Summing Amplifier as a Multi-Valued Logical Element for Fuzzy Control

V. VARSHAVSKY [1], V. MARAKHOVSKY [2], I. LEVIN[3], N. KRAVCHENKO[4]
[1,3,4] School of Engineering, Bar Ilan University, ISRAEL
[2] The University of Aizu, JAPAN
vviktor@zahav.net.il, marak@u-aizu.ac.jp, i.levin@ieee.org, natali_kra@list.ru

Abstract: - It is offered to implement fuzzy devices as multi-valued logic functions, using directly analog input variables and forming the output variables as analog ones as well. It is offered to use CMOS summing amplifiers as basic elements for designing appropriate circuits. It has been proved that a CMOS summing amplifier is a functionally complete element in arbitrary-valued logic. In a plenty of cases this approach enables principally simplification of fuzzy logic controllers for a broad class of applications. All mentioned above is illustrated by examples.

Key-Words: Fuzzy logic control, Fuzzy rules, Fuzzy inference, Multi-valued logic, Multi-threshold element, Summing amplifier, Operational amplifier.

1 Introduction

Fuzzy Logic Control is a methodology bridging Artificial Intelligence and traditional Control Theory. Fuzzy Logic Control is implemented in the only cases when accuracy is not of high necessity or importance.

On the other hand, as it is stated in [1], "Fuzzy Logic can address complex control problems, such as robotic arm movement, chemical or manufacturing process control, antiskid braking systems, or automobile transmission control with more precision and accuracy, in many cases, than traditional control techniques have… . Fuzzy Logic is a methodology for expressing operational laws of a system in linguistic terms instead of mathematical equations."

The Fuzzy Logic methodology [2, 3] comprises three phases:

1. *Fuzzifycation* – transformation of analogue (continuous) input variables to linguistic ones, e.g. transformation of temperature to the terms "cool", "warm", "hot" or that of speed to the terms "negative big (NB)", "negative small (NS)", "zero (Z)", "positive small (PS)", "positive big (PB)". Such transformation is realized by introduction of so-called membership functions. Both a range of value and a degree of membership defines these membership functions. For linguistic variables it is important not only which membership function a variable belongs to, but also a relative degree to which it is a member. A variable can have a weighted membership in several membership functions at the same time.

2. *Inference.* A connection between input and output variables, i.e. fuzzy controller behavior specification, is determined by the system of rules of "IF… THEN …"-type. For instance: "IF the temperature is worm THEN the speed is PS" or "IF the speed is NB THEN force is ZERO", etc. Since input linguistic variables are weighted, the output variables can be obtained weighted as well.

3. *Defuzzification.* Weighted values of output linguistic variables obtained as a result of fuzzy inference have to be transformed to analogue (continuous) variables. This procedure is also based on membership functions. Two major methods are used for defuzzyfication:

- maximum defuzzification method, wherein an output value is determined by the linguistic variable with the maximum weight;
- centroid calculation defuzzification method, wherein an output value is determined by the weighted influence of all the active output membership functions.

According to our concept, for a broad class of fuzzy controllers specification comes to building up a table connecting the input and output membership functions. Usually in this case the membership functions evenly divide the ranges of output variables' variations. Often the membership functions can be brought to even scale by increasing the number of gradations. Therefore, specification tables represent nothing but tables determining a specific multi-valued logical function. And what is more, for a number of implementations one can possibly try to neglect weighting and determining values of input linguistic variables and to implement continuous-valued variables.

The above idea was in the focus of our research. We dealt with searching for and investigating of such basic multi-valued functions, which, from the one hand, would present a complete functional basis in the multi-valued logic, and from the other hand, could be efficiently implemented by CMOS technology.

2 Summing Amplifier as a Multi-valued Logical Element

Summing amplifier's behavior, accurate to the members therein the infinitesimal order is determined by the amplifier's gain factor in disconnected condition (Fig.1), is described as follows:

$$V_{out} = \begin{cases} 0 & \text{if} \quad \sum_{j=1}^{n} \frac{R_0}{R_j} \cdot (V_j - \frac{V_{dd}}{2}) \le -\frac{V_{dd}}{2} \\ \frac{V_{dd}}{2} - \sum_{j=1}^{n} \frac{R_0}{R_j} \cdot (V_j - \frac{V_{dd}}{2}) & \textit{in } \text{other cases} \\ V_{dd} & \text{if} \quad \frac{V_{dd}}{2} \le \sum_{j=1}^{n} \frac{R_0}{R_j} \cdot (V_j - \frac{V_{dd}}{2}) \end{cases} \qquad (1)$$

where V_{dd} - the supply voltage, V_j - the voltage on j^{th} input, R_j - the resistance of j^{th} input, R_0 - the feedback resistance, and $V_{dd}/2$ - the midpoint of the amplifier. Dependence of V_{out} on

$\sum_{j=1}^{n} \frac{R_0}{R_j} \cdot (V_j - \frac{V_{dd}}{2})$ is shown in Fig.2a.

