Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI'94) • Las Vegas, NV • Oct. 2-5, 1994

An Autonomous Fuzzy Logic Architecture for Multisensor Data Fusion

R. E. Gibson
D. L. Hall
J. A. Stover
The Pennsylvania State University
Applied Research Laboratory
P.O. Box 30
State College, PA 16804-0030

Abstract—Fuzzy logic techniques have become popular to address various processes for multisensor data fusion. Examples include: (1) fuzzy membership functions for data association, (2) evaluation of alternative hypotheses in multiple hypothesis trackers, (3) fuzzy-logic-based pattern recognition (e.g., for feature-based object identification), and (4) fuzzy inference schemes for sensor resource allocation. These approaches have been individually successful but are limited to only a single subprocess within a data fusion system. At The Pennsylvania State University, Applied Research Laboratory, a general-purpose fuzzy logic architecture has been developed that provides for control of sensing resources, fusion of data for tracking, automatic object recognition, control of system resources and elements, and automated situation assessment. This general architecture has been applied to implement an autonomous vehicle capable of self-direction, obstacle avoidance, and mission completion. The fuzzy logic architecture provides interpretation and fusion of multisensor data (i.e., perception) as well as logic for process control (action). This paper provides an overview of the fuzzy logic architecture and a discussion of its application to data fusion in the context of the Department of Defense (DoD) Joint Directors of Laboratories (JDL) Data Fusion Process Model. A new, robust, fuzzy calculus is introduced. An example is provided by modeling a component of the perception processing of a bat.

I. Introduction

In recent years, multisensor data fusion technology has rapidly evolved. Numerous prototype systems have been developed for Department of Defense (DoD) applications [1] such as ocean surveillance, air-to-air and surface-to-air defense, battlefield intelligence, surveillance and object acquisition, and strategic warning and defense. In addition, data fusion techniques have been applied to non-DoD applications [2] [3]. The DoD Joint Directors of Laboratories (JDL) Data Fusion Group has acted as a "technology gatekeeper," planning and organizing the annual Data Fusion Systems conference held at Johns Hopkins University since 1987, codifying data fusion terminology via a lexicon [4], establishing a process model [5], and creating a taxonomy which relates algorithms and techniques to elements of the process model [6].

The JDL fusion process model, shown in Fig. 1, and the associated algorithm taxonomy provides a basis for understanding how a particular algorithm or technique may be applied to multisensor data fusion. The JDL model specifies four fusion processes:

- Level one processing (object refinement) transforms sensor data into a consistent reference frame, refines and extends estimates of an object's position/kinematics (i.e., tracking), estimates an object's attributes, and refines the estimate of an object's identity. Level one techniques include data association, estimation, pattern recognition, decision level fusion (e.g., Bayesian, Dempster Shafer methods), and approximate reasoning techniques. An object in this context is application specific (i.e., military unit, weapon, object, equipment fault, etc.)

- Level two processing (situation refinement) develops descriptions of current relationships among objects and events in the context of the environment.

- Level three processing (threat refinement) projects the current "situation" into the future and draws inferences about threats, vulnerability, and opportunity for operations.

- Finally, level four processing (process refinement) monitors process performance to provide information for real-time control and long-term improvement.

The JDL process model is (admittedly) a pedagogical artifice but has been useful for communication among data fusion researchers involved in diverse applications. A detailed taxonomy ([5] and [6]) maps specific algorithms to the processing levels and to functions within each processing level.

0-7803-2072-7/94 $4.00 © 1994 IEEE

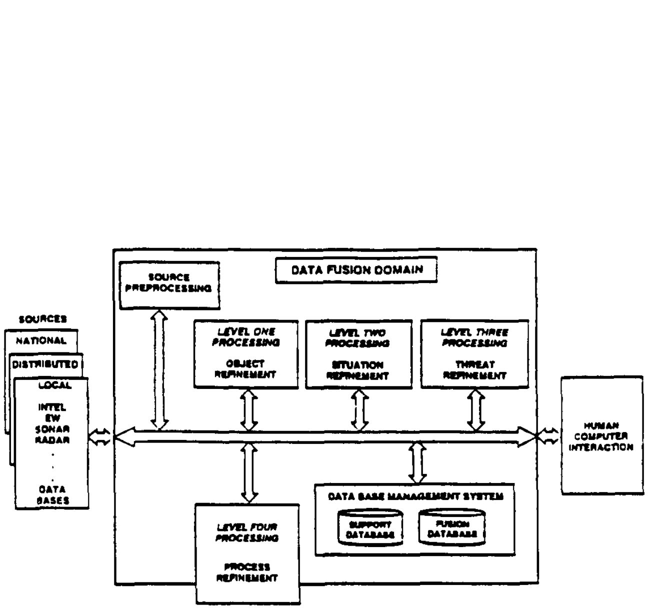


Fig. 1. JDL Data Fusion Process Model.

