

Cleveland State University
EngagedScholarship@CSU



Electrical Engineering & Computer Science Faculty
Publications

Electrical Engineering & Computer Science
Department

5-1995

Fuzzy Logic for Digital Phase-Locked Loop Filter Design

Daniel J. Simon

Cleveland State University, d.j.simon@csuohio.edu

Hossny El-Sherief

TRW System Integration Group

Follow this and additional works at: https://engagedscholarship.csuohio.edu/enece_facpub

 Part of the [Electrical and Computer Engineering Commons](#)

How does access to this work benefit you? Let us know!

Publisher's Statement

© 1995 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

Original Citation

Simon, D. & El-Sherief, H. (1995). Fuzzy logic for digital phase-locked loop filter design. *Fuzzy Systems, IEEE Transactions on* 3(2), 211-218, doi: 10.1109/91.388174.

Repository Citation

Simon, Daniel J. and El-Sherief, Hossny, "Fuzzy Logic for Digital Phase-Locked Loop Filter Design" (1995). *Electrical Engineering & Computer Science Faculty Publications*. 152.

https://engagedscholarship.csuohio.edu/enece_facpub/152

This Article is brought to you for free and open access by the Electrical Engineering & Computer Science Department at EngagedScholarship@CSU. It has been accepted for inclusion in Electrical Engineering & Computer Science Faculty Publications by an authorized administrator of EngagedScholarship@CSU. For more information, please contact library.es@csuohio.edu.

Fuzzy Logic for Digital Phase-Locked Loop Filter Design

Dan Simon, *Member, IEEE*, and Hossny El-Sherief, *Senior Member, IEEE*

Abstract—The problem of robust phase-locked loop design has attracted attention for many years, particularly since the advent of the global positioning system. This paper proposes and demonstrates the use of a fuzzy PLL to estimate the time-varying phase of a sinusoidal signal. It is shown via simulation results that fuzzy PLL's offer performance comparable to analytically derived PLL's (e.g. Kalman filters and H_∞ estimators) when the phase exhibits high dynamics and high noise. The fuzzy PLL rules are optimized using a gradient descent method and a genetic algorithm.

I. INTRODUCTION

FUZZY systems have recently been used extensively and successfully in control problems, but relatively few applications have been made to estimation problems. This is somewhat surprising in view of the close relationship between the fields of control and estimation. This paper discusses and demonstrates the application of fuzzy estimation to phase-locked loop (PLL) design. A PLL is used to track (estimate) the phase of a sinusoidal signal which may have a time-varying frequency. Many analytic methods have been used for PLL design [12], [13], but nonanalytic (decision-making) approaches have not yet been applied to PLL's.

Automatic control methods based on artificial intelligence approaches (e.g., fuzzy systems, neural networks, and genetic algorithms) have emerged as an alternate paradigm to analytic control theory [15]. Fuzzy systems especially have gained prominence due to the surprising ease with which fuzzy controllers can be designed. In fact, it is the greatest advantage of fuzzy controllers (their simplicity) which has perhaps been the primary cause of their initial slow acceptance among the control community. Good overviews of fuzzy logic can be found in [3], [10], and [26].

PLL design is of fundamental importance in the design of global positioning system (GPS) receivers, especially when the trajectory of those receivers exhibit high dynamics in a noisy environment. GPS is a satellite-based navigation system which provides position and velocity information to users [7]. GPS receivers determine user position by tracking a known binary pseudo-random (PR) code transmitted by the GPS satellites. After the PR code is removed from the signal, the receiver has access to the sinusoidal carrier. Since the sinusoid is transmitted at a known frequency, the frequency which the

receiver tracks can be used to compute the doppler between the user and the satellite. The satellite orbit is known fairly accurately, so the doppler frequency can be used to obtain the user's velocity. A GPS receiver can therefore be used as a navigation instrument in place of more expensive and complex inertial instruments. Because of its continuous global coverage and the passive nature of the receiver, GPS is being used in a wide variety of aerospace applications [4], [20].

A GPS receiver's PLL typically updates its phase estimate at a 50 Hz rate. This rate is chosen because an *a priori* unknown binary navigation message is modulated onto the GPS signal at 50 Hz [7]. So a PLL rate slower than 50 Hz would be subject to 180-degree phase jumps at the boundaries of the navigation message bits, while a rate faster than 50 Hz would result in a noisier frequency (hence velocity) estimate. The design of PLL's which can maintain lock on the GPS carrier phase has proven to be a challenging task, particularly if the receiver trajectory is highly dynamic or if the signal is very noisy. If the PLL loses lock on the signal, then the user will not be able to compute relative doppler frequencies and the error of the GPS-derived velocity information will grow without bound. It is therefore desirable to provide robust algorithms for the GPS receiver's PLL. The results presented in this paper establish fuzzy estimation as a viable alternative for PLL design.

II. FUZZY ESTIMATION

Consider a discrete, time-invariant system given by

$$x_{k+1} = f(x_k) + v_k \quad (1)$$

$$z_k = h(x_k) + n_k \quad (2)$$

where x_k is the state vector, z_k is the measurement, and v_k and n_k are noise processes. The problem of finding an estimate \hat{x}_k for x_k based on measurements z_i ($i \leq k$) is known as the *a posteriori* filtering problem. One popular form of a recursive estimator is the predictor/corrector, which is given by

$$\hat{x}_k = \hat{f}(\hat{x}_{k-1}) + g(z_k, \hat{x}_{k-1}) \quad (3)$$

where $\hat{f}(\cdot)$ is an estimate of the function which maps the state from one time step to the next, and $g(\cdot)$ is the correction function. Often the process model $f(\cdot)$ is already known or can be found using system identification methods. In that case, only the correction mapping $g(\cdot)$ needs to be determined. Various analytic methods have been used for obtaining the correction mapping [1], [17], [24], [25]. Alternatively, the correction mapping could be implemented as a fuzzy function.