When comparing the values of $m = 2k+1$ voltage levels of the circuit with those of multi-valued variable as $x_j = \frac{2 \cdot V_j - V_{dd}}{V_{dd}} \cdot k$ and designating $R_0/R_j = \omega_j$ (1) can be represented as (2). Graphical view of (2) is shown in Fig.2,b.

Later on we will call the functional element, whose behavior is determined by system (2), a multi-valued threshold element. In the simplest case when $\omega_j = 1, \ j = 1,2,3$ we will call it the majority functions and designate as $y = maj(x_1, x_2, x_3)$.

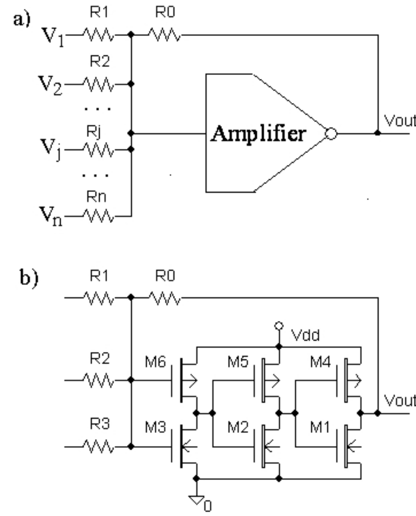

Figure 1. Summing amplifier: a) general structure; b) CMOS implementation example.

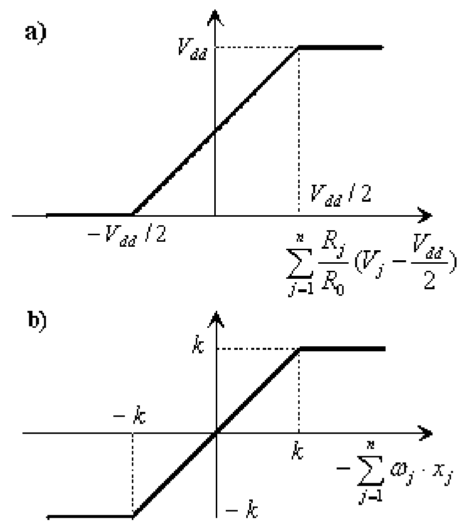

Figure 2. Summing amplifier's behavior representation: within voltage coordinates (a); within multi-valued variables coordinates (b).

$$y_j = \begin{cases} -k & \text{if} \quad \sum_{j=1}^{n} \omega_j \cdot x_j \ge k \\ -\sum_{j=1}^{n} \omega_j \cdot x_j & \text{if} \quad k > \sum_{j=1}^{n} \omega_j \cdot x_j > -k \\ +k & \text{if} \ -k \ge \sum_{j=1}^{n} \omega_j \cdot x_j \end{cases} \qquad (2)$$

3 Functional Completeness of a Threshold Element in Multi-Valued Logic

The basic operation (or set of basic operations) is called functionally completed in arbitrary-valued logic, if either function of this logic can be represented as superposition of basic operations.

There are some known functionally completed sets. So, it is clear, that for proving functional completeness of some new function it is sufficient to show that the functions of the known functionally completed set can be represented as superposition of the considered function. One of functionally completed functions in m-valued logic is the Webb function [4]:

$$w(x, y) = [\max(x, y) + 1]_{\bmod m} .$$

So, for proving functional completeness of threshold operation in multi-valued logic it is sufficient to show how the Webb function can be represented by this operation.

First of all let us represent the function $\max(x_1, x_2)$ by threshold functions. To do this let us consider the function $f_a(x)$ diagram, such as

$$f_a(x) = \max(x, a) = \begin{cases} a & \text{if} \quad a \geq x \\ x & \text{if} \quad x > a \end{cases}.$$

This function diagram is shown in Fig.3,a.