Many of these algorithms are described in [7] and [8]. The specific techniques applicable to a particular data fusion process depend upon the details of the fusion process, the nature of the sensor data, the types of inferences sought, and system constraints (e.g., computing resources, available a prior data; etc.) [9].

Recently, the application of fuzzy logic techniques to data fusion processes has become popular. Fuzzy logic [10] involves extension of Boolean set theory and Boolean logic (i.e., two-valued logic) to a continuous-valued logic via the concept of membership functions. Membership functions are ad hoc continuous functions defined on the interval [0,1] which may be used to quantify "fuzziness" or imprecise concepts. Thus, fuzzy membership quantifies the extent to which a concept or attribute is inherently imprecise (e.g., the human attribute "tall" or "short" vice the precise measure of height). By contrast, other techniques, such as probability, quantify the extent to which a precise concept (e.g., height) is unknown [11].

Fuzzy logic has been applied to data fusion processing in a number of ways. Son, Kim, Song, and Jhee [12] describe the use of fuzzy logic for detection of known signals. Kewley [13] provides an example of fuzzy logic for identify declaration (i.e., for level one processing) based on Electronic Support Measures (ESM) data. Hall [7] discusses the use of fuzzy logic for expert systems. These, in turn, are applicable to situation assessment (level two processing), threat assessment (level three processing), and process refinement (level four). Finally, several developers have used fuzzy logic for processing control (e.g., [14] and [15]).

At The Pennsylvania State University, Applied Research Laboratory, Stover and Gibson have developed a general-purpose fuzzy logic architecture. The architecture provides for fusion of data for tracking (level one positional fusion), automatic object recognition (level one identity declaration), automated threat assessment (level three processing), and control of system resources (level four processing). The fuzzy logic architecture provides interpretation and fusion of multisensor data (i.e., perception) as well as logic for process control (action). Thus, the fuzzy logic architecture spans all four levels of data fusion processing and provides both interpretive data analysis as well as proactive planning for system control. In this paper, we describe the general architecture and summarize the fuzzy logic calculus.

II. Intelligent Controller Processing Architecture

A diagram of the architectural layout of the controller system is shown in Fig. 2. Sensor data are processed using algorithms such as A/D conversion, filtering, weighting, beamforming, and discrete Fourier transforms to prepare for detection processing. Single-sensor detection processing includes background noise estimation, thresholding, clustering, etc. The output of detection processing is detection reports which contain a set of variables such as range, SNR, azimuth, signal amplitude, etc., that characterize the physical attributes of detected objects. In the Prototype Intelligent Controller (PIC) architecture, these sensor reports are processed to fuse the new information into previously formed representations of the external-world objects. Object properties are divided into two classes, "physical" and "inferred," the former representing measurable extrinsic physical concepts such as range, position, velocity, etc., and the latter representing intrinsic properties that are not directly measurable, but must be inferred (e.g., the inference that an object is a "prey"). Values of physical properties are, in general, computed by adaptive estimators, such as predictor-corrector filters. For example, velocity may not be observed directly but may be estimated by a recursive filter that computes velocity estimates from noisy position measurements over time.

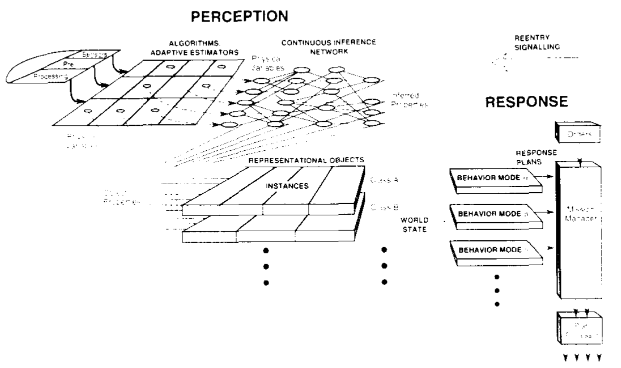


Fig. 2. Prototype Intelligent Controller (PIC) Architecture.