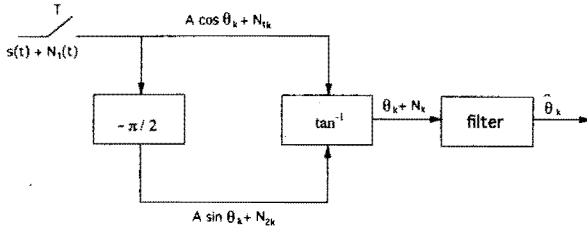


Fig. 1. Phase-locked loop architecture.

Recent applications of fuzzy estimators to target tracking have been reported in [16] and [22].

Typically, fuzzy logic has been applied to systems which are too complex to easily lend themselves to more mathematically rigorous control methods. But some systems are difficult to control even though an analytical system model is known. At times fuzzy controllers are used instead of more conventional controllers because the fuzzy kind are easier to design, cheaper to produce, and more robust (i.e., less susceptible to system changes or noise) [15], [18].

Similarly, some systems behave in a way that makes it difficult to accurately estimate their states, even though the mathematical system model is known. One source of difficulty is the conflicting requirements of tracking high frequency changes in the state while at the same time rejecting noise. The remainder of this section shows how a fuzzy estimator can be used to estimate the phase of a time-varying signal.

A. Phase Estimation

Consider the problem of tracking a sinusoidal signal with an unknown, time-varying phase

$$s(t) = A \cos \theta(t). \quad (4)$$

This signal is corrupted by noise. The device used to track the phase is called a phase-locked loop. The PLL architecture considered in this paper is shown in Fig. 1. Note from Fig. 1 that the output of the arctan phase discriminator is modulo 2π . That is, the phase discriminator does not know the difference between θ radians and $\theta + 2\pi$ radians. If the phase estimation error suddenly goes from zero to some multiple of 2π , it is said that a cycle slip has occurred. So it is more important in a PLL to prevent cycle slips than it is to maintain a small phase error. If the PLL maintains lock on the phase, the PLL contribution to a GPS receiver's velocity error is small compared to other sources of velocity error [19], [20]. For instance, the velocity error due to typical PLL tracking errors may be on the order of 0.01 ft./sec. But the velocity error due to all other sources may be on the order of 0.10 ft./sec. If a cycle slip occurs, then the velocity error due to the PLL tracking error momentarily jumps to 0.90 ft./sec. So undetected cycle slips can be catastrophic. In some cases, the noise is so high or the phase dynamics are so severe that the estimation error begins growing without bound. In this case it is said that loss of lock has occurred, and the user loses all velocity information from the GPS receiver. So for a GPS receiver, it is primarily loss of lock and secondarily cycle slips which are of greatest concern (rather than phase error).

TABLE I
FUZZY RULE BASE FOR PLL ESTIMATION. NL = NEGATIVE LARGE, NM = NEGATIVE MEDIUM, NS = NEGATIVE SMALL, Z = ZERO, PS = POSITIVE SMALL, PM = POSITIVE MEDIUM, PL = POSITIVE LARGE

		input 1						
		NL	NM	NS	Z	PS	PM	PL
input 2	NL	NL	NL	NM	NS	Z	PS	PM
	NM	NL	NL	NM	NS	Z	PS	PM
	NS	NL	NM	NS	Z	PS	PM	PL
	Z	NL	NM	NS	Z	PS	PM	PL
	PS	NL	NM	NS	Z	PS	PM	PL
	PM	NM	NS	Z	PS	PM	PL	PL
	PL	NM	NS	Z	PS	PM	PL	PL

The fuzzy estimator structure used to obtain a phase estimate is given by

$$\hat{\theta}_k^- = \hat{\theta}_{k-1}^+ + T\hat{\omega}_{k-1} \quad (5)$$

$$\hat{\theta}_k^+ = g(z_k, \hat{\theta}_k^-) \quad (6)$$

where T is the update period of the phase estimate (typically 0.02 seconds) and z_k is the noisy measurement of the phase (modulo 2π). (The determination of the frequency estimate $\hat{\omega}$ is discussed in Section II-B.) The fuzzy correction mapping $g(\cdot)$ has two inputs

$$(\text{input } 1)_k = z_k - \hat{\theta}_k^- \quad (7)$$

$$(\text{input } 2)_k = (\text{input } 1)_k - (\text{input } 1)_{k-1}. \quad (8)$$

The output of the correction mapping is a fuzzy variable which is determined by correlation-product inference. The fuzzy rule base for the mapping $g(\cdot)$ was chosen as shown in Table I. (Note that there is some redundancy in the rules, i.e., the same control surface could be generated with fewer rules since rows 1-2 of the table are identical, rows 3-5 are identical, and rows 6-7 are identical. But the number of rules was kept at 49 to provide greater flexibility for the optimization discussed in Section III.) Triangular input and output membership functions were chosen as shown in Fig. 2.

The initial rule base and membership function shapes were constructed on the imprecise basis of experience, and trial and error. An appropriate initial knowledge base is critical, because without an initial knowledge, we cannot proceed any further with any optimization schemes. In spite of its importance, the generation of initial knowledge remains as a difficult and ill-defined task in the construction of fuzzy logic systems.

In general, we denote the centroid and half-width of the i th fuzzy membership function of the j th input by c_{ij} and b_{ij} . So the degree of membership of a crisp input x in the i th category of the j th input is given by

$$f_{ij}(x) = \begin{cases} 1 - |x - c_{ij}|/b_{ij} & |x - c_{ij}| \leq b_{ij}/2 \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

The fuzzy output is mapped into a crisp numerical value using centroid defuzzification [10].

$$g(z_k, \hat{\theta}_k^-) = \frac{\sum_{j=1}^n m(y_j) y_j J_j}{\sum_{j=1}^n m(y_j) J_j} \quad (10)$$

where y_j and J_j are the centroid and area of the j th output fuzzy membership function and n is the number of fuzzy

output sets. The fuzzy output function $m(y)$ is computed as

$$m(y) = \text{fuzzy output function} = \sum_{i,k} m_{ik}(y) \quad (11)$$

$$m_{ik}(y) = \text{consequent fuzzy output function when} \\ (\text{input 1} \in \text{class } i) \text{ and } (\text{input 2} \in \text{class } k) \\ = w_{ik} m_{oik}(y) \quad (12)$$

$$w_{ik} = \text{activation level of consequent when} \\ (\text{input 1} \in \text{class } i) \text{ and} \\ (\text{input 2} \in \text{class } k) \\ = \min [f_{i1}(\text{input 1}), f_{k2}(\text{input 2})] \quad (13)$$

$$m_{oik}(y) = \text{fuzzy function of consequent which is activated} \\ \text{when } (\text{input 1} \in \text{class } i) \text{ and} \\ (\text{input 2} \in \text{class } k). \quad (14)$$

