GNSS Solutions:

Carrier–to–Noise Density and AI for INS/GPS Integration

"GNSS Solutions" is a regular column featuring questions and answers about technical aspects of GNSS. Readers are invited to send their questions to the columnist, **Dr. Mark Petovello**, Department of Geomatics Engineering, University of Calgary, who will find experts to answer them. His e-mail address can be found with his biography at the conclusion of the column.

What is C/N_0 and how is it calculated in a GNSS receiver?

C/N_0 (carrier-to-noise density) is the ratio of received carrier (i.e., signal) power to noise density. Higher C/N_0 results in reduced data bit error rate (when extracting the navigation data from the GNSS signals) and reduced carrier and code tracking loop jitter. Reduced carrier and code tracking loop jitter, in turn, results in less noisy range measurements and thus more precise positioning.

Note that C/N_0 is *not* the same as SNR (signal-to-noise ratio), although the terms are sometimes used interchangeably. Effectively, C/N_0 assumes that the noise has infinite bandwidth (and thus power) and therefore characterizes it using a density, that is, as the amount of noise power per unit of bandwidth (i.e., watts/Hz).

Conversely, SNR considers the total noise power in a certain bandwidth (i.e., watts). C/N_0 can be derived from SNR if the noise bandwidth of the SNR measurement is known. For example, one manufacturer's GPS receiver displays SNR as a figure of merit for a GNSS signal; however, in this receiver, C/N_0 is typically 30 decibels higher than the displayed SNR.

C/N_0 provides a metric that is more useful for comparing one GNSS receiver to another than SNR because the bandwidth of the receivers is eliminated in the comparison. How the effective noise bandwidth (NBW) of a GNSS receiver is chosen is beyond the scope of this article but can be computed/defined based on a receiver's hardware implementation, as will be briefly discussed later.

Theoretical C/N_0

When developing the analog front end for a digital-sampling GNSS receiver, it is useful to examine a theoretical estimate of C/N_0 as an aid in determining the total gain required by the analog front end and the receiver noise temperature required for the desired level of receiver performance.

An estimate of C/N_0 is given by

$$C/N_0 = S_r + G_a - 10 \cdot \log_{10}(k) - 10 \cdot \log_{10}(T_{sys}) - L \text{ (dB-Hz)}$$

Some typical values for GPS L1 CA code are:

S_r = received GPS L1 CA signal power = 158.5 dBW

G_a = receiver antenna gain toward satellite = 0dBi

k = Boltzman's constant = 1.38×10^{-23} Watt-sec/K

T_{sys} = System Noise Temperature = $T_{source} + T_{receiver}$ = Source + Receiver Noise Temperature

L = implementation losses = 2dB

With an open view of the sky, a GPS L1 CA-code receiver will typically see a C/N_0 range of 35 to 50 dB-Hz depending on the elevation angle of the transmitting satellite and the gain
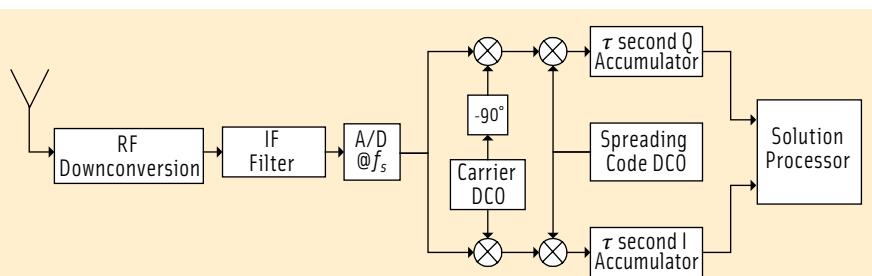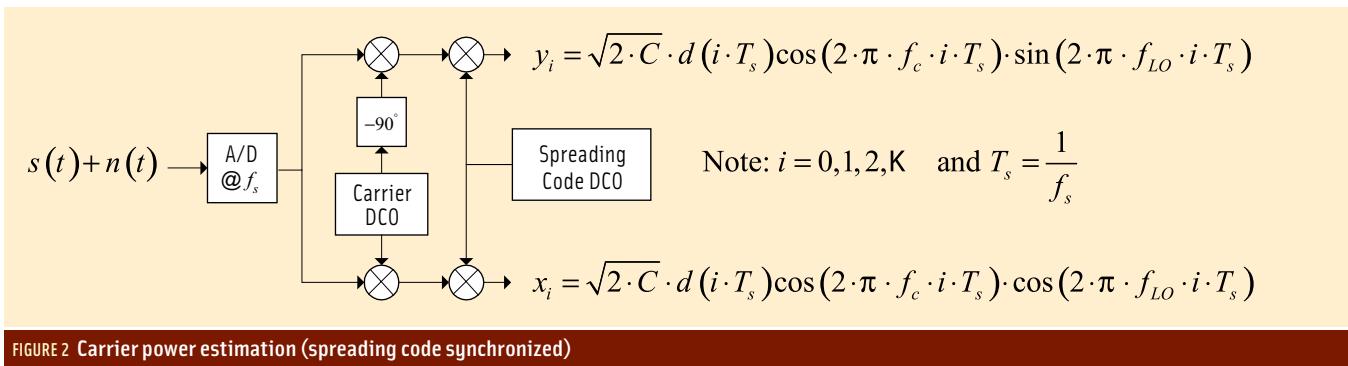