Inferred properties are established in the PIC architecture by a pattern recognition/classification structure called "Continuous Inference Network" (CINET). The CINET is an intelligent multinode fuzzy pattern classifier (Fig. 2). Each node in the network generates a confidence factor for the existence of some property in the sensory data. Node inputs are usually a set of subproperties with varying confidences of existence and varying degrees of significance to the output property. Typical node transfer functions are mathematical models of the "and" and "or" of ordinary language, "and" representing a necessity condition (all input properties must exist with high confidence to produce a high confidence in the output properties' existence) and "or" representing a sufficiency condition (any one input property existing with high confidence can produce a high confidence in the existence of the output property). Thus, each node performs a fuzzy pattern recognition based on fuzzy input properties and application of fuzzy logic connectives (viz. fuzzy "AND," fuzzy "OR," etc.) This is in contrast to Boolean (binary) mathematical logic in which properties are assumed to either exist or not, and each node generates only one of two values, 0 (false) or 1 (true).

The CINET classifier retains the continuity of physical variables and their information content rather than mapping them down to a binary decision. Moreover, the CINET transfer functions allow explicit formulation of fuzzy logic conditions. In addition, the CINET nodes also may use transfer functions (other than those of "and" and "or,") such as a weighted average. For situations in which the relationship between the input subproperties and the output property is not well known, a neural network may be used to represent the node (after its training is completed), provided it maps continuously to the closed unit interval [0,1].

Both the "adaptive estimator" and CINET levels are components of a data fusion process performing level one position/kinematic/attribute fusion as well as higher level inferences related to object identity declaration, situation assessment, and threat assessment. These representations define the perceived world for the system and serve as the basis for generation of behavioral response.

In addition to reactive interpretation of data (viz. automated inferences), the PIC architecture provides proactive responses to guide the system, control sensors, etc. A response is implemented as a collection of independent operations, referred to as behavior modes, each responsible for some class of behavior such as Search or Avoid. Each operation examines representational objects for relevance; e.g., an Avoid operation may look for objects of the class "Obstacle" that have positive confidence factors for the property "Threat." When such conditions exist, an operation will inform the top level Mission Manager that it wants to respond. If allowed to proceed, it will generate and execute a plan to carry out its response. For organizational simplicity, each behavior and the Mission Manager is constructed with the same internal structure. The output of the plan processor is a set of commands to effector subsystems.

III. Fuzzy Logical Operators

The basis for the intelligent fuzzy logic pattern recognition within the PIC architecture involves the definition of fuzzy membership functions and fuzzy logic operators. A brief summary is provided here. The fuzzy pattern recognition processing transforms sensor observations into updates of the representational models. Multiple sensors may be involved to gather information of a particular class. The outputs of these sensors (and associated detectors and estimators) are variables that represent extrinsic physical attributes, such as distance, length, azimuth, SNR, speed, etc., for each object detected. These form the set of physical and dynamical properties of the representational objects. Physical properties of an object are assumed to be precise (i.e., nonfuzzy) once a detection decision has been made, although estimation error variances may be included to quantify the uncertainty of the variable. By contrast, "fuzzy" properties are inferred from the values of physical properties since they cannot be detected directly by sensors. Consider, for example, the declaration that an object should be labeled as a "Prey." Sensors do not output "Prey" directly; however, physical variables may allow the existence of "Prey" to be inferred. In general, "Prey" will be defined in terms of a collection of subproperties, each with its own confidence of existence and degree of significance with respect to establishing the output property. We model this hierarchical structure in terms of fuzzy logical operators or truth functions.

A physical variable is a real- or complex-valued variable derived from the sensor data stream, representing a physical quantity as discussed above. A fuzzy logical operator (truth function, existence function) is a continuous function from a subspace of R^n onto the closed unit interval [0,1]. Of particular interest are truth functions defined on R^1. These effect the inference of the existence of a fuzzy property from a single physical variable; for example, the property "large" from the physical variable "length." (We adopt the convention that a property exists only if the truth value of its defining statement is positive.) Truth functions defined on R^1 transform physical variables to fuzzy properties.

A connective C is a fuzzy logical operator defined on the n-dimensional unit cube $[0,1]^n$ such that:

(1) $C(0) = 0$ (Connectives do not create existence from nothing), and
(2) $C(1) = 1$ (Connectives do not dissipate certainty). (1)

A connective is discriminatory if it is strictly monotonically increasing; i.e.,

$$c_i > c_i^1 \Rightarrow C(c_1, \ldots c_i \ldots c_n) > C(c_1, \ldots c_i^1 \ldots c_n),\ i = 1, \ldots n. \quad (2)$$

We are interested in modeling the "and" and "or" connectives of natural language in a way that preserves the intuitive meanings of these connectives while at the same time meeting the needs of designers of autonomous system controllers. Since autonomous system technology is in its infancy, designers' needs are not yet fully understood. The following is a preliminary list of requirements for "and" that we have found to be useful.