B. Frequency Estimation

One of the inputs to the fuzzy estimator discussed above is the frequency estimate $\hat{\omega}$. This estimate must be computed from the phase estimates using numerical differentiation, which is in itself a challenging task. We will assume that we have the current and the past three phase estimates available. With this in mind, we use the method of undetermined coefficients [2] to obtain the following expression for the frequency.

$$\omega(t) = \left[-\theta(t-3\tau) + \frac{3}{2}\theta(t-2\tau) - 3\theta(t-\tau) + \frac{11}{6}\theta(t) \right] / \tau - \frac{193\tau^3}{72}\theta^{(4)}(\zeta) \quad (15)$$

where τ is a time step to be determined later, and ζ is an unknown constant in $[t-3\tau, t]$. It is our objective in the remainder of this section to determine an appropriate time step τ . Denoting the error in the phase estimate as $\tilde{\theta}$, we obtain

$$\omega(t) = \left[-\hat{\theta}(t-3\tau) + \frac{3}{2}\hat{\theta}(t-2\tau) - 3\hat{\theta}(t-\tau) + \frac{11}{6}\hat{\theta}(t) \right] / \tau \\ + \left[-\tilde{\theta}(t-3\tau) + \frac{3}{2}\tilde{\theta}(t-2\tau) - 3\tilde{\theta}(t-\tau) + \frac{11}{6}\tilde{\theta}(t) \right] / \tau - \frac{193\tau^3}{72}\theta^{(4)}(\zeta). \quad (16)$$

So if we estimate ω as

$$\hat{\omega}(t) = [-\hat{\theta}(t-3\tau) + \frac{3}{2}\hat{\theta}(t-2\tau) - 3\hat{\theta}(t-\tau) + \frac{11}{6}\hat{\theta}(t)] / \tau \quad (17)$$

then we obtain the following expression for the frequency estimation error:

$$\hat{\omega}(t) = \frac{-193\tau^3}{72}\theta^{(4)}(\zeta) + \left[-\tilde{\theta}(t-3\tau) + \frac{3}{2}\tilde{\theta}(t-2\tau) - 3\tilde{\theta}(t-\tau) + \frac{11}{6}\tilde{\theta}(t) \right] / \tau. \quad (18)$$

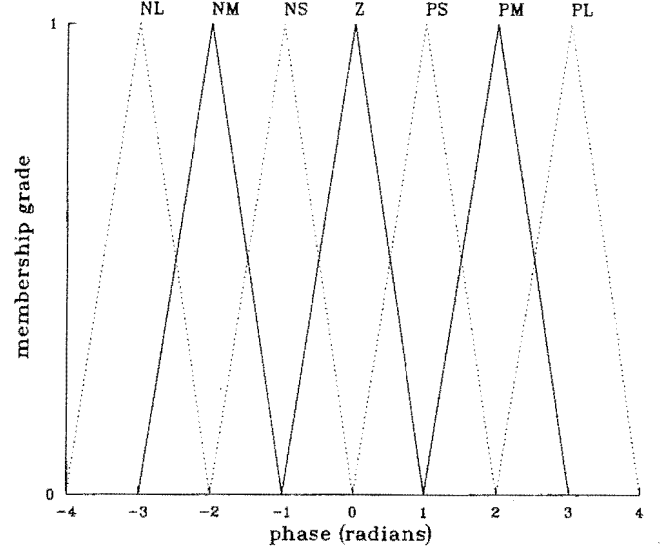


Fig. 2. Nominal triangular membership functions.

Now if we treat the time functions in the above equation as random processes and make the simplifying assumption that $\theta^{(4)}(t)$ and $\tilde{\theta}(t)$ are independent, we obtain the following expression for the variance of the frequency estimation error

$$E(\hat{\omega}^2) = \left(\frac{193\tau^3}{72} \right)^2 E([\theta^{(4)}]^2) + \frac{530}{36\tau^2} E(\tilde{\theta}^2). \quad (19)$$

Assume that we have a one-sigma phase estimation error on the order of a half-radian, and that the standard deviation of the fourth derivative of the satellite-to-receiver range is 100 g/sec² (see Section IV for the relationship between range derivatives and phase derivatives). Assuming that $\hat{\theta}$ is obtained at a 50 Hz rate (see Section I), τ must be a multiple of 0.02 seconds. Then we obtain the following numbers for the standard deviation of the frequency estimation error

$$\sigma_{\hat{\omega}} = \begin{cases} 49 \text{ rad/sec} & (\tau = 0.02 \text{ sec}) \\ 30 \text{ rad/sec} & (\tau = 0.04 \text{ sec}) \\ 60 \text{ rad/sec} & (\tau = 0.06 \text{ sec}). \end{cases} \quad (20)$$

We see that $\tau = 0.04$ seconds should be used in (17) to estimate ω . This finding is critical to the success of the fuzzy PLL, because large errors in $\hat{\omega}$ result in cycle slips and loss of lock.

III. OPTIMIZATION OF FUZZY MEMBERSHIP FUNCTIONS

The set of membership functions shown in Fig. 2 have been defined on the basis of heuristics. Perhaps other membership functions would result in better performance by the fuzzy PLL. In this section, two methods are used to optimize the fuzzy membership functions: gradient descent and a genetic algorithm.