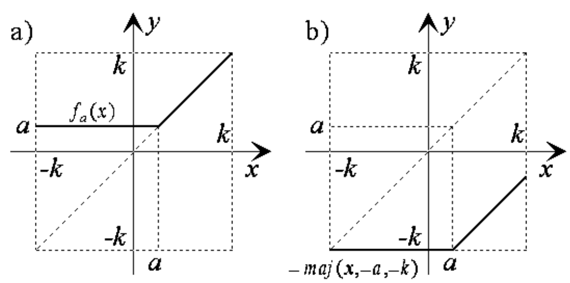

Figure 3. Diagrams of $f_a(x)$ (a) and $-maj(x,-a,-k)$ (b) functions.

The $-maj(x,-a,-k)$ function diagram is shown in Fig.3,b. Actually, as far as $x < a$ $x - a - k < -k$ and $-maj(x,-a,-k) = -k$. Note, that for all the x values $f_a(x) + maj(x,-a,-k) = a + k$, as follows from Fig.2, hence

$$f_a(x) = -maj[-maj(x,-a,-k), a, k]. \quad (3)$$

Taking into consideration $-maj(a,b,c) = maj(-a,-b,-c)$, it follows from (3):

$$\max(x_1, x_2) = maj[maj(-x_1, x_2, k), -x_2, -k] .$$

Now let us consider the function $(x+1)_{\bmod m}$ representation by threshold functions. To make it clear let us turn to the following pictures (Fig.4).

It is clearly seen from the Fig.4 that the above constructions should be considered as the way to prove functional completeness only, but in no circumstances as a synthesis method. The methods

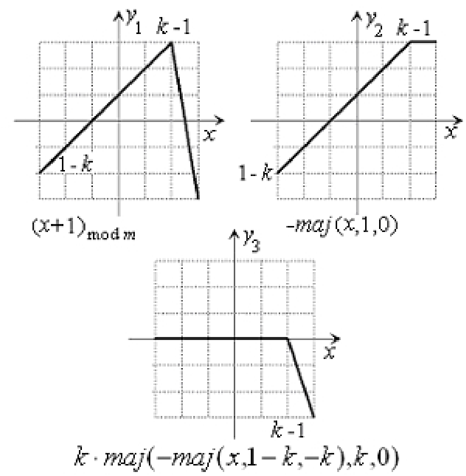

Figure 4. Implementation of $(x+1)_{\bmod m}$ function.

of synthesizing circuits in the proposed base are to be developed in future. However, as it will be shown below, for a number of real circuits the proposed base allows designing simple and efficient circuits.

4 Fuzzy Devices as Multi-Valued and Analog Circuits

Fuzzy devices or fuzzy controllers implementing fuzzy control functions usually have the structure shown in Fig.5.

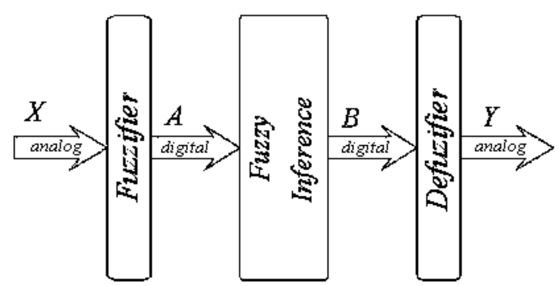

Figure 5. Fuzzy device structure.

Analog variables $X = \{x_1, x_2, ..., x_n\}$ go the fuzzy device output. Fuzzifier converts a set of analog variables x_j into that of weighted linguistic (digital) variables $A = \{a_1, a_2, ..., a_n\}$ of the type "*variable x_j has an average positive value with the weight of 0.3 and a large positive value with the weight of 0.4*".

Fuzzy Inference block, based on the fuzzy rules like "*if a_j has an average positive value and a_i has a small negative value, b_k has a small positive value*" generates a set of weighted linguistic (digital) variables values $B = \{b_1, b_2, ..., b_k\}$.

Defuzzifier converts a set of weighted linguistic (digital) variables $B = \{b_1, b_2, ..., b_k\}$ into a set of output analog variables $Y = \{y_1, y_2, ..., y_k\}$.

As a rule, Fuzzifier and Defuzzifier are implemented as ADA (analog-digital-analog) converter, i.e. by hardware implementation. Fuzzy inference is usually implemented as microprocessor software.