1. The mathematical model, AND, should match the intuitive meaning of the "and" of ordinary language so that ordinary knowledge can be implemented directly. The connective AND models a necessity condition; i.e., the output property, P, exists only if each of a collection $\{Q_1, Q_2, \ldots Q_n\}$ of subproperties exist. The truth value or confidence factor for the existence of P, Cf(P), is computed by

$$CF(P) = AND\ (Cf(Q_1),\ Cf(Q_2), \ldots Cf(Q_n)). \quad (3)$$

2. AND should be discriminatory.

3. The function AND should also have the capability of combining information of varying degrees of significance as well as of varying degrees of existence. This implies the use of n additional weight parameters, one for each input property.

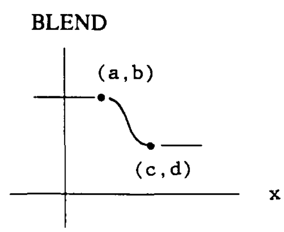
A similar set of requirements can be established for the model OR of "or," differing only in the intuitive meaning; i.e., OR models a sufficiency condition.

In accordance with these specifications, models for AND and OR are families of fuzzy logical operators parameterized by the number of input properties n. While there is an infinite number of functional definitions satisfying the three conditions that could be used to model "and" and "or," the following is a set of functional models that we have used for a number of years in classifier designs apparently having no pathological behavior.

Prior to defining models for AND and OR, the issue of making the initial transition from physical variable to fuzzy property must be addressed. Instead of choosing a linear function for this, we define a utility function, referred to as BLEND, that produces a smooth transition between any two points in the plane, (a,b) and (c,d), by use of a sine curve:

$$\begin{aligned} BLEND\ (a,b,c,d,x) &= b, \quad x \leq a, \\ &= d, \quad x \geq c, \\ &= .5\ [\,d + b + (d-b)\sin(\pi((x-a)/(c-a)-.5))\,], \\ & a < x < c. \end{aligned} \quad (4)$$

BLEND has the graph



For example, we may specify the existence of the fuzzy property "Tall" by

$$Cf\ (Tall) = BLEND\ (66,\ 0,\ 74,\ 1,\ Height). \quad (5)$$

Combinations of BLENDS may be used to produce various truth function shapes such as "bandpass," "bandstop," etc., as application needs dictate.

The functional model used for AND is the product:

$$AND\ (C,\ W,\ n) = [\prod_{i=1}^{n} (1 - w_i + w_i c_i)]^u, \text{ where}$$
$C = (c_1, c_2, \ldots c_n)$, the input confidence vector,
$W = (w_1, w_2, \ldots w_n)$, the weight vector,
n = number of inputs (subproperties),
$u = a + b \exp(-c(\sum w_i - 1))$, and parameters a, b, c being 0.1, 0.9, and 0.3 respectively. (6)

The weights, w_i, control the relative significance of subproperties. A value of 1.0 for w_i specifies that p_i is a required subproperty since, if it does not exist ($c_i = 0$), then it produces a zero multiplier, driving the output to zero; i.e., the output property does not exist, regardless of the existence of the other subproperties. Conversely, if $w_i < 1$, the multiplying factor is positive even if $c_i = 0$; i.e., the nonexistence of p_i does not imply nonexistence of the output property but merely lowers its existence confidence factor by an amount controlled by w_i. Thus, "supportive" properties are specified by using a weight less than 1.0. The exponent u is used to control the tendency of a product of numbers less than one to converge to zero. Alternative versions of u, such as the nth root, were tried but judged to be incapable of providing adequate control under all variations of the parameters.

The functional model used for OR is defined using the Euclidean norm of the weighted input vector:

$$\begin{aligned} OR(C, W, n) &= BLEND(a, b, \sqrt{n}, 1, E), \text{ where} \\ E &= \sqrt{\sum_{i=1}^{n} (w_i c_i)^2} \\ u &= Max\{w_i c_i, i = 1, 2, \ldots n\} \\ a &= 2u - \sqrt{n} \\ b &= 2u - 1, \text{ and} \\ &W \text{ and } C \text{ are defined as for AND.} \end{aligned} \tag{7}$$

The Euclidean norm, E, was chosen to model OR since its value is at least as large as the largest vector component, which corresponds to the intuitive meaning of "or." However, the norm does not map onto the unit interval [0,1] but onto $[0,\sqrt{n}]$. Consequently, BLEND is used to compress the Euclidean norm smoothly to [0,1] with compression beginning at the maximum weighted input confidence factor.