A. Gradient Descent

If the fuzzy membership functions are triangular, gradient descent can be used to optimize the centroids and the widths of the input membership functions, and the centroids of the

output membership functions [5]. Consider an error function given by

$$E = \frac{1}{2N} \sum_{q=1}^n E_q^2$$

$$E_q \equiv \hat{\theta}_q - \theta_q \quad (21)$$

where N is the number of training samples. We can optimize E by using the partial derivatives of E with respect to: a) the centroids of the input fuzzy membership functions; b) the half-width of the input fuzzy membership functions; and c) the centroids of the output fuzzy membership functions.

Input Fuzzy Membership Function Centroids: Using the relationships of (9) and following, we obtain

$$\frac{\partial E}{\partial c_{ij}} = \frac{1}{N} \sum_{q=1}^n E_q \frac{\partial \hat{\theta}_q}{\partial c_{ij}} \quad (22)$$

$$\frac{\partial \hat{\theta}_q}{\partial c_{ij}} = \sum_{p=1}^n \frac{\partial \hat{\theta}_q}{\partial m_p} \frac{\partial m_p}{\partial c_{ij}} \quad [m_p \equiv m(y_p)] \quad (23)$$

$$\frac{\partial \hat{\theta}_q}{\partial m_j} = \frac{J_j(y_j - \hat{\theta}_q)}{\sum_{i=1}^n m_i J_i} \quad (24)$$

$$\frac{\partial m_p}{\partial c_{ij}} = \sum_{k,l} r_{klp} \frac{\partial w_{kl}}{\partial c_{ij}} \quad (25)$$

where r_{klp} and $\partial w_{kl}/\partial c_{ij}$ are given as follows

$$r_{klp} = \begin{cases} 1 & \text{if } [(\text{input } 1) \in \text{class } k \text{ and} \\ & (\text{input } 2) \in \text{class } l] \Rightarrow \\ & (\text{output} \in \text{class } p) \\ 0 & \text{otherwise.} \end{cases} \quad (26)$$

$$\frac{\partial w_{kl}}{\partial c_{ij}} = \begin{cases} \partial f_{k1}/\partial c_{ij} & \text{if } f_{k1}(\text{input } 1) \\ & \leq f_{l2}(\text{input } 2) \\ \partial f_{l2}/\partial c_{ij} & \text{otherwise.} \end{cases} \quad (27)$$

The partials of the membership grades $f(\cdot)$ with respect to the input centroids are

$$\frac{\partial f_{k1}(\text{input } 1)}{\partial c_{i2}} = 0 \quad (28)$$

$$\frac{\partial f_{l2}(\text{input } 2)}{\partial c_{i1}} = 0 \quad (29)$$

$$\frac{\partial f_{k1}(\text{input } 1)}{\partial c_{i1}} = \begin{cases} 2 \text{sign}[(\text{input } 1) - c_{i1}]/b_{i1} & (i = k) \\ 0 & (i \neq k) \end{cases} \quad (30)$$

$$\frac{\partial f_{l2}(\text{input } 2)}{\partial c_{i2}} = \begin{cases} 2 \text{sign}[(\text{input } 2) - c_{i2}]/b_{i2} & (i = l) \\ 0 & (i \neq l). \end{cases} \quad (31)$$

Input Fuzzy Membership Function Half-Widths: Again using (9) and following, it can be shown that

$$\frac{\partial E}{\partial b_{ij}} = \frac{1}{N} \sum_{q=1}^n E_q \frac{\partial \hat{\theta}_q}{\partial b_{ij}} \quad (32)$$

$$\frac{\partial \hat{\theta}_q}{\partial b_{ij}} = \sum_{p=1}^n \frac{\partial \hat{\theta}_q}{\partial m_p} \frac{\partial m_p}{\partial b_{ij}} \quad [m_p \equiv m(y_p)] \quad (33)$$

$$\frac{\partial \hat{\theta}_q}{\partial m_j} = \frac{J_j(y_j - \hat{\theta}_q)}{\sum_{i=1}^n m_i J_i} \quad (34)$$

$$\frac{\partial m_p}{\partial b_{ij}} = \sum_{k,l} r_{klp} \frac{\partial w_{kl}}{\partial b_{ij}} \quad (35)$$

where r_{klp} is given in (26) and $\partial w_{kl}/\partial b_{ij}$ are given as follows

$$\frac{\partial w_{kl}}{\partial b_{ij}} = \begin{cases} \partial f_{k1}/\partial b_{ij} & \text{if } f_{k1}(\text{input } 1) \leq f_{l2}(\text{input } 2) \\ \partial f_{l2}/\partial b_{ij} & \text{otherwise.} \end{cases} \quad (36)$$

The partials of the membership grades with respect to the half-widths of the input fuzzy membership functions are given as

$$\frac{\partial f_{k1}(\text{input } 1)}{\partial b_{i2}} = 0 \quad (37)$$

$$\frac{\partial f_{l2}(\text{input } 2)}{\partial b_{i1}} = 0 \quad (38)$$

$$\frac{\partial f_{k1}(\text{input } 1)}{\partial b_{i1}} = \begin{cases} [1 - (\text{input } 1)]/b_{i1} & (i = k) \\ 0 & (i \neq k) \end{cases} \quad (39)$$

$$\frac{\partial f_{l2}(\text{input } 2)}{\partial b_{i2}} = \begin{cases} [1 - (\text{input } 2)]/b_{i2} & (i = l) \\ 0 & (i \neq l). \end{cases} \quad (40)$$

Output Fuzzy Membership Function Centroids: The partials of the objective function E with respect to the centroids of the output fuzzy membership functions are given as

$$\frac{\partial E}{\partial y_j} = \frac{1}{N} \sum_{q=1}^n E_q \frac{\partial \hat{\theta}_q}{\partial y_j} \quad (41)$$

$$\frac{\partial \hat{\theta}_q}{\partial y_j} = \frac{m_j J_j}{\sum_{i=1}^n m_i J_i} \quad (42)$$

The gradient descent rule is then used to update the independent variables from one iteration to the next as follows

$$c_{ij}(k+1) = c_{ij}(k) - \eta_c \frac{\partial E(k)}{\partial c_{ij}} \quad (43)$$

$$b_{ij}(k+1) = b_{ij}(k) - \eta_b \frac{\partial E(k)}{\partial b_{ij}} \quad (44)$$

$$y_j(k+1) = y_j(k) - \eta_y \frac{\partial E(k)}{\partial y_j} \quad (45)$$

where η_c, η_b and η_y are gradient descent step sizes.