FIGURE 1 **Typical GNSS receiver block diagram**

$$y_i = \sqrt{2 \cdot C} \cdot d\left(i \cdot T_s\right) \cos\left(2 \cdot \pi \cdot f_c \cdot i \cdot T_s\right) \cdot \sin\left(2 \cdot \pi \cdot f_{LO} \cdot i \cdot T_s\right)$$

Note: $i = 0, 1, 2, \mathrm{K}$ and $T_s = \dfrac{1}{f_s}$

$$x_i = \sqrt{2 \cdot C} \cdot d\left(i \cdot T_s\right) \cos\left(2 \cdot \pi \cdot f_c \cdot i \cdot T_s\right) \cdot \cos\left(2 \cdot \pi \cdot f_{LO} \cdot i \cdot T_s\right)$$

FIGURE 2 **Carrier power estimation (spreading code synchronized)**

pattern of the receive antenna. The system noise temperature, T_{sys}, is usually dominated by receiver noise temperature, but we cannot ignore source noise temperature. Source noise is either antenna noise temperature (sometimes called sky temperature) or that of a signal simulator.

An antenna might typically yield a source noise temperature of 100 K whereas a signal simulator would have a source noise temperature of 290 K. In comparison, a receiver might have a noise temperature of around 420 K. This, along with the 100 K assumption about antenna source noise temperature, would yield a C/N$_0$ of 40.94 dB-Hz.

Readers interested in a more in depth discussion of antenna noise temperature or on the impact of C/N$_0$ on GNSS receiver performance are referred to the Additional Resources section at the end of this article.

Estimating CN$_0$ in a GNSS Receiver

Carrier power to noise density can be estimated accurately based on measurements taken in a digital sampling GNSS receiver. **Figure 1** displays the block diagram of a typical GNSS receiver.

Carrier Power Estimation. Refer to **Figure 2** for the following discussion of carrier power estimation development. Assume the spreading code and carrier DCOs (digitally controlled oscillators) are synchronized with the received signal $s(t) = \sqrt{2 \cdot C} \cdot d\left(t\right) \cdot c\left(t\right) \cdot \cos\left(2 \cdot \pi \cdot f_c \cdot t\right)$. Note that the power of $s(t)$ is C, $c(t) = \pm 1$ is the spreading code, $d(t) = \pm 1$ is the data modulated onto the signal, and f_c is the carrier frequency.