The definitions given above for AND and OR do not satisfy a strict DeMorgan's Law; i.e.:

$$\overline{OR}(C, W, n) \neq AND(\overline{C}, W, n), \text{ where } \overline{c}_i = 1 - c_i \text{ denotes negation.} \tag{8}$$

Nevertheless, using these definitions, a "fuzzy" DeMorgan's law holds in the sense that the terms on each side of the equation are "close" and can, in general, be interchanged in applications. An alternative approach is to force a DeMorgan's law by defining one connective in terms of the other through it. For example, we may define OR by:

$$OR(C, W, n) = \overline{AND}(\overline{C}, W, n). \tag{9}$$

Fig. 3 shows a comparison of the product and Euclidean norm definitions previously given for AND and OR respectively, and the MIN-MAX definitions defined by

$$\begin{aligned} AND\text{-}Z(C) &= MIN(c_1, c_2, \ldots c_n) \\ OR\text{-}Z(C) &= MAX(c_1, c_2, \ldots c_n). \end{aligned} \tag{10}$$

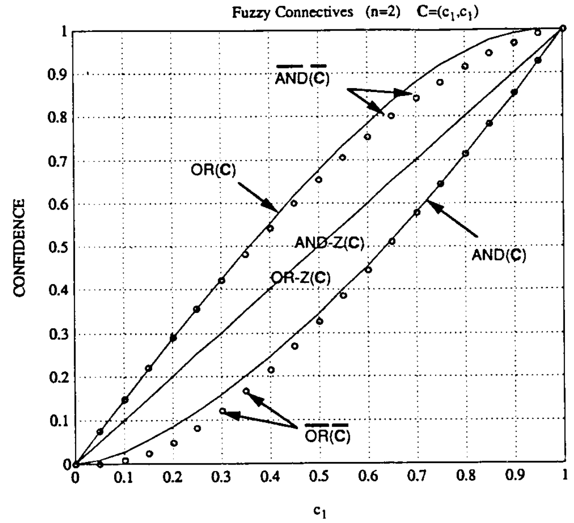


Fig. 3. Comparison of Formulations of AND and OR along the cut $c_1 = c_2$.

The circular points plot the connective counterparts of AND and OR defined through DeMorgan's Law. These points can also be interpreted as showing how far the AND and OR functions deviate from DeMorgan's Law (for this case). It may appear that the best approach would be to define OR from the AND product definition via DeMorgan's law (i.e., forcing satisfaction of DeMorgan's law by definition). However, there is another factor that makes an insistence on a strict DeMorgan's law questionable for autonomous system applications. This is discussed next.

IV. Negation

The meaning of the word "not" as used in ordinary language is more complex than provided for by the logical definition

$$CF(\text{not } P) = 1 - CF(P). \tag{11}$$

A statement such as "That is not a rat" can be accepted on the basis of observation of only one property, say "It has wings." Use of the formula would require that the complete set of characterizing properties of "rat" be identified to generate a confidence factor for the property "rat" for an object being observed. For autonomous systems, this is generally not possible because at any given time, information about an external-world object will be incomplete. Furthermore, the validity of the formula at any time depends on the degree of completeness of the information about P. The confidence in P may be low because insufficient sensory information is available. In low light or from a distance, it may be difficult to confidently identify the property "wings" on a rat-like object sitting on the ground. For example, if the pattern recognition processor generates a confidence factor of 0.01 for "wings" for an object, this does not imply that one can conclude the object does not have wings with a confidence of 0.99. Based on the sensory information available, the confidence that the object does not have wings may properly also be low. The key requirement here is that autonomous systems must be able to operate and make decisions with incomplete information. If the knowledge being programmed into the classifying structure involves the use of "not," then some estimation of the degree of completeness of information with respect to a property P needs to be factored into the formula above. We introduce the following model for NOT, illustrated in Fig. 4.

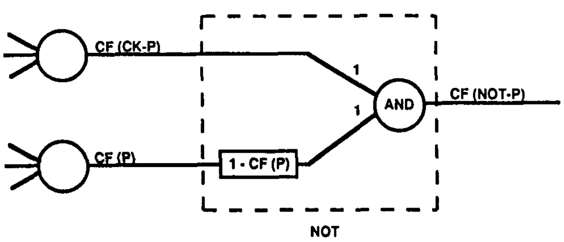


Fig. 4. Conceptual Model of NOT.