The gradient descent learning method is attractive in that it is conceptually straightforward and typically converges quickly. But it also has some notable drawbacks: optimization is limited to a differentiable function like the quadratic given in (21), gradient descent tends to converge to local minima, and practical learning is limited to certain types of fuzzy membership functions [5]. These considerations motivate the use of genetic algorithms (GA's) for the optimization of the fuzzy PLL.

B. Genetic Algorithms

GA's are optimization methods which are motivated by the concept of "survival of the fittest" in biological evolution. More details and references about GA's can be found in [9], [11], and [14]. GA's can be used to optimize a fuzzy PLL as follows. First, a population of fuzzy membership functions is created. Each member of the population is represented as a binary string. For instance, if symmetric trapezoidal membership functions are used as in Fig. 3, then each member of the population should have enough "genetic" information to represent three parameters for each membership function. There are seven membership functions each for the two inputs and the output (see Table I). This makes a total $3 \times (7+7+7) = 63$ parameters. Each parameter is coded using (say) m bits, so each population member is a binary string of $63m$ bits. The binary representation of a parameter is mapped into the analog number P in the parameter range $[P_{\min}, P_{\max}]$. If a linear mapping is appropriate, the mapping can be performed according to

$$P = P_{\min} + \frac{b}{(2^m - 1)}(P_{\max} - P_{\min}) \quad (46)$$

where b is the integer value represented by the m -bit string.

The fitness of each member of the population is evaluated according to some predetermined method. In PLL design, the most important performance criteria are (in order of descending importance): (a) probability of loss of lock; (b) probability of cycle slip; and (c) phase estimation error. So the fitness of a population member can be measured on the basis of a set of Monte-Carlo simulations of the fuzzy PLL.

$$\text{fitness} = k_1 P(\text{loss of lock}) + k_2 P(\text{cycle slip}) + k_3 e_{\text{rms}} \quad (47)$$

where e_{rms} is the average RMS phase estimation error of the fuzzy PLL over a set of Monte-Carlo simulations, and the k_i are user-specified constants. Unlike gradient descent, GA's can minimize probability of cycle slips and probability of loss of lock in addition to minimizing phase estimation error.

Once the fitness of each member of the population is evaluated, the weakest members are killed off. The fittest members reproduce according to two mechanisms: parthenogenesis (cloning) and crossover. In cloning, a member of the population is identically reproduced. This ensures that extremely fit genetic structures remain in the population. In crossover, two members of the population mate, and the genetic structure of the offspring is a combination of the two parents. For instance, if each member has k bits of genetic information, then bits $(1 \rightarrow r)$ of parent A and bits $[(r+1) \rightarrow k]$ of parent B can be copied into the offspring, where the number r is randomly generated.

Finally, there is a small but nonzero probability of mutation in the offspring. Each bit of the offspring has a small probability of being "flipped." Mutation helps reinject information which may have been lost in the current generation. A cycle of fitness evaluation, reproduction, and mutation is referred to as a "generation."

Note that the GA optimizes trapezoidal membership functions rather than triangular membership functions (as with the

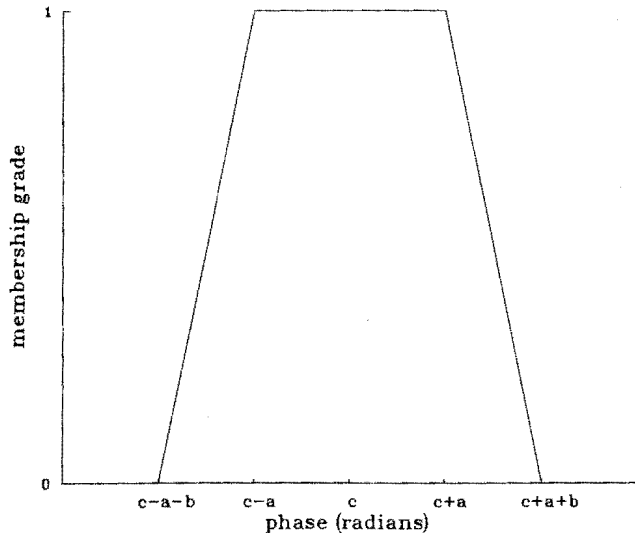


Fig. 3. Trapezoidal membership functions.

gradient descent algorithm). This allows the exploitation of the higher degree of generality inherent in the GA.

IV. SIMULATION STUDY

The fuzzy PLL estimation filter discussed in this paper was simulated for a GPS receiver used for missile navigation. The simulated missile trajectory originated at Vandenberg Air Force Base in California, and ended in the South Pacific. The behavior of the PLL was investigated by examining its ability to track the phase between the missile and one GPS satellite during the first 60 seconds of boost (i.e., during Stage I). The filter rate was fixed at 50 Hz. The satellite-to-missile range, range rate, and range acceleration are depicted in Fig. 4. The relationship between the phase θ and the range ρ is given by

$$\rho(t) = \frac{c\theta(t)}{2\pi f} \quad (48)$$

where c is the speed of light and f is the transmitted frequency of the sinusoid. We concentrate in this paper on tracking the GPS L1 carrier at a transmitted frequency of 1.575 GHz. We assume without loss of generality (see Fig. 1) that the magnitude of the carrier is unity.