On the other hand, there is the set of output analog variables that unambiguously corresponds to each set of input analog variables; hence a Fuzzy Device could be specified as a functional analog signals converter

$$Y(X) = \{y_1(X), y_2(X), ..., y_k(X)\}$$

and its output Y determines a system of n-dimensional surfaces. Let us consider the possibility of specification of these surfaces as that of their piecewise-linear approximation.

Let $m = 2k + 1$ linguistic variable a_j values[1] correspond to analog variable x_j. Then basing on fuzzy rules system we can specify a system of m-valued logic functions, as follows:

$$B(a_1, a_2, ... a_n) = \{b_1(A), b_2(A), ..., b_k(A)\}. \quad (4)$$

Note that most publications describing fuzzy controllers contain the tables, specifying fuzzy controllers' behavior as (4) and a plenty of publications contain piecewise-linear approximations of the corresponding surfaces.

The apparent conclusion can be made from the things mentioned above: if a fuzzy controller is represented as (4), it can be implemented as superposition of multi-valued threshold elements. In this case, owing to linear behavior of the threshold element in the zone between the saturation levels ((2) and Fig.2,b) natural linear approximation appears between the discrete points of specification.

[1] An odd number of linguistic variable values do not break the generality of consideration.

5 Implementation of the Fuzzy Controllers as Threshold Elements Circuits

5.1 Example 1

This example is taken from [5]: *"Design of a Rule-Based Fuzzy Controller for the Pitch Axis of an Unmanned Research Vehicle"*.

The fuzzy control rules for the considered device depend on the error value $e = ref - output$ and changing of error $ce = \frac{old\, e - new\, e}{sampling\ period}$. Fuzzifier brings seven levels of linguistic variables for each (NB – negative big (-3); NM – negative middle (-2); NS – negative small (-1); ZO – zero (0); PS – positive small (1); PM – positive middle (2); PB – positive big (3)) in correspondence with error and changing of error values. Output has the same seven gradations. At that defuzzification of NB value generates $V_{out} = 0\text{V}$, that of ZO value generates $V_{out} = 1.75\text{V}$ and that of PB value generates $V_{out} = 3.5\text{V}$, if the supply voltage $V_{dd} = 3.5\text{V}$. The corresponding 49 fuzzy rules are represented in Table1.

Table 1. Table of fuzzy rules

	Error (e)							
Change of error (ce)		NB	NM	NS	ZO	PS	PM	PB
	NB	ZO	PS	PM	PB	PB	PB	PB
	NM	NS	ZO	PS	PM	PB	PB	PB
	NS	NM	NS	ZO	PS	PM	PB	PB
	ZO	NB	NM	NS	ZO	PS	PM	PB
	PS	NB	NB	NM	NS	ZO	PS	PM
	PM	NB	NB	NB	NM	NS	ZO	PS
	PB	NB	NB	NB	NB	NM	NS	ZO

Table 2 represents Table 1 as function of 7-valued logic.

Table 2. Table of the 7-valued function

	Error (e)							
Change of error (ce)		-3	-2	-1	0	+1	+2	+3
	-3	0	+1	+2	+3	+3	+3	+3
	-2	-1	0	+1	+2	+3	+3	+3
	-1	-2	-1	0	+1	+2	+3	+3
	0	-3	-2	-1	0	+1	+2	+3
	+1	-3	-3	-2	-1	0	+1	+2
	+2	-3	-3	-3	-2	-1	0	+1
	+3	-3	-3	-3	-3	-2	-1	0

It is seen from Table 1 and Table 2 that the function is symmetric with respect to "North-West – South-East" diagonal and depends on $e - ce$. This kind of dependence is shown in Fig.6.

It apparently follows from comparison of Fig.2 and Fig.6 that in order to reproduce the function specified by Table 2 it is quite sufficient to have one two-input summing amplifier and one inverter.

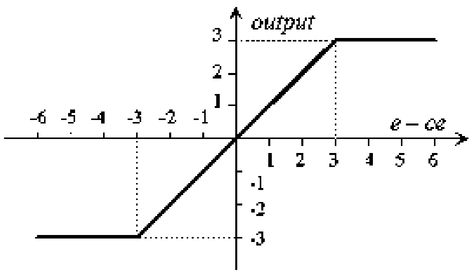

Figure 6. Graphical representation of the function, specified by Table 2.