The dashed block defines NOT, which can be explained as "The confidence factor for 'NOT-P' is the complement of the confidence factor for P and complete knowledge about P exists." The property "complete knowledge about P," (CK- P) is a fuzzy concept; i.e., it has infinite degrees of truth.

Estimation of the current degree of completeness of information with respect to a property is useful to autonomous system processing, not only for computing negation, but also as a measure of classification reliability. No general approach to computing CK-P is known, since it is property-dependent. However, if the physical variables that provide input to the classifier nodes are outputs of adaptive estimators such as Kalman filters, then the degree of convergence of the filters can be used to compute confidence factors for CK-P. If adaptive filters are used to estimate dynamics of moving objects, then a confidence factor CK-Dynamics can be generated from estimates of the degree of convergence of the filters. For example, if we define "Mobile" by the statement "An object is Mobile if it is Not-stationary and has a Believable-speed-estimate," the property "Believable-speed-estimate" can be identified with CK-Dynamics.

Since for autonomous system applications (operation with incomplete knowledge), negation must be modeled as a property-dependent function, a strict DeMorgan's law is, in general, unrealizable. Consequently, the only motivation for modeling one of AND or OR through DeMorgan's law operating on the other is for reasons of simplicity or esthetics. The property-dependence of negation seems to constitute a barrier to the development of a (quasi) Boolean algebra for fuzzy set theory applicable to autonomous systems.

V. An Example

Fuzzy logical operators can be used in a variety of data fusion modules, such as detection decisions, fuzzy correlation, multilevel (hierarchical-property-based) classification, and introspective analysis (e.g., confusion modeling, [16]).

An example is provided of a relatively simple, multilevel classifier roughly based on a bat's sonar processing described in [17]. The knowledge underlying the definition of "Prey" is expressed by the statements:

> An object is (has the property) "Prey" if it is "Prey-size" and "Moving" and has "Insect-like-buzz."
>
> An object is "Moving" if it is "Not-stationary" and has a "Believable-speed-estimate"; and
>
> An object has "Insect-buzz" if it has"Doppler-flutter" and "Insect-like-period."

The logical form of the statements defining "Prey" generates the processing structure directly. Fig. 5 shows the resulting CINET segment. Blend functions map the physical variables SIZE, DS (Doppler shift), PERIOD, and SPEED into confidence factors for subproperties that are involved in the definition of "Prey." The property "Believability" for various dynamical properties is identified with CK-Dynamics.

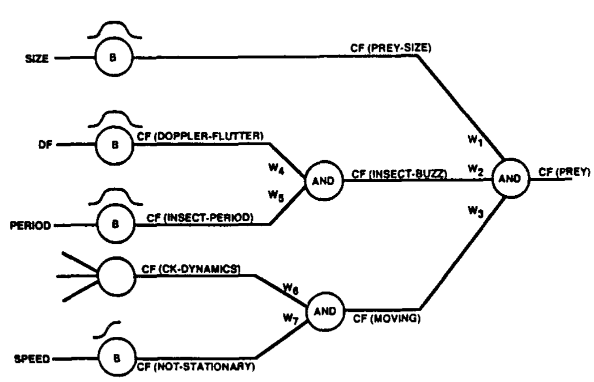


Fig. 5. CINET Classifier for "Prey."

The example shows that the transformation from knowledge representing relationships among properties and subproperties to the processing structure that performs pattern recognition for the autonomous system is almost trivial. Parameters such as weights that designate relative significance among properties and BLEND parameters that map from physical variable values to confidence factors are set to reflect the intricacies of the relationships that are understood but not contained explicitly in the defining statements. These parameters may actually be variables that are computed by on-board functions as appropriate to dynamically changing situations. This provides a capability for interpretation within context or, in other words, a capability for autonomous adaptation of interpretation of sensory data appropriate to the interpretation of the current situation from the top level (via the "Reentry Signalling" lines shown in Fig. 2).