Recall that the spreading code DCO is assumed to be synchronized with $s(t)$, that is, $c(t)$ is wiped off of x_i and y_i, the inputs to the in-phase (I) and quadrature (Q) accumulators respectively. Also recall that the carrier DCO is assumed to be synchronized with $s(t)$, that is, $f_c = f_{LO}$. Applying a little bit of trigonometric manipulation results in

$$x_i = \frac{\sqrt{2 \cdot C}}{2} \cdot d\left(i \cdot T_s\right) \cdot \left[1 + \cos\left(4 \cdot \pi \cdot i \cdot T_s \cdot f_c\right)\right] + \text{noise}$$

The accumulations are synchronized so that the data, $d(i \cdot T_s)$, are constant over any given accumulation period. This constant value of the data will be referred to as $\overline{d}$ (Note: $\overline{d} = \pm 1$). With an A/D sampling frequency of f_s samples per second and an accumulation period of τ seconds, the output of the I accumulator is approximately

$$I_{ACC} \approx f_s \cdot \tau \cdot \overline{d} \cdot \frac{\sqrt{2 \cdot C}}{2} + \text{zero mean noise}$$

The sign of the output of the accumulator may change from one accumulation to the next depending on the data, $\overline{d}$. However, for carrier power output estimation the accumulator output is squared; so, the sign of the data does not matter. Also note that the high-frequency term $\cos\left(4 \cdot \pi \cdot i \cdot T_s \cdot f_c\right)$ has a negligible (may or may not be zero depending on the accumulator) effect on the accumulator output, hence, the approximate equality in the previous equation.

The receiver squares and averages the output of the I accumulator which yields

$$\overline{I_{ACC}^2} \approx (f_s \cdot \tau)^2 \cdot \frac{C}{2}$$

At this point in the analysis, assuming a reasonably strong signal, the noise is negligible compared to the signal (because the receiver has spreading code and carrier synchronization and because of averaging). So, the estimate of carrier power is

$$C = \frac{2 \cdot \overline{I_{ACC}^2}}{(f_s \cdot \tau)^2}$$

The other channel, y_i, performs a similar operation that is 90 degrees out of phase. In the case of a quadrature spreading code (such as GPS L5) this would have to be taken into account.

To impart a visual understanding of what is going on in the receiver; refer to the scatter plot shown in **Figure 3**. This scatter plot displays the actual τ second accumulator outputs of a GNSS receiver that is tracking a GLONASS L1 signal modulated with binary phase shift keyed (BPSK) data.

Each individual dot is one reading of the I and Q accumulators. The I value is given on the horizontal "I" axis and Q value on the vertical "Q" axis in Figure 3. Due to BPSK data, $d(t)$, that is modulated onto the signal, two large clusters of small dots appear on the "I" axis on either side of the "Q" axis. Th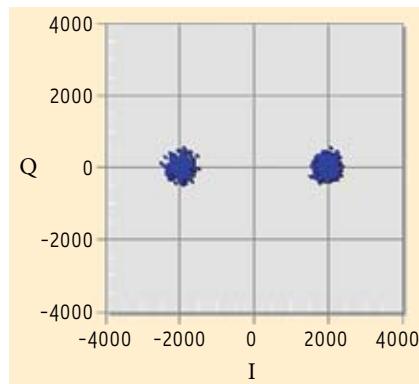e SNR of the signal depicted in Figure 3 is the ratio of the mean squared to the variance of the data in the figure. C/N_0 can be determined once the noise bandwidth of this measurement is known.

Because this is a somewhat important concept to grasp, let's restate it in a different way. Referring again to Figure 3, the carrier power can be interpreted as the distance of the two clusters (on the I axis) from the origin, and the noise power is related to the spread of the data within the two clusters among themselves.

With this in mind, by taking the ratio of the mean squared to the variance of the data, the absolute value of the axis values is irrelevant. To get from noise power to noise density, the noise bandwidth of the receiver must be known.

Noise Power Estimation

Noise power can be estimated using the Q arm of the receiver (i.e., the upper arm in Figure 1 and Figure 2). Note that noise power can also be estimated using the I accumulator if the mean value of the de-spread signal is accounted for. On a more complicated note, if a receiver were to display different

FIGURE 3 GLONASS BPSK signal after code and carrier synchronization and τ second accumulation (actual τ second accumulator outputs of a GNSS receiver).

I and Q variances (noise powers) that would be an indication of a phase noise problem with the receiver. A discussion of phase noise is beyond the scope of this article.

When the GNSS receiver is properly code and carrier synchronized, the transmitted signal, $s(t)$, is rotated into the I accumulator and thus it may safely be ignored in the Q-channel. Again, recall that a quadrature spreading code such as GPS L5 would require estimating noise on a non-code tracking channel or dealing with the mean value of the signal on a tracking channel.