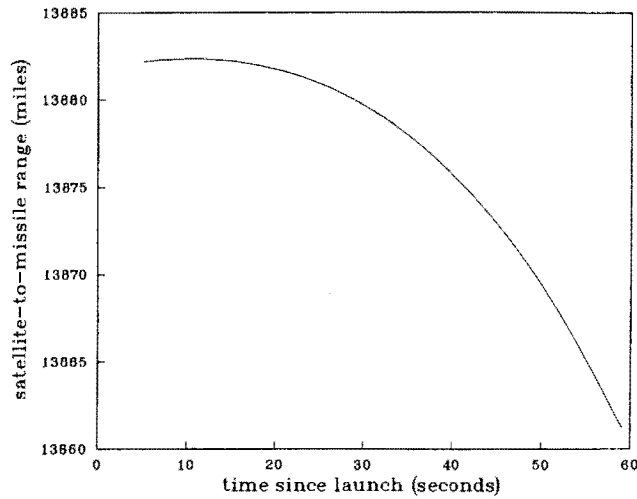
Typical carrier-to-noise ratios (CNR's) for GPS are around 30 to 40 dB-Hz [6]. But if atmospheric conditions are severe or jamming is present, the CNR could drop into the twenties or even lower. The CNR is related to the variance of the measurement noise (R) by [8, p. 282]

$$\text{CNR} = \frac{1}{2TR} \quad (49)$$

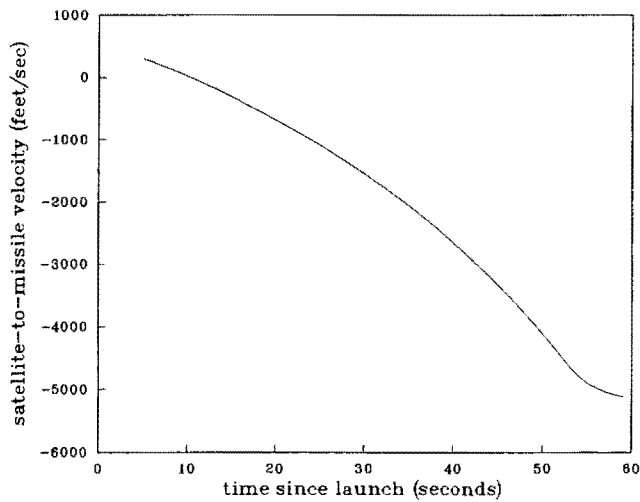
where $T = 0.02$ seconds is the filter rate. The measurement noise in the simulations was generated with a laplacian (exponential) probability density function.

A. Training Set-Up

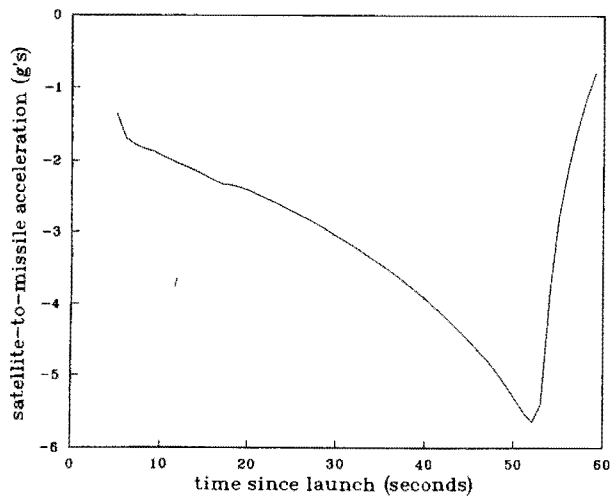
The gradient descent method and genetic algorithm described earlier were used to optimize the fuzzy membership functions. Training took place from 45–55 seconds following



(a)



(b)



(c)

Fig. 4. (a) Satellite-to-missile range, (b) Satellite-to-missile range rate, (c) Satellite-to-missile range acceleration.

launch, the most dynamic 10-second interval of stage 1 (see Fig. 4). CNR was set to 18 dB-Hz during training.

The gradient descent learning parameters η_c , η_b and η_y were all set to 0.3. The membership functions were constrained

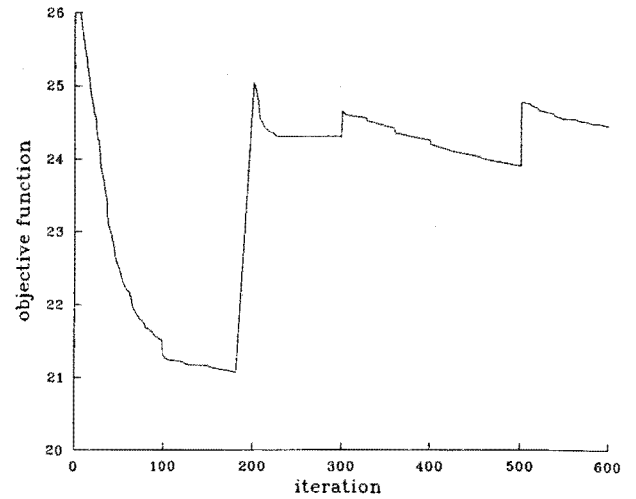


Fig. 5. Gradient descent training of fuzzy membership functions.

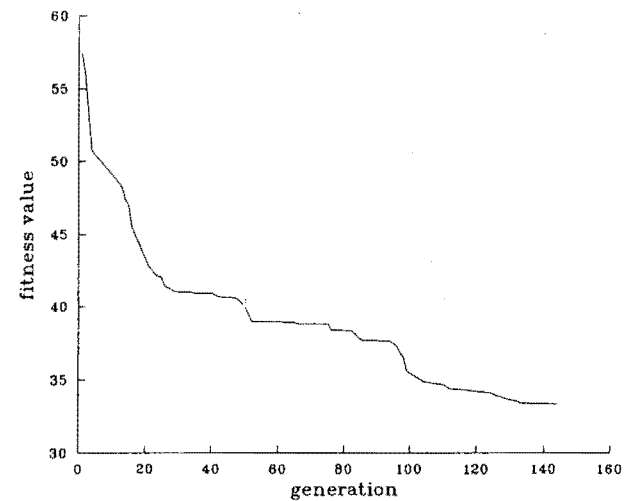


Fig. 6. Genetic training of fuzzy membership functions.

to be symmetric triangles, and the error function E in (21) was optimized with respect to 35 parameters, the centroids of each of the membership functions (21 total), and the width of each of the input membership functions (14 total). As training progressed and PLL performance improved, more and more samples were used in the training procedure. For the first 200 iterations, a single 10-second simulation was used for each update of the parameters, so N in (21) was 10 seconds \times 50 Hz = 500. For the next 100 iterations, 10 Monte-Carlo samples of 10 seconds each were used for each update. For the next 200 iterations, 30 Monte-Carlo samples were used. Finally, for the last 100 iterations, 100 Monte-Carlo samples were used. So N in (21) gradually increased from 500–5000–15 000–50 000. Fig. 5 shows the decrease of the objective function as training progressed. The sudden jumps in the objective function value are due to the increases in the size of the training set.