The resulting processing structure has a functional power that may be obscured by its ease of generation and pictorial simplicity. Equation (12) below is a partial expansion of the fuzzy logical operator that generates the confidence factor for the property "Prey" from the four physical variables SIZE, DS, PERIOD, and SPEED. It is a highly nonlinear function with 23 adaptation parameters that may be dynamically

$$
\begin{aligned}
CF(PREY) &= F(SIZE, DS, PERIOD, SPEED) \\
&= \Big\{ \Big[1 - w_1 + w_1 \{.5(b_4 + b_2 + (b_4 - b_2)\sin(\pi(\frac{SIZE - b_1}{b_3 - b_1} - .5)))\}\Big] \\
&\quad * \Big[1 - w_2 + w_2\Big\{ [1 - w_4 + w_4\{.5(b_8 + b_6 + (b_8 - b_6)\sin(\pi(\frac{DS - b_5}{b_1 - b_5} - .5)))\}] \\
&\qquad\qquad * [1 - w_5 + w_5\{.5(b_{12} + b_{10} + (b_{12} - b_{10})\sin(\pi(\frac{PERIOD - b_{11}}{b_{11} - b_9} - .5)))\}] \\
&\qquad\qquad \Big\}^{(.1 + .9\exp(-.3(w_4 + w_5 - 1)))} \Big] \\
&\quad * \Big[1 - w_3 + w_3\Big\{ [1 - w_6 + w_6 * CF(CK-DYNAMICS] \\
&\qquad\qquad * [1 - w_7 + w_7\{.5(b_{16} + b_{14} + (b_{16} - b_{14})\sin(\pi(\frac{SPEED - b_{15}}{b_{15} - b_{13}} - .5)))\}] \\
&\qquad\qquad \Big\}^{(.1 + .9\exp(-.3(w_6 + w_7 - 1)))} \Big] \Big\}^{(.1 + .9\exp(-.3(w_1 + w_2 + w_3 - 1)))} ,
\end{aligned}
\tag{12}
$$

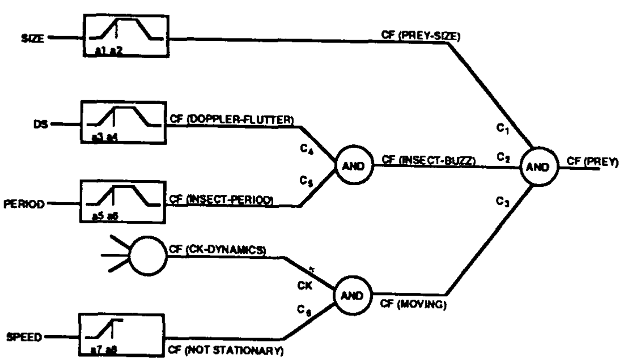
where $b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{15}, b_{16}, w_1, w_2, w_3, w_4, w_5, w_6$, and w_7 are adaptation parameters.

(12)

adjusted as discussed above. The number of parameters is a measure of the degree to which the processing structure of Fig. 5 can be adapted to specifics of an application. If the designer were forced to work with just the functional form of F(SIZE, DS, PERIOD, SPEED), it would be a difficult task to determine values for the 23 parameters that would tune the function to the application. However, since the network is a direct representation of the knowledge, parameter settings are intuitive, at least to a first approximation. Fine tuning can be accomplished by observing the classifier performance to a variety of inputs which is, in effect, fine tuning the designer's knowledge.

The "CK-Dynamics" property would be, in general, the output of a multistaged CINET segment; if the complete expansion of that property were included in the equation below, it would increase in size considerably. In practice, the designer never needs to be concerned with expanded forms of fuzzy logical operators but can think and design in terms of property relationships.

For comparison, Fig. 6 shows an implementation of the same classifier using the Min-Max definitions of fuzzy "and" and "or." Linear truth functions are used rather than BLENDS to do the initial mapping from physical variable to confidence factor. The resulting fuzzy logical operator is the minimum of five terms, four of them linear functions of the input variables SIZE, DS, PERIOD, and SPEED. It can be seen that the minimum of these terms is completely specifying the output confidence factor for "Prey," producing the same value on output for any values of the other terms so long as they stay above the minimum; i.e., the operator is nondiscriminatory. A bat operating with this classifier could not distinguish between two insects, one of which had, say 0.1 confidence in each of the subproperties of "Prey" and the other having 0.1 confidence in the "Moving" subproperty but 0.9 confidence in the "Prey-size" and "Insect-buzz" subproperties. Such inability to discriminate is considered unacceptable for autonomous system perception processing.



$$CF(PREY) = AND(C_1, C_2, C_3) = MIN(C_1, C_2, C_3)$$
$$= MIN(C_1, MIN(C_4, C_5), MIN(CK, C_6))$$
$$= MIN(C_1, C_4, C_5, CK, C_6)$$
$$= MIN\left(CK, \frac{SIZE - a_1}{a_2 - a_1}, \frac{DS - a_3}{a_4 - a_3}, \frac{PERIOD - a_5}{a_6 - a_5}, \frac{SPEED - a_7}{a_8 - a_7}\right),$$

where $a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8$ are truth function parameters

Fig. 6. Implementation with Min-Max Logic.