The input to the Q accumulator is y_i. Then the Q accumulator output is

$$Q_{ACC} = \sum_{i=1}^{f_s \cdot \tau} y_i$$

Squaring the output of the accumulator and averaging (estimating the expected value of the squared output) yields

$$E\left\{Q_{ACC}^2\right\} = \overline{Q_{ACC}^2} = E\left\{\left(\sum_{i=1}^{f_s \cdot \tau} y_i\right)^2\right\}$$

Assuming that the y_i are zero-mean and uncorrelated results in

$$E\left\{Q_{ACC}^2\right\} = \overline{Q_{ACC}^2} = \sum_{i=1}^{f_s \cdot \tau} E\left\{y_i^2\right\} = f_s \cdot \tau \cdot E\left\{y_i^2\right\} = f_s \cdot \tau \cdot \sigma_y^2$$

This means that the accumulated and averaged noise power, $E\left\{Q_{ACC}^2\right\}$, is just the variance, σ_y^2, of the samples y_i multiplied by the number of accumulation samples.

Note that it is assumed $\sigma_x^2 = \sigma_y^2$ (no phase noise) and that the variance of $n(t)$ is $\sigma_n^2 = \sigma_y^2 + \sigma_x^2 = 2 \cdot \sigma_y^2$. Also, note that the term $f_s \cdot \tau$ is not squared in the result for $E\left\{Q_{ACC}^2\right\}$. This follows from the stated assumptions and the definition of expected value from probability theory.

The assumption that the y_i are uncorrelated is valid given that the choice of f_s is not excessively large compared to the noise bandwidth (NBW) of the intermediate frequency (IF) filter. A full discussion of how f_s, noise bandwidth, and for that matter τ, are chosen is beyond the scope of this article.

C/N_0 Estimation

Given the carrier power and noise power estimates from the previous sections and knowledge of the noise bandwidth of the IF filter, the C/N_0 can be estimated. In a typical GNSS receiver, the IF filter sets the NBW. This might typically be a two megahertz–wide SAW (surface acoustic wave) filter for a low performance GPS L1 CA receiver.

The noise bandwidth of a SAW filter is approximately the same as its three-decibel bandwidth. It is then straightforward to convert from noise power to noise density by dividing by the noise bandwidth:

$$N_0 = \frac{\sigma_n^2}{NBW} = \frac{2 \cdot \sigma_y^2}{NBW} = \frac{2 \cdot \overline{Q_{ACC}^2}}{f_s \cdot \tau \cdot NBW}$$

The estimates of the power of the received signal, *C*, and N_0 as previously derived, result in a calculation of carrier-to-power density equation, as follows:

$$\frac{C}{N_0} = \frac{\overline{I_{ACC}^2} \cdot NBW}{\overline{Q_{ACC}^2} \cdot f_s \cdot \tau}$$

Again, refer to the scatter plot in Figure 3 to help visualize this result.

As stated earlier, many receivers display SNR as a figure of merit for a given signal. Typically, this SNR equation is

$$SNR = 10 \cdot \log_{10}\left(\frac{\overline{I_{ACC}^2}}{2 \cdot \overline{Q_{ACC}^2}}\right) dB$$

Given this definition of SNR and the derived C/N_0, the two are related as follows:

$$\frac{C}{N_0}(dB\text{-}Hz) = SNR(dB) + 10 \cdot \log_{10}\left(\frac{2 \cdot NBW}{f_s \cdot \tau}\right)$$

This conversion from SNR to C/N_0 is specific to the given definition of SNR. However, regardless of how a manufacturer defines SNR, an analysis similar to that given in this article can be applied to determine C/N_0.

Additional Resources

[1] Kaplan, E. D., *Understanding GPS Principles and Applications,* Chapters 5 and 6 by Phillip Ward, Artech House, Boston, 1996

[2] Parkinson, B. W. and J. Spilker, *Global Positioning System: Theory and Applications Volume 1,* p. 344 by A.J. Van Dierendonck, American Institute of Aeronautics and Astronautics, Washington, 1996

BRAD BADKE

BRAD BADKE received bachelor's and master's degrees in electrical engineering from Kansas State University and a Ph.D. in electrical engineering from Arizona State University. For the past six years Brad Badke has been a member of the core GPS receiver development group at Hemisphere GPS. Previously, he was involved in research and development in the field of wireless communications.