The population size used for the GA was fixed at 40 members. One hundred Monte-Carlo samples of 10 seconds each were used to evaluate the fitness of each member of the population. The fitness function parameters in (47) were set at $k_1 = 20$, $k_2 = 10$, and $k_3 = 1$. The fittest 10% of the population was cloned at the end of each generation. The least

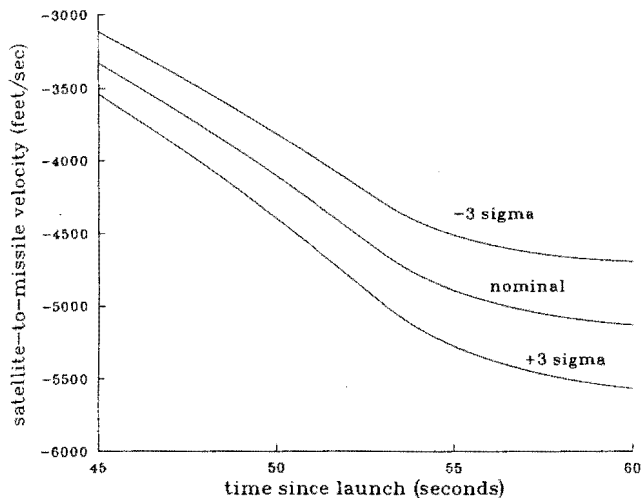


Fig. 7. Satellite-to-missile range rates for nominal and off-nominal missile trajectories.

fit 50% of the population was exterminated at the end of each generation. Then the fittest 50% of the population (including the clones) mated to restore the population size to 40. The probability of an offspring undergoing a 1-bit mutation was set at 1%. Five bits were used for each parameter, giving a $1/32$ radian (approximately 1.8 degree) resolution for each parameter. The membership functions were constrained to be symmetric trapezoids, and the fitness function in (47) was optimized with respect to 63 parameters, three parameters for each of the seven membership functions of the two inputs and the output (see Fig. 3). Fig. 6 shows the improvement in fitness as the population evolved.

The resulting optimized membership functions were quite different from the original membership functions. Apparently the new membership functions reflected information about the trajectory which was used in the simulation, although it was difficult to interpret the changes intuitively.

B. Simulation Results

Perhaps the most successful PLL design approach reported to date for highly dynamic trajectories in a noisy environment is the use of H_2 (Kalman) or hybrid H_2/H_∞ filtering [21], [23]. Therefore, in this section we compare the performance of the fuzzy PLL's with the H_2 and H_2/H_∞ PLL's. The H_2 PLL is a fourth order Kalman filter with a state vector composed of the phase and its first three derivatives. The H_2/H_∞ PLL is a fourth order hybrid Kalman/minimax filter, with a filter gain given by

$$K = dK_2 + (1 - d)K_\infty \quad (50)$$

where K_2 is the steady-state Kalman gain, K_∞ is the H_∞ gain, and $d \in [0, 1]$ is the relative weight given to H_2 performance. In [21] it is shown that $d \approx 0.4$ is near-optimal for GPS PLL design (with respect to loss of lock). So the H_2/H_∞ results presented in this section were obtained with $d = 0.4$. K_2 in (50) was obtained for $\text{CNR} = 18$ dB-Hz. (K_∞ is independent of CNR.) The optimal forgetting factor used to compute K_2 was found to be 1.055.

TABLE II
PROBABILITIES OF CYCLE SLIPS AND LOSS OF LOCK FOR VARIOUS PLL METHODS FOR THE NOMINAL TRAJECTORY

PLL method	CNR = 18		CNR = 19		CNR = 20	
Kalman	0.39	0.29	0.17	0.13	0.06	0.04
H_2/H_∞	0.60	0.05	0.34	0.02	0.13	0.00
nominal fuzzy	0.96	0.73	0.80	0.45	0.53	0.24
gradient fuzzy	0.54	0.38	0.31	0.18	0.14	0.09
genetic fuzzy	0.34	0.06	0.24	0.09	0.07	0.00

TABLE III
PROBABILITIES OF CYCLE SLIPS AND LOSS OF LOCK FOR VARIOUS PLL METHODS FOR THE $\pm 3\sigma$ TRAJECTORIES (CNR = 18)

PLL method	$+3\sigma$		-3σ	
Kalman	0.41	0.31	0.37	0.27
H_2/H_∞	0.63	0.06	0.59	0.02
nominal fuzzy	0.98	0.75	0.93	0.70
gradient fuzzy	0.61	0.44	0.46	0.24
genetic fuzzy	0.44	0.15	0.41	0.13

Table II shows a comparison of various PLL methods using the trajectory shown in Fig. 4 for $\text{CNR} = 18, 19$ and 20 . It is seen that the H_2/H_∞ and GA fuzzy PLL's are best as far as $P(\text{loss of lock})$, but the Kalman and GA fuzzy PLL's are best as far as $P(\text{slip})$. As expected, the GA fuzzy PLL performs better than the gradient fuzzy PLL due to the fact that the GA has more degrees of freedom in its search for the optimum rules.

Since the optimal fuzzy PLL's were trained for a specific trajectory, it is of interest to examine how robust these fuzzy PLL's are to departures from the nominal trajectory. Fig. 7 shows the satellite-to-missile velocity for three cases: the nominal missile trajectory, and plus and minus three sigma missile trajectories. The off-nominal trajectories were obtained by simulating plus and minus three sigma thrust magnitude, burn rate, and drag. Table III shows a comparison of the different PLL's for the $\pm 3\sigma$ trajectories. The results are seen to be similar to those observed in Table II. Comparison of Tables II and III show the robustness of the optimal fuzzy PLL's to departures from the nominal trajectory.