V. SUMMARY

In this paper, we have described a general-purpose fuzzy logic architecture that provides the capability for fusion of multisensor data to achieve high-level (i.e., abstract) inferences about objects and situations. A new fuzzy logic calculus is introduced with connectives (AND, OR) that mimic the intuitive meaning of those of natural language. In addition, a new operator is introduced for logical negation that accounts for both the confidence in a property as well as the extent to which the property is observable or known (in a current environment). Finally, we have introduced a continuous transformation that maps physical parameters into fuzzy membership functions (the BLEND function). The architecture introduced here provides for both data fusion as well as proactive control based on the dynamic data fusion inferences. This architecture and fuzzy logic calculus have been successfully used for control of an autonomous vehicle and have application in many other areas, such as process controllers or advisory systems.

REFERENCES

[1] D. L. Hall, R. J. Linn, and J. Llinas, "A Survey of Data Fusion Systems," Proceedings of the SPIE Conference on Data Structure and Target Classification, pp. 13-29, April 1991.

[2] R. J. Hansen, D. L. Hall, and S. K. Kurtz, "A New Approach to the Challenge of Machinery Prognostics," accepted for publication in The Proceedings of the International Gas Turbine Institute Conference, 1994.

[3] R. C. Luo and Ming-Hsiumm Lin," Robot Multisensor Fusion and Integration: Optimum Estimation of Fused Sensor Data," Proceedings of the 1988 IEEE International Conference on Robots and Automation, Philadelphia, PA, April 1988, pp. 24-29.

[4] "Lexicon for Multisensor Data Fusion," report published by the Joint Directors of Laboratories Data Fusion Group, available from NAWC, Warminster, PA, 1987.

[5] O. Kessler, K. Askin, N. Beck, J. Lynch, F. White, D. Buede, D. Hall, and J. Llinas, "Functional Description of the Data Fusion Process," report for the Office of Naval Technology, published by NAWC, Warminster, PA, November 1991.

[6] D. Hall, O. Kessler, J. Llinas, and F. White, Models and Techniques for Multisensor Data Fusion, a monograph prepared for the ONT (in press), October 1994.

[7] D. L. Hall, Mathematical Techniques in Multisensor Data Fusion, Artech House, Inc., 1992.

[8] E. Waltz and J. Llinas, Multisensor Data Fusion, Artech House, Inc. 1990.

[9] D. Hall and R. J. Linn, "Algorithms Selection for Data Fusion Systems," Proceedings of the 1987 Tri-Service Data Fusion Symoposium, Johns Hopkins APL, 1: 100-110, June 1987.

[10] B. R. Gaines, "Foundations of Fuzzy Reasoning," Int. J. Man-Machine Studies, vol. 8, pp. 623-668, 1976.

[11] B. R. Gianes, "Fuzzy and Probability Uncertainty Logics," Inform. Contr., vol. 38, no. 2, pp. 154-169, 1978.

[12] J. C. Son, S. Y. Kim, I. Song, and Y. K. Jhee, "Statistical Characteristics and Performance Analysis of Locally Optimum Fuzzy Detector for Known Signals," Signal Processing, 33, pp. 259-268, 1993.

[13] D. J. Kewley, "Notes on the Use of Dempster-Shafer and Fuzzy Reasoning to Fuse Identity Attribute Data," Technical Memorandum SRL-0094-TM, Dept. of Defense, Defense Science and Technology Organization, Commonwealth of Australia, Available from DTIC No. AD-A267 426.

[14] G. P. Bradford, "A Fuzzy Logic Controller System," U.S. Army Missile Command, Redstone Arsenal, AL, Technical Report RD-GC-92-26 (DTIC HD-A259 723), November 1992.

[15] S. C. Lee, "Intelligent Control Based on Fuzzy Logic and Neural Net Theory," in Proceedings of the Second Internatural Conference on Fuzzy Logic and Neural Networks, (T. Yamakaua, ed.), Iizuka, Japan, pp. 759-764, 1990.

[16] J. A. Stover and R. E. Gibson, "Modeling Confusion for Autonomous Systems," SPIE Vol. 1710, Science of Artificial Neural Networks, (1992), pp. 547-555.

[17] N. Suga, "Biosonar and Neural Computation in Bats," Scientific American, June 1990.

This research was supported by the Office of Naval Research, Department of the Navy, under Prime Contract N00039-92-C-0100. Approved for publication release; distribution is unlimited.