What are the merits and limitations of artificial intelligence methods for INS/GPS integration?

Most current modules that integrate inertial navigation and GPS (INS/GPS) technologies typically rely on Kalman filtering (KF) to exploit their individual benefits and provide a reliable navigation solution. However, KF-based integration techniques for INS/GPS suffer from several limitations

related to its predefined dynamics model, observability (i.e., the ability to determine, or observe, all of the relevant system parameters), the necessity of having accurate stochastic models of sensor random errors and accurate a priori covariance information for both INS and GPS data.

Over the years, non-linear integration modules based on artificial intelligence (AI) were proposed either

> **A Kalman filter is a recursive algorithm that calculates error estimates based on external measurements of INS and GPS.**

as a complete replacement for KF or with augmentation by KF. Such modules were usually targeted for robust

positioning applications in urban canyons, especially when these solutions incorporated low-end tactical grade or micro-electro-mechanical system (MEMS)–based sensors.

In this article, we wish to answer important questions and address some of the numerous concerns raised about AI-based INS/GPS integration.

Features of AI Compared to KF for INS/GPS Integration

AI techniques are generally platform-independent systems that do not require detailed knowledge of the integrated sensors and technologies, unlike Kalman filtering that requires accurate stochastic models of inertial sensor errors and covariance matrices of both INS and GPS data. However, AI modules still require correct system parameters obtained through training in order to be able to provide a reliable

navigation solution. These parameters are unique to the inertial sensors and the GPS receivers used and are independent of the moving platform or the trajectory.

A Kalman filter is a recursive algorithm that calculates error estimates based on external measurements of INS and GPS. An AI module is more heuristic, such that once the parameters of an AI module are set, it will map the input to an output. Furthermore, AI techniques do not require mathematical models of the moving platform (e.g., such as the dynamic error model of INS and the measurement model for GPS).

Instead, AI systems learn the relationships between the associated inputs and their respective errors during the training/update process to tune the internal AI system parameters. For loosely coupled INS/GPS integration, during GPS outages the AI module
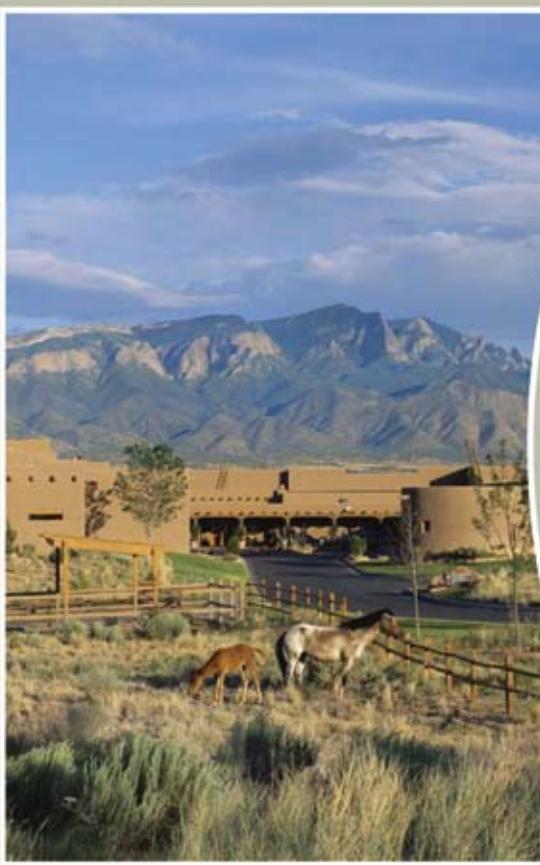
uses its recently updated parameters to predict the INS errors. In this respect, the AI system is able to account for the non-linear errors that are only approximated in traditional KF-based approaches due to the linearization of the INS dynamic error model.

Current AI–Based Techniques for INS/GPS Integration

Several different AI techniques are presently used to integrate INS/GPS. The core of these techniques uses the methods of artificial neural networks (ANN), fuzzy logic, and hybrid neuro-fuzzy algorithms. A detailed discussion of each of these topics is beyond the scope of this article, but we will provide an overview here. Details are available in the material cited in the "Additional Resources" section at the end of this column.