One potential drawback of a fuzzy estimator relative to the steady-state H_2 and H_2/H_∞ filters is the increased real-time computational effort. Most of the computational time of the steady-state filters consists of a simple multiplication of a 4×4 matrix with a four-element vector. Fuzzy filtering, however, includes fuzzification, correlation-product inference, and defuzzification, and is significantly more time-consuming. But the effort is still small enough to not be of concern in the real-time application considered in this paper.

V. CONCLUSION

A fuzzy estimator has been proposed and applied to phase-locked loop design. This approach offers the benefits of fuzzy logic (intuitiveness, simplicity, robustness, and maintainability) while providing performance on par with analytical estimation methods (Kalman and H_∞ filtering). In addition, the fuzzy estimator offers the possibility of training if a nominal receiver trajectory is known *a priori*. It is thus recommended that fuzzy logic be given serious consideration

for PLL design in particular, and for state estimation in general.

ACKNOWLEDGMENT

The authors thank S. Pham of the TRW Systems Integration Group for assistance in generating the data represented in Fig. 7.

REFERENCES

- [1] B. Anderson and J. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.
- [2] K. Atkinson, *An Introduction to Numerical Analysis*. New York: Wiley, 1989.
- [3] J. Bezdek, "Editorial: Fuzzy models—What are they, and why?" *IEEE Trans. Fuzzy Systems*, vol. 1, no. 1, pp. 1–5, Feb. 1993.
- [4] J. Dougherty, H. El-Sherief, D. Simon, and G. Whitmer, "A design approach for a GPS user segment for aerospace vehicles," in *Amer. Contr. Conf.*, San Francisco, CA, June 1993, pp. 935–939.
- [5] F. Guély and P. Siarry, "Gradient descent method for optimizing various fuzzy rule bases," in *IEEE Int. Conf. Fuzzy Syst.*, San Francisco, CA, Mar. 1993, pp. 1241–1246.
- [6] W. Hurd, J. Statman, and V. Vilnrotter, "High dynamic GPS receiver using maximum likelihood estimation and frequency tracking," *IEEE Trans. Aerospace Electron. Syst.*, vol. 23, pp. 425–436, July 1987.
- [7] P. Janiczek and S. Gilbert, Eds., *Global Positioning System Papers*. Washington, DC: Institute of Navigation, 1980, 1984, 1986, vols 1–3.
- [8] M. Jeruchim, P. Balaban, and K. Shanmugan, *Simulation of Communication Systems*. New York: Plenum, 1992.
- [9] C. Karr and E. Gentry, "Fuzzy control of pH using genetic algorithms," *IEEE Trans. Fuzzy Systems*, vol. 1, no. 1, pp. 46–53, Feb. 1993.
- [10] B. Kosko, *Neural Networks and Fuzzy Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
- [11] M. Lee and H. Takagi, "Integrating design stages of fuzzy systems using genetic algorithms," in *IEEE Int. Conf. on Fuzzy Systems*, San Francisco, CA, Mar. 1993, pp. 612–617.
- [12] W. Lindsey and C. Chie, "A survey of digital phase-locked loops," *Proc. of the IEEE*, vol. 69, pp. 410–431, Apr. 1981.
- [13] ———, Eds., *Phase-Locked Loops*. New York: IEEE Press, 1986.
- [14] D. Maclay and R. Dorey, "Applying genetic search techniques to drivetrain modeling," *IEEE Contr. Syst. Mag.*, vol. 13, no. 53, pp. 50–55, June 1993.
- [15] E. Mamdani, "Twenty years of fuzzy control: Experiences gained and lessons learnt," in *IEEE Int. Conf. Fuzzy Syst.*, San Francisco, CA, Mar. 1993, pp. 339–344.
- [16] C. Moore, C. Harris, and E. Rogers, "Utilising fuzzy models in the design of estimators and predictors: An agile target tracking example," in *IEEE Int. Conf. Fuzzy Syst.*, San Francisco, CA, Mar. 1993, pp. 679–684.
- [17] K. Nagpal and K. Poolla, "On l_1 filtering and smoothing," in *IEEE Conf. Dec. Contr.*, Tucson, AZ, Dec. 1992, pp. 1238–1242.
- [18] D. Schwartz and G. Klir, "Fuzzy logic flowers in Japan," *IEEE Spectrum*, vol. 29, no. 7, pp. 32–35, July 1992.
- [19] D. Simon and H. El-Sherief, "GPS receiver design for integrated navigation systems," in *Institute of Navigation 49th Annu. Meet.*, Cambridge, MA, June 1993, pp. 437–445.
- [20] ———, "Design of global positioning system receivers for integrated inertial navigation systems," in *IEEE Conf. Dec. Contr.*, San Antonio, TX, Dec. 1993, pp. 1476–1477.
- [21] ———, "Hybrid H_2/H_∞ estimation for phase-locked loop filter design," in *Amer. Contr. Conf.*, Baltimore, MD, June 1994, pp. 2212–2216.
- [22] C. Tao, W. Thompson, and J. Taur, "A fuzzy logic approach to multidimensional target tracking," in *IEEE Int. Conf. Fuzzy Syst.*, San Francisco, CA, Mar. 1993, pp. 1350–1355.
- [23] V. Vilnrotter, S. Hinedi, and R. Kumar, "A comparison of frequency estimation techniques for high-dynamic trajectories," *JPL Publication 88-21*. Pasadena, CA: Jet Propulsion Lab., Sep. 1988.
- [24] P. Voulgaris, "Optimal l_∞ to l_∞ estimation," in *Amer. Contr. Conf.*, San Francisco, CA, June 1993, pp. 940–944.
- [25] I. Yaesh and U. Shaked, "A transfer function approach to the problems of discrete-time systems: H_∞ -optimal linear control and filtering," *IEEE Trans. on Automatic Contr.*, vol. 36, pp. 1264–1271, Nov. 1991.
- [26] T. Yamakawa, "A fuzzy inference engine in nonlinear analog mode and its application to a fuzzy logic control," *IEEE Trans. Neural Networks*, vol. 4, no. 3, pp. 496–522, May 1993.