0) is the same as a pass of input (0,1), $Switch(1,0) = Pass(0,1) = (0,1)$. Similarly, $Switch(0,1) = Pass(1,0) = (1,0)$, $Switch(0,0) = Pass(0,0) = (0,0)$, and $Switch(1,1) = Pass(1,1) = (1,1)$. If we interpret (1,0) as "True" and (0,1) as "False", then sequences of Switch and Pass can be used to calculate certain Boolean functions on given arguments. We can compute Boolean XOR and XNOR with a string of Switching and Passing elements Figure 4.

The basic XOR/XNOR operation is one of the simpler operations in directed logic. In order to compute XOR's with n operators requires only n elements. The XOR gate computes XOR simply through the relationship of the number of even Boolean 1s to an XOR "True," if there are an even number of Boolean 1s then the XOR will return "True." If there are an odd number of 1's then the XNOR will return "True." For example, to calculate $XOR(v_1, v_2, \dots, v_n)$ requires the

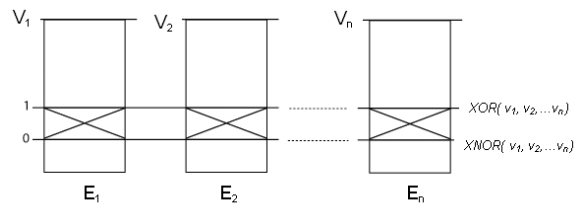


Fig. 4. Simple XOR/XNOR computed with a series of directed logic elements

string of elements E_1, E_2, \dots, E_n where E_i is Pass if $v_i = 0$ and Switch if $v_i = 1$. In Figure 4 the flow of light through the elements is directed by the control input v_1, v_2, \dots, v_n , E_1 receives the input (1,0) and thereafter the output of each element is the input to the next.

The simple XOR/XNOR shown here does not allow for cascading with other logical operators. For that reason when XOR's need to be cascaded another structure is used, figure 5. The full XOR/NOR is more complex and can be understood easier by taking a look at the other logic circuits such as the OR/NOR.

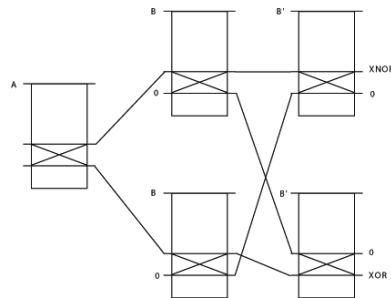


Fig. 5. Fully Cascadable XOR/XNOR

The OR/NOR circuit depicted in Figure 6 starts with the input vector (1,0).

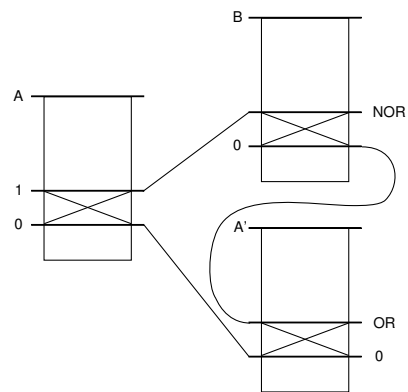


Fig. 6. OR/NOR circuit in directed logic

The information signal, A, is introduced into the two elements, A and A'. The input B activates the operation of element B at the same time as input A. The output vector of the initial element is split into its component scalars which

are then composed with other scalars to form new vectors that serve as the inputs to subsequent elements. Although it is tempting to see the intermediate scalars as Boolean values, this should be resisted. In some instances both outputs of an element will be 0. (This happens in the B element of Figure 6 when $A = 0$.) If these were true Boolean values, this would amount to a violation of the law of excluded middle.

It is worth spending a bit of time understanding what is going on in Figure 6. The output vector of the first A element is split into two paths, it may be helpful to think of the top path as the negative path and the bottom path as the positive path. One of these two paths will carry the scalar 1, the other will carry the scalar 0. In this sense the position of the scalar 1 carries the information of the value of A. If A is positive (the scalar takes the bottom path), then it is switched to the OR output line without the need to check the value of B. If, on the other hand, A is negative, then it becomes necessary to check the value of B. A negative result from B yields the scalar 1 at NOR, a positive result sends it down to the second A element. Since the scalar 1 only passes through B when A is negative, it will be passed through the second A element to the OR output. The reader should take the time to convince himself that the 1 will always arrive at either OR or NOR while the other will have a scalar 0. The extra 0 input at B merely ensures that every path has either a 1 or a 0 scalar. It should be reiterated here that the (1, 0) input vector elements are maintained throughout the whole network and are merely redirected.

For convenience, the switches in directed logic are discussed in terms of the switch being a Mach-Zehnder interferometer (MZI), but the results obtained have also been achieved in silicon. The change of path length that switched the output from one of two possible output paths were made by adjusting either polarization or phase modulation in one arm of an MZI.

V. OPTICS BACKGROUND

In recent work in the optics field the old problem of how to do logic in optics has been solved [20], [21]. In the past optical logic implementations tried to mimic electronic logic with optics. This failed because photons don't behave the same way as electrons. The main components used in the new optical logic architecture were the Mach-Zehnder interferometer (MZI) (figure 7) in bulk optics and the waveguide in its silicon counterpart as a 2 x 2 switch.

The output was controlled by adjusting one arm of the MZI/waveguide to change the path length in the MZI/waveguide. The light is split by the first beam-splitter with part of the light source being reflected to the top mirror and part passing through on to the lower mirror. The two light paths then pass through the top beam-splitter and emerge through the two output paths. The input light is given by amplitude A and a phase of θ , providing an input of $Ae^{i\theta}$.

In directed logic (section IV), an optical Boolean logic architecture, only one of the light sources is generally used and the output path is controlled by a π phase change along one of the arms. In the case of A' light is input at both inputs.

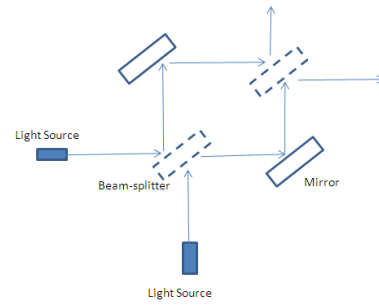


Fig. 7. Basic Mach-Zehnder Interferometer (MZI)

The A' MZI is used to reintegrate the A and B MZIs to produce an OR output. The A' is the opposite to the A MZI as shown in figure 6. The output of the light passing through the first beam-splitter of figure 8 to the arm marked as (a) is given by formula (1).

$$\frac{Be^{i\psi}}{\sqrt{2}} + \frac{Ae^{i\theta + (\frac{\pi}{2})}}{\sqrt{2}} \quad (1)$$

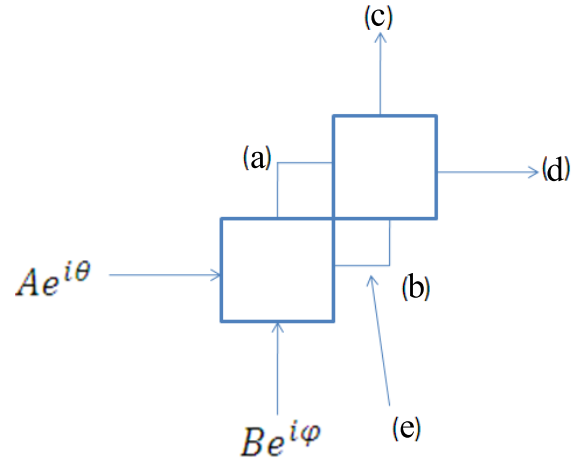


Fig. 8. Mach-Zehnder Interferometer (MZI)

The output of the light passing through the first beam-splitter of figure 8 to the arm marked as (b) is given by formula (2).

$$\frac{Ae^{i\theta}}{\sqrt{2}} + \frac{Be^{i\psi + (\frac{\pi}{2})}}{\sqrt{2}} \quad (2)$$

The portion of light that is split at the beam-splitter at 90 degrees has an addition of $\pi/2$ to the phase.

The light at position (a) in figure 8 then continues through the second beam-splitter where it is combined with light at position (b) in figure 8. The result is given by formula (3) and (4) at positions (c) and (d) of figure 8.

$$\frac{Ae^{i\psi}}{\sqrt{2}} + \frac{Ae^{i\psi + (\pi)}}{\sqrt{2}} + \frac{Be^{i\theta + (\frac{\pi}{2})}}{\sqrt{2}} + \frac{Be^{i\theta + (\frac{\pi}{2})}}{\sqrt{2}} \quad (3)$$

VI. FUZZY DIRECTED LOGIC

$$\frac{Be^{i\psi}}{\sqrt{2}} + \frac{Be^{i\psi+(\pi)}}{\sqrt{2}} + \frac{Ae^{i\theta+(\frac{\pi}{2})}}{\sqrt{2}} + \frac{Ae^{i\theta+(\frac{\pi}{2})}}{\sqrt{2}} \quad (4)$$

The two light beams with phase A at position (c) represented by formula (3) destructively interfere and the result of the light sources with amplitude B is given by (e), which is the same as the input, but with a π phase change.

$$Be^{i\psi+(\pi)} \quad (5)$$

The two light beams with phase B at position (d), represented by formula (4), destructively interfere. The result of the light sources with amplitude A is given by formula 6, which is again the same as the input, but with a π phase change.

$$Ae^{i\psi+(\pi)} \quad (6)$$

As can be seen the overall result is an unchanged amplitude and an additional π change of the phase. If only one input beam was entered the result would be only one output at position (d) of figure 8.

These results are obtained with a zero phase change at (e) of figure 8. If a π phase change was introduced at (e) of figure 8 with only one input of formula (7) our results would be different.

$$Ae^{i\psi} \quad (7)$$

Formula (1) of figure 8 would result in formula (8) and formula (2) would result in formula (9).

$$\frac{Ae^{i\psi+(\frac{\pi}{2})}}{\sqrt{2}} \quad (8)$$

$$\frac{Ae^{i\psi}}{\sqrt{2}} \quad (9)$$

After passing through the final beam-splitter we have formula (10) at position (c) of figure 8 and formula (11) at position (d) of figure 8.

$$\frac{Be^{i\psi+(\pi)}}{\sqrt{2}} + \frac{Be^{i\psi+(\pi)}}{\sqrt{2}} \quad (10)$$

$$\frac{Be^{i\psi+(\frac{\pi}{2})}}{\sqrt{2}} + \frac{Be^{i\psi+(\frac{3\pi}{2})}}{\sqrt{2}} \quad (11)$$

The light at position (d) of figure 8 destructively interferes to zero leaving all the light at position (c) of figure 8. This allows directed logic to switch on 1 (π) and pass light through the MZI on zero.

This paper concentrates on producing an optical fuzzy logic OR/NOR. Directed logic circuits produce OR and NOR in one circuit along with the AND and NAND in one circuit. The OR/NOR directed logic circuit Figure 6 made from three Mach-Zehnder interferometers is shown in figure 9.

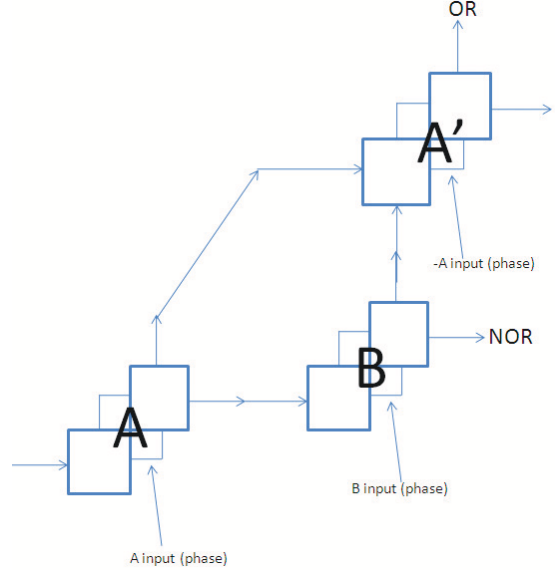


Fig. 9. Fuzzy OR/NOR circuit

Input A has a phase change of δ , B has a phase change of ϵ and A' has a phase change of ω . The A' is a reverse of A so $\omega = -\delta$. In directed logic a zero phase change on one of the arms of an MZI is considered an input of Boolean one and a π phase change is considered an input of Boolean zero. Since Boolean logic is a subset of fuzzy logic where inputs and outputs are restricted to $\{1,0\}$ a fuzzy directed logic implementation must also provide the same outputs for inputs of $\{1,0\}$. The fuzzy values used in this implementation are as multiples of π . For example, a phase change of $\frac{\pi}{2}$ is an input of 0.5 and a phase change of $\frac{\pi}{4}$ is an input of 0.25. As shown in section 4 whenever light is reflected at 90 degrees a $\frac{\pi}{2}$ phase is added this type of directed logic switch. When the light is passed directly through the beam-splitter no phase change is added. When one arm of the MZI has an additional phase change of δ , ω , or ϵ then these values are added to the phase. The resultant formula of the calculations for the OR output are given by formula 12. The calculation is simplified by setting "A" equal to 1 and θ to zero. The amplitude of the output is then the sum of the cosine of each exponent of 12.

$$\begin{aligned} & \frac{Ae^{i\theta+\frac{\pi}{2}+\epsilon} + Ae^{i\theta+\frac{3\pi}{2}+\epsilon+\omega} + Ae^{i\theta+\frac{\pi}{2}+\delta+\epsilon} + Ae^{i\theta+\frac{3\pi}{2}+\delta}}{8} \\ & + \frac{Ae^{i\theta+\frac{3\pi}{2}+\epsilon+\omega+\delta} + Ae^{i\theta+\frac{5\pi}{2}+\omega} + Ae^{i\theta+\frac{3\pi}{2}} + Ae^{i\theta+\frac{5\pi}{2}+\omega+\delta}}{8} \\ & + \frac{Ae^{i\theta+\frac{3\pi}{2}+\omega} + Ae^{i\theta+\frac{3\pi}{2}+\omega+\delta} + Ae^{i\theta+\frac{3\pi}{2}} + Ae^{i\theta+\frac{\pi}{2}+\delta}}{4} \quad (12) \end{aligned}$$

The actual calculation is simplified by the fact that any term without a δ , ω , or ϵ is a constant and because of the relationship between δ and ω , i.e. $\omega = -\delta$. The formula for the NOR output is given formula 13.

$$\frac{Ae^{i\pi} + Ae^{i\pi+\delta} + Ae^{i\pi+\epsilon} + Ae^{i\pi+\delta+\epsilon}}{4} \quad (13)$$

There is one other output for the OR NOR directed logic circuit, the junk out, in Boolean directed logic this output is always zero. This, however, may not be the case in fuzzy directed logic. The formula for this output is given by formula 14.

$$\frac{Ae^{i\frac{\pi}{2}+\epsilon} + Ae^{i\frac{5\pi}{2}+\omega} + Ae^{i\frac{\pi}{2}+\delta+\epsilon} + Ae^{i\frac{3\pi}{2}} + Ae^{i\frac{3\pi}{2}+\delta}}{8} + \frac{Ae^{i\frac{3\pi}{2}+\epsilon+\omega} + Ae^{i\frac{3\pi}{2}+\delta+\epsilon+\omega} + Ae^{i\frac{5\pi}{2}+\delta+\omega}}{8} + \frac{Ae^{i\frac{3\pi}{2}} + Ae^{i\frac{\pi}{2}+\delta} + Ae^{i\frac{3\pi}{2}+\omega} + Ae^{i\frac{\pi}{2}+\omega+\delta}}{4} \quad (14)$$

VII. ANALYSIS

The formula from 12 provides a means for calculating and verifying the fuzzy OR outputs. The resultant fuzzy values, for various inputs, can be seen in table VII and the surface plot for these values can be seen in figure 10. Table VII shows that for the Boolean inputs 0, 1 the fuzzy OR outputs are consistent with Boolean logic. Since Boolean logic is a subset of fuzzy logic it is important that Boolean logic holds in any fuzzy operator. In addition the fuzzy OR operator must satisfy all the requirements for a fuzzy operator; boundary conditions, monotonicity, commutativity, and associativity.

Fuzzy A	Fuzzy B	Output
1	1	1
1	0	1
0	1	1
0	0	0
0.50	0.50	0.75
0.75	0.11	0.86
0	0.25	0.15
0.90	0.30	0.98
0.11	0.40	0.37
0.25	0.50	0.57
0.30	0.60	0.73
0.40	0.75	0.90
0.60	0	0.65

TABLE VII
FUZZY OR OUTPUTS

A. Boundary Conditions

Boundary conditions : $u(a,0) = a$

The boundary conditions state that if one input is zero then the output is equal to the non zero input. Table VIII shows the output values for $\{.25, 0\}$, $\{.5, 0\}$, and $\{.75, 0\}$. As can be seen the fuzzy OR gate does not agree with the

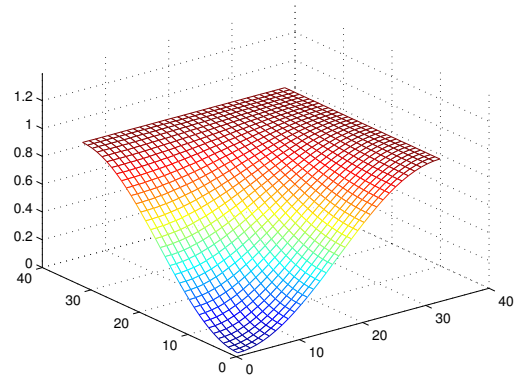


Fig. 10. Fuzzy OR graph

given boundary condition. Here, the fuzzy OR over estimates high input values and underestimates low input values. For the half way point the boundary conditions are met in full. There is, in the literature, other examples of weaker boundary conditions [22].

Fuzzy A	Fuzzy B	Output
0.25	0	0.15
0.5	0	0.5
0.75	0	0.85

TABLE VIII
BOUNDARY CONDITIONS FUZZY OR

B. Monotonicity

monotonicity : $b \leq d$ implies $u(a,b) \leq u(a, d)$

Monotonicity state that if $b \leq d$ than the disjunction of any third value a, with b should also be less than the disjunction of a with d. As table IX shows with $a = .5$, $b = .25$, and $d = .5$ $b \leq d$ and the output $u(a,b) = .57$ is less than $u(a,d) = .75$.

A	D	B	Output
0.5	0.5		0.75
0.5		0.25	0.57

TABLE IX
MONOTONICITY U(A,B), U(A,D) FUZZY OR

C. Commutativity

Commutativity : $u(a,b) = u(b,a)$

Commutativity states that the results from $\{a,b\}$ should be the same as $\{b,a\}$. Table X shows the outputs for $a = .25$ $b = .5$ and $a = .5$ and $b = .25$. The outputs in both case are the same. Table 9 also shows this with the Boolean values $\{1, 0\}$ and $\{0, 1\}$

A	B	Output
0.25	0.5	.57
0.5	0.25	.57

TABLE X
COMMUTATIVITY $u(A,B)$, $u(A,D)$ FUZZY OR

D. Associativity

Associativity : $u(a, u(b, d)) = u(u(a,b), d)$

Associativity states that A or $(B$ or $C) = (A$ or $B)$ or C . To show associativity in this case would require the investigation of cascading. Directed uses a novel system of cascading that requires its own explanation. Because of this cascading is out of the scope of this paper and will be left for further investigation.

The fuzzy OR operator satisfies all the requirements for a fuzzy operator; boundary conditions, monotonicity, commutativity, and associativity. These show that a fuzzy OR can indeed be produced from the Boolean optical directed logic circuit.

The graph for the NOR output is shown in figure 11, in this case the Fuzzy NOR does appear to be the opposite to fuzzy OR. There does appear to be an issue with fuzzy NOR as table XI shows. Here the output for $\{1,1\}$ is shown to be 0, which is correct, but the output for $\{.5 .5\}$ also is shown to be zero. These two sets of inputs should not be the same and would indicate a problem with fuzzy NOR. In Boolean directed logic the junk out is ignored since it is unused. In fuzzy logic the problem with fuzzy NOR would seem to indicate that this junk out should not be ignored and reintegrated with the NOR. Because no overall amplitude for the circuit can be lost the NOR given by $1 - OR$ must be reflected in the junk out and the NOR combined.

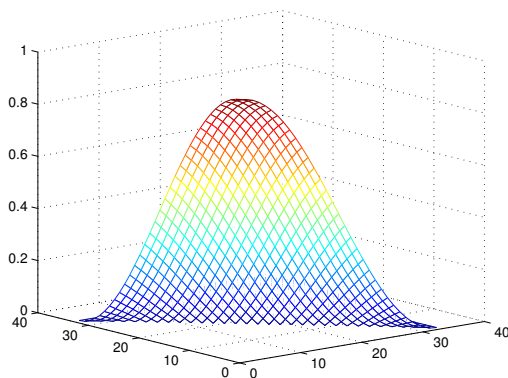


Fig. 11. Fuzzy NOR graph

VIII. CONCLUSION

Directed logic is a low-power Boolean system of logic that has extended Boolean logic from the electronic domain to

Fuzzy A	Fuzzy B	Output
1	1	0
1	0	0
0	1	0
0	0	1
0.50	0.50	0
0.75	0.11	0.08
0	0.25	0.85
0.90	0.30	0.04
0.11	0.40	0.55
0.25	0.50	0.25
0.30	0.60	0.08
0.40	0.75	0.94
0.60	0	0.34

TABLE XI
FUZZY NOR OUTPUTS

the optical domain. Although directed logic had its beginning in the optical domain it can just as easy to implement in electronics. This paper begins to extend optical directed logic to include fuzzy logic. Initially, the research was aimed at producing an fuzzy optical OR/NOR. The results, so far, have indeed shown that optical Fuzzy logic using the directed logic paradigm is indeed achievable. The research carried out so far was unable to show fuzzy NOR. Directed logic has three outputs per OR/NOR circuit. In Boolean logic only two of these outputs are used, one provides the OR output another provides the NOR output. The third output is a junk out and is unused in Boolean logic. Further research should be undertaken to study the nature of the junk output. Reintegrating the junk out with the NOR output may produce a fuzzy NOR gate. The directed logic AND/NAND gate and other logic gates should also be investigated for application to fuzzy logic. Once all the optical fuzzy logic gates are in place a low-power optical fuzzy controller for satellite motion control becomes practicable.

REFERENCES

- [1] Willem H. Steyn. Fuzzy Control for a Non-linear MIMO Plant Subject to Control Constraints. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(10):1565-1571, October, 1994.
- [2] Omer Faruk Bay and Ali Bülent Usakli. Survey of Fuzzy Logic Applications in Brain-Related Researches. *J. Med. Syst.*, 27(2):215-223, 2003.
- [3] Matsuoka, K. and Divakaruni, S. M. Fuzzy logic applications in electric utility power plants. *Proc. SPIE Applications of Fuzzy Logic Technology*, Bruno Bosacchi; James C. Bezdek; Eds., 2061:122-139, 1993.
- [4] Ramot, D.; Friedman, M.; Langholz, G.; Kandel, A. Complex fuzzy logic. *Fuzzy Systems, IEEE Transactions on*, 11(4):450-461, August, 2003.
- [5] He, M., Leung, H. and Jennings, N. R. A fuzzy logic based bidding strategy for autonomous agents in continuous double auctions. *IEEE Trans on Knowledge and Data Engineering*, 15(6):1345-1363, 2003.
- [6] James Matthews. An Introduction to Fuzzy Logic. <http://www.generation5.org/content/1999/fuzzyintro.asp>, December 1999.
- [7] Senmao Lin, Shugun Zhang, Caisheng Chen, Itsuo Kumazawa. Optical multiple-variable fuzzy logic array using shadow casting. *NMicrowave and Optical Technology Letters*, 10(2):106-109, 1993.

- [8] Lin, Senmao; Kumazawa, Itsuo A new nonlinear spatial encoding method for cascaded fuzzy logic. *Optics Communications*, 107(5-6):461-470, May, 1994.
- [9] E. Gur, D. Mendlovic, and Z. Zalevsky Optical Generation of Fuzzy-Based Rules. *Applied Optics*, 41:4753-4761, 2002.
- [10] Eran Gur, David Mendlovic, and Zeev Zalevsky Optical implementation of fuzzy-logic controllers. *Applied Optics*, 37(29):6937-6945, 2002.
- [11] Liren Liu OPTICAL IMPLEMENTATION OF PARALLEL FUZZY LOGIC. *Optics Communications*, 73(3):183-187, 1989.
- [12] Hideo Itoh, Seiji Mukai, and Hiroyoshi Yajima Optoelectronic fuzzy inference system based on beam-scanning architecture. *Applied Optics*, 33(8):1485-1490, march, 1994.
- [13] Shuqun Zhang, Caisheng Chen Parallel optical fuzzy logic gates based on spatial area-encoding technique. *Optics Communications*, 107:11-16, April, 1994.
- [14] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8: 338353, 1965.
- [15] L.A. Zadeh. Knowledge Representation in Fuzzy Logic. *IEEE Transactions on Knowledge and Data Engineering* , 01(1):89-100, March, 1989.
- [16] Ramot, D.; Milo, R.; Friedman, M.; Kandel, A., Complex fuzzy sets. *Fuzzy Systems, IEEE Transactions on* , 10(2):171-186, April, 2002.
- [17] Edward Fredkin and Tommaso Toffoli, Conservative logic, *International Journal of Theoretical Physics*, 3(4):219-253, 1982.
- [18] Joseph Shamir, H. John Caulfield, William Micelli, and Robert J. Seymour, Optical computing and the fredkin gates *Applied Optics*, 10:1604-1607, 1986.
- [19] A. J. Poustie and K. J. Blow, Demonstration of an all-optical fredkin gate. *Optics Communication*, 174:317-320, 2000.
- [20] James Hardy and Joseph Shamir. Optics inspired logic architecture *Optics Express*, 15:150165, January 2007.
- [21] H. John Caulfield, R. A. Soref, L. Qian, A. Zavalin, and J. Hardy. Generalized optical logic elements GOLEs. *Optics Communications*, 271:365376, March 2007.
- [22] George J. Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, *Prentice Hall, NJ*, 1995.