ANNs are based on a network that is thought to mimic the way a brain processes data and learns. The fundamental unit of an artificial network is the neuron. **Figure 1** depicts the general architecture of a multi-layer feed-forward neural network (MFNN).

An MFNN has at least one input layer and one output layer of neurons. More intricate networks will also have one or several hidden layers. Interconnections between neurons have an associated weight (w_{ij}), where the higher the weight, the stronger the connection between the two neurons. The interconnection weights are considered to be the memory of the network.

The architecture of Figure 1 can be employed to establish a model between the INS errors (the MFNN output) and the INS position and velocity (the MFNN inputs). Such an empirical non-linear model is built during the update procedure (training) while the GPS position and velocity updates are available. Should a GPS outage occur, the module switches to prediction mode where the recent empirical model is used to process the INS position and velocity at the inputs and provides the corresponding errors at the output.
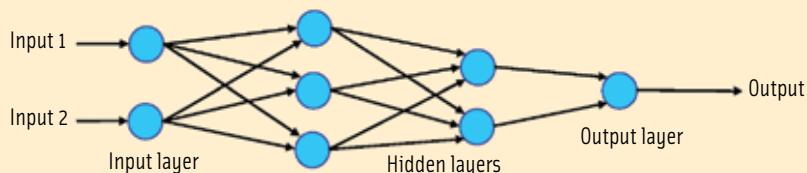

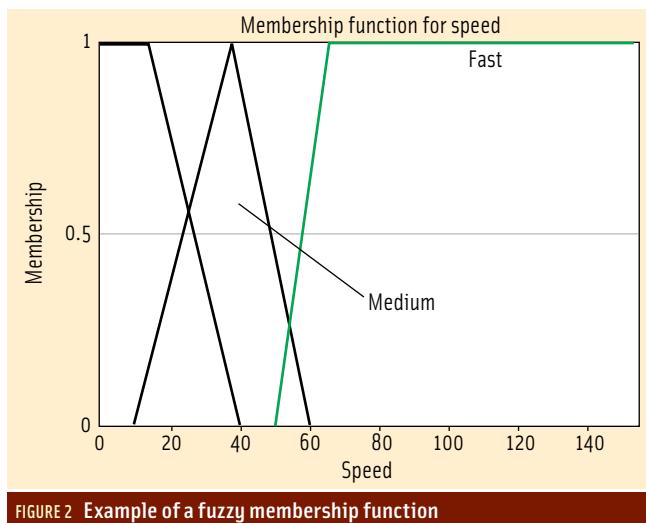FIGURE 1 Multi-layer feed-forward neural network

FIGURE 2 **Example of a fuzzy membership function**

Fuzzy-based AI techniques employ the use of fuzzy set theory, fuzzy operators, and fuzzy inference rules to map input to output. In general, the INS/GPS measurements are *fuzzified* in a step in which the measurements are mapped to a linguistic variable through fuzzy membership functions. An example of a membership function is shown in **Figure 2**.

In performing this function, INS/GPS measurements (or their difference) are the inputs that are classified into sets with an associated membership value (normally between 0 and 1) that rates the degree to which that value belongs to that set. Fuzzy sets are represented linguistically, with terms such as "slow", "medium" and "fast."

Once the measurement has been fuzzified, the mapping of input to an output is done through the fuzzy inference system with fuzzy rules and operators. The fuzzy rules are IF-THEN rules, for example, "If the GPS VELOCITY is STATIONARY and the INS VELOCITY is SLOW, then INS ERROR is HIGH".

After all the fuzzy rules are resolved, the outputs are then aggregated using fuzzy operators to determine their degree of membership to the fuzzy output sets. The fuzzy outputs can then be defuzzified using various fuzzy operations, to produce a useable defuzzified output value.

Neuro-fuzzy–based techniques or adaptive neuro-fuzzy inference systems (ANFIS) are hybrid systems combining the concepts of neural networks and fuzzy logic systems. An ANFIS is functionally equivalent to a fuzzy inference system and employs a neural-network learning algorithm such as the backpropagation gradient descent method to tune the initial parameters of the fuzzy model.