Note that inversion of logic variables lying within $-k \div +k$ interval is the operation of diametric negation $\bar{x} = -x$; the operation $\overline{V_{out}} = V_{dd} - V_{in}$ corresponds thereto in the space of summing amplifiers' output voltages. Thus CMOS circuit containing 12 transistors and 5 resistors, which implements our function, is shown in Fig.7.

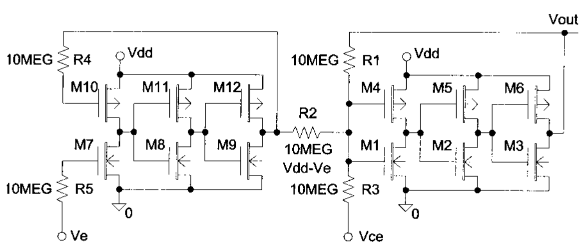

Figure 7. CMOS implementation of the fuzzy controller, specified by Tab.2.

In this circuit left six transistors M7-M12 compose the main body of the summing amplifier used as an inverter. Right 6 transistors M1-M6 compose the main body of the summing amplifier.

5.2 Example 2

This example is taken from [6]: "Manipulator for Man-Robot Cooperation (Control Method of Manipulator/Vehicle System with Fuzzy Inference)".

In the considered example the experimental manipulator has two force/torque sensors. One of them is the operational force sensor F_h; the other is "the environmental force sensor" - ω. Both of the values measured has three linguistic variables – S (small), M (middle) and B (big) and five fuzzy rules, as follows:

If $\omega = S$ then Output=B;
If $\omega = B$ then Output=S;
If $\omega = M$ and $F_h = S$ then *Output*=S;
If $\omega = M$ and $F_h = M$ then *Output*=M;
If $\omega = M$ and $F_h = B$ then *Output*=B;

In this case *Output* is three-valued logic function, specified as in Table 3.

Table 3. Table of ternary function.

		F_h		
		-1	0	+1
ω	-1	+1	+1	+1
	0	-1	0	+1
	+1	-1	-1	-1

It can be simply proved by trivial substitution that $Output = maj(2\omega, -F_h, 0)$ and CMOS implementation coincides with the circuit of Fig.7, therein $V_e = V_{F_h}$, $V_{ce} = V_\omega$ and $R3 = 5\,\text{MEG}$.

5.3 Fuzzy Controller for Washing Machine

This example is taken from Aptronix Incorporated (http://www.aptronix.com/fuzzynet).

5.3.1 Controller specification

Input variables:
Dirtiness of clothes "degree": Large (L), Medium (M), and Small (S);
Type of dirtiness "degree": Greasy (G), Medium (M), and Not Greasy (NG).

Output variable:
Wash time "minute": Very Long (VL), Long (L), Medium (M), Short (S), and Very Short (VS).

Fuzzy rules:
Fuzzy rules are represented as the Table 4.

Table 4. Matrix of linguistic variables

Wash time		Dirtiness of clothes		
		S	M	L
Type of dirt.	NG	VS	S	M
	M	M	M	L
	G	L	L	VL

According our approach the table of linguistic variables (Table 4) can be transformed into the table of multi-valued variables (Table 5).

Table 5. Matrix of multi-valued variables

Wash time (X)		Dirtiness of clothes (Y)		
		-1	0	+1
Type of dirt.	-1	-2	-1	0
	0	0	0	+1
	+1	+1	+1	+2

5.3.2 Functional decomposition

We will use as basic functions inverting summing amplifier with saturation:

$$\sigma(\sum_{j=1}^{k}\alpha_j \cdot x_j) = \begin{cases} -a & \text{if } \sum_{j=1}^{k}\alpha_j \cdot x_j \geq a \\ -\sum_{j=1}^{k}\alpha_j \cdot x_j & \text{if } -a < \sum_{j=1}^{k}\alpha_j \cdot x_j < a \\ +a & \text{if } -a \geq \sum_{j=1}^{k}\alpha_j \cdot x_j \end{cases}$$

Let's now:

$$\begin{array}{ccccc} \text{wash time} & & \varphi_1 & & \varphi_2 \\ \begin{vmatrix} -2 & -1 & 0 \\ 0 & 0 & +1 \\ +1 & +1 & +2 \end{vmatrix} & = & \begin{vmatrix} -1 & -1 & 0 \\ 0 & 0 & +1 \\ +1 & +1 & +2 \end{vmatrix} & + & \begin{vmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \end{array} \quad (4)$$

or $(wash\ time) = \sigma(-\varphi_1, -\varphi_2)$.

Let us take into consideration a function of one variable

$$\varphi_3(Y) = \sigma(0.5 \cdot Y, -2) = \begin{vmatrix} +2 & +2 & +1 \end{vmatrix}. \quad (5)$$

In (5) Y corresponds to "dirtiness of clothes", but Y here varies from –2 t0 +2 as the output value represented in the matrix of multi-valued variables, which in turn becomes input values X and Y via the feedback route, varies from –2 to +2. This assumption is necessary as the range of output value and the input value are different. Thus Y here is as follows

$$Y = \begin{vmatrix} -2 & 0 & +2 \end{vmatrix}$$

and has the values saturated at –2 and +2.

Now the following intermediate sum is introduced:

$$\phi_1 = \varphi_3(Y) - 0.5 \cdot X - 2 = \begin{vmatrix} +1, +1, & 0 \\ 0, & 0, & -1 \\ -1, -1, -2 \end{vmatrix}. \quad (6)$$

Here X corresponds to "type of dirt." And varies from –2 to 2 as follows

$$X = \begin{vmatrix} -2 \\ 0 \\ +2 \end{vmatrix}.$$

From (4) and (5) it is easy to see that $\phi_1 = -\varphi_1$, so

$$(wash\ time) = \sigma(\varphi_3(Y), -0.5 \cdot X, -2, -\varphi_2).$$

Now let us introduce next function:

$$\varphi_4(X,Y) = \sigma(X, Y, +4) = \begin{vmatrix} 0, -2, -2 \\ -2, -2, -2 \\ -2, -2, -2 \end{vmatrix} \quad (7)$$

and again like (6) we can form the second intermediate sum:

$$\phi_2 = 0.5 \cdot \varphi_4(X,Y) + 1 = \begin{vmatrix} +1, 0, 0 \\ 0, 0, 0 \\ 0, 0, 0 \end{vmatrix} = -\varphi_2. \quad (8)$$

Finally

$$(wash\ time) = \sigma(\varphi_3(Y), -0.5 \cdot X, -1, 0.5 \cdot \varphi_4(X,Y)) \quad (9)$$

On the Fig.8 we can see CMOS implementation of the (9), (5) and (7). The circuit is implemented as the superposition of four multi-valued threshold elements.

The upper left portion of Fig.8 corresponds to $\varphi_3(Y)$ in (9). Resistor $R2$ is 20 MEG and is twice $R1$ so that $Y/2$ is achieved. Resistor $R3$ is ideally 10 MEG but it is actually 9.3 MEG to be adjusted to the actual amplification gain of the portion. $R3$ realizes "-2" in $\varphi_3(Y)$, which is the lower saturation value.

The lower left portion corresponds to $0.5 \cdot \varphi_4(X,Y)$ in (9). Resistors $R5$ and $R6$, which are 9.5 MEG, are ideally 10 MEG. Resistor $R6$ is 5 MEG and is half of $R4$ to achieve "4" in $\varphi_4(X,Y)$, which is twice the upper saturation value. Resistor $R13$ connecting the lower left portion to upper right portion is 18 MEG (ideally it is 20 MEG). $R13$ is approximately twice $R10$ to realize "0.5" in $0.5 \cdot \varphi_4(X,Y)$.

The upper right portion corresponds to the main body of the summing amplifier and has itself the function to add "-1" in (9). Resistor $R11$ is 20 MEG that is twice $R10$. Thus "-1" is realized as the result of the lower saturation value "-2" multiplied by "1/2" that is $R10/R11$.

The lower right portion corresponds to $-0.5 \cdot X$ in (9). Here "0.5" is achieved by resistor $R14$ being 20 MEG, which is twice $R11$.

The result of SPICE simulation of the circuit Fig.8 is shown in Fig.9 where the horizontal and vertical axes correspond to variable Y and to the output voltage respectively. In Fig.9, when the output value is "-2", the output voltage becomes 0V. This is true at (0V, 0V). Likewise when the output value is "-1", the output voltage is around

0.9V. This is true at (1.75V, 0.9V). When the output value is "0", "1" or "2", the output voltage is around 1.8V, 2.5V or 3.2V respectively.

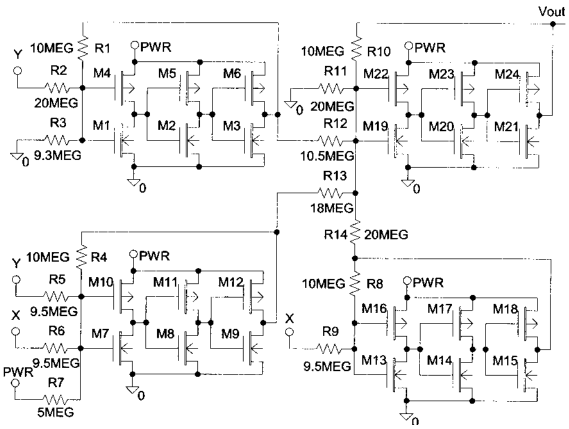

Figure 8. CMOS implementation of fuzzy controller for washing machine.

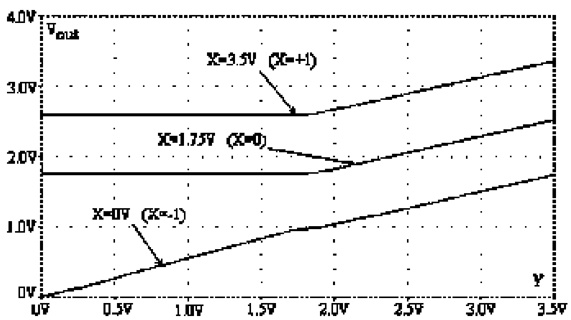

Figure 9. Results of SPICE simulation for controller on the Fig.8.

6 Conclusion

In the above examples of controllers, push-pull summing amplifiers are used. The summing amplifier however does not have to be push-pull type. It may be differential type or any other types of amplifiers.

The proof of functional completeness shown in section 2 provides the possibility of implementing arbitrary function of multi-valued logic in the base of summing amplifiers. However all mentioned above doesn't answer the question concerning the efficiency of such implementation and the techniques of synthesizing circuits in the offered base. Though the given examples show possible high efficiency and effectiveness of the implementation offered, this is true only in regard to specific examples, chosen in special way.

Techniques of synthesizing fuzzy devices in the offered base and the problems of implementability under the conditions of real production should be resolved on further work stages.

References:

[1] "*Implementation of Fuzzy Logic*", Texas Instruments, SPRA028, January 1993.

[2] "*An Introduction to Fuzzy Logic Applications in Intelligent Systems*" (Kluwer International Series in Engineering and Computer Science, 165) by Ronald R. Yager, Lotfi A. Zadeh (Editor), Kluwer Academic Publishers, January 1992, 356 p.

[3] "*Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems*": Selected Papers by Lotfi A. Zadeh (Advances in Fuzzy Systems - Applications and Theory), by George J. Klir (Editor), Bo Yuan (Editor), Lotfi Asker Zadeh, World Scientific Pub Co; Vol. 6, June 1996, 826 p.

[4] Post E., "Introduction to a general theory of elementary propositions", *Amer. J. Math.,* 43, 1921, pp. 163-185.

[5] Deepak Sabharwal and Kuldip S. Rattan, "Design of a Rule-Based Fuzzy Controller for the Pitch Axis of an Unmanned Research Vehicle", Department of Electrical Engineering, Wright State University, Dayton, OH 45435, CH3158-3/92/0000-0449©1992 IEEE, *Fuzzy Control Systems*, CRC Press LLC, Abraham Kandel (Edited by), Lotfi A. Zadeh (Foreword by), September 1993, pp.81-86.

[6] Yoshio Fujisawa, Toshio Fukuda, Kazushiro Kosuge, Fumihito Arai, Eiji Muro, Haruo Hoshino, Takashi Miyazaki, Kazuhiko Ohtsubo, Kazuo Uehara, "Manipulator for Man-Robot Cooperation (Control Method of Manipulator/Vehicle System with Fuzzy Inference)", 0-7801-0582-5/92 ©1992 IEEE", *Fuzzy Control Systems*, CRC Press LLC, Abraham Kandel (Edited by), Lotfi A. Zadeh (Foreword by), September 1993, pp.168-172.

[7] V. Varshavsky, V. Marakhovsky, I. Levin, and N. Kravchenko, "*Fuzzy Device*", the application number is 2003-190073, filed to Japan's Patent Office, Feb. 6, 2003.