Training an AI module

Before the output of an AI module can be considered reliable, it must first be trained. For optimal performance, training is undertaken during GPS signal availability to determine the best AI system parameters. Ensuring that the initial training data contain a wide variety of vehicle dynamics — such as fast acceleration, sudden stops, slow/fast traffic, and so forth — is extremely important. This ensures generalization of the system parameters so that the AI-module is capable of reliably interpreting the input data and mapping it to an appropriate output, regardless of the operating environment.

In the case of a neural network, the initial weight parameters are usually random and require module-training to obtain correct initial weights prior to the actual navigation mission. The error between the actual output of the neural network and the desired output (i.e., the GPS value) is then used to update the weights between neurons.

The weight update procedure is based on the selected learning algorithm (e.g., Quickpropagation, Levenberg-Marquardt) that would minimize the overall error function. The training data is reprocessed until a user-chosen minimum error is reached or a certain number of training epochs are completed. At this point the weights are set.

For fuzzy systems, much of the initial parameters are set up during the design of the system. Using training data, the designer would need to run the system through training trials to determine an initial collection of fuzzy sets, membership functions, fuzzy rules, fuzzy operators and antecedent/consequent parameters, when applicable, which can be further tuned during the navigation mission.

Neuro-fuzzy systems are trained in a similar manner as in neural networks. Once the designer has set up the initial parameters of the neuro-fuzzy system similar to that developed for a regular fuzzy system, the system parameters are then further tuned and updated using a neural-network learning algorithm during the navigation mission.

AI–Module Design Considerations

A major consideration is the actual architecture of the system. In the case of ANN, designers must choose from among many different types of architectures (e.g., multi-layer feed forward networks, and radial basis function neural networks). For fuzzy systems, the fuzzy method to use (e.g., Mamdani), the number of rules for the fuzzy inference system, the fuzzy operators, and the membership functions are important design criteria to examine.
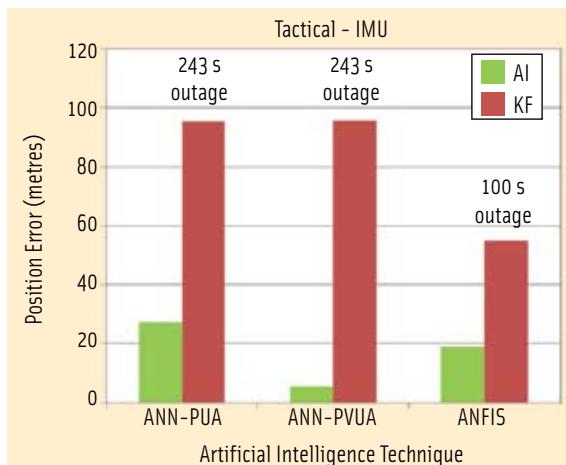
FIGURE 3 Tactical IMU comparison chart
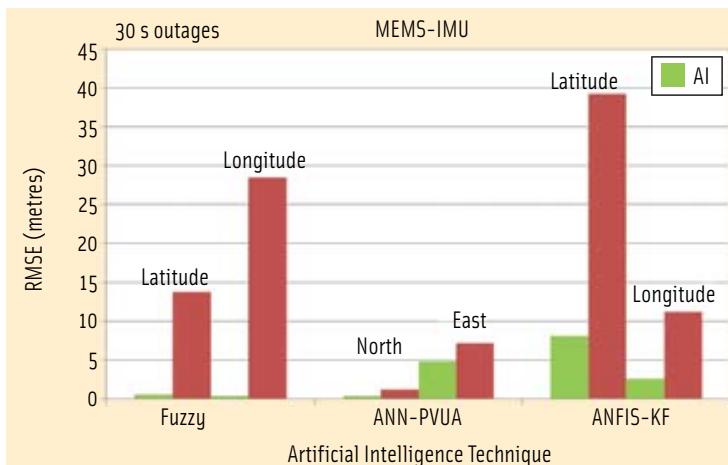
FIGURE 4 MEMS-based IMU comparison chart

AI technique	GPS Outage	Position error (m)		Error
		AI	KF	
ANN – PUA	243 s	27.16	95.5	RMSE
ANN – PVUA	243 s	5.41	95.8	RMSE
ANFIS	100 s	19	55	Max

TABLE 1 Tactical IMU data table

	GPS Outage	Latitude/North RMSE (m)		Longitude/East RMSE (m)	
		AI	KF	AI	KF
Fuzzy	30 s	0.61	13.75	0.42	28.43
ANN–PVUA	30 s	0.37	1.24	4.88	7.22
ANFIS – KF hybrid	30 s	8.1	39.2	2.6	11.2

TABLE 2 MEMS-based IMU data table

The complexity of an AI system is partially determined by the number of inputs into the AI module. Using only the position vector as input is simpler to implement but does not perform as well as a system that receives position and velocity vectors that requires a larger and more complex architecture.

For an ANN, the number of layers and neurons associated with the hidden layer is also a major design consideration. Generally, the more hidden neurons and layers, the greater the ability of the neural network to predict navigation errors, but this also requires longer training and update times. The designer could train several different network architectures to determine the best fit for the application. A similar design consideration for fuzzy systems is the number of fuzzy-inference rules used.

Finally, a designer also needs to use a correct learning algorithm. The designer must determine important features such as the computation complexity, storage space, prediction accuracy and training speed for the chosen AI architecture.

Comparison of AI and KF Module Performance

Overall, during medium to long GPS outages an AI module performs better than a KF module. In contrast, KF modules tend to give better results during short GPS outages.

Figure 3 presents performance results for three AI-based techniques tested on a tactical grade INS as well as corresponding results using Kalman filtering. The first two AI approaches use an ANN with a position-update architecture (PUA) and a position/velocity-update architecture (PVUA). The third approach uses an ANFIS. **Table 1** gives the details related to Figure 3.

Figure 4 shows some performance results for three AI techniques tested on a MEMS-based IMU compared to those from a KF. **Table 2** gives the details related to Figure 4, comparing Kalman filtering of a fuzzy module, PVUA, and an augmented ANFIS/FK module. Note that the results of Figure 4 and Table 2 belong to different road test trajectories; thus, KF results are not similar in the three cases.

Manufacturers

The test results reported in Figure 3 and Table 1 used the HG1700 ring laser gyro IMU from **Honeywell Inc.**, Phoenix, Arizona, USA, for evaluating the ANN–PUA and ANN–PVUA techniques and the LN-200 fiber-optic IMU from **Northrop Grumman Corporation**, Woodland Hills, California, USA. In the results reported in Figure 4 and Table 2, gyros from **Analog Devices, Inc.**, Norwood, Massachusetts, USA,

Mark Petovello is an Assistant Professor in the Department of Geomatics Engineering at the University of Calgary. He has been actively involved in many aspects of positioning and navigation since 1997 including GNSS algorithm development, inertial navigation, sensor integration, and software development.

Email: mark.petovello@ucalgary.ca

and accelerometers from **Colibrys SA**, Neuchâtel, Switzer-land, were used in testing fuzzy and ANFIS-KF hybrid tech-niques and a MEMS IMU from **Crossbow Technology, Inc.**, San Jose, California, USA, for the ANN-PVUA test.

Additional Resources

[1] Abdel-Hamid, W., and A. Noureldin N. El-Sheimy, "Adaptive Fuzzy Modeling of Low Cost Inertial Based Positioning Errors," *IEEE Transactions on Fuzzy Systems,* V15 (3), pp. 519–529, June 2007

[2] Chiang, K-W., and A. Noureldin N. and El-Sheimy, "The Utilization of Artificial Neural Networks for Multi-Sensor System Integration in Navigation and Positioning Instruments," *IEEE Transactions on Instrumentation and Measurement,* V55 (5), pp: 1606 – 1615, October 2006

[3] Noureldin, A., and T. Karamat, M. Eberts, and A. El-Shafie, "Performance Enhancement of MEMS Based INS/GPS Integration for Low Cost Navigation Applications," *IEEE Transactions on Vehicular Technology,* V58 (3), pp. 1077–1096, March 2009

[4] Noureldin, A., and A. El-Shafie and N. El-Sheimy, "Adaptive Neuro-Fuzzy Module For INS/GPS Integration Utilizing Position And Velocity Updates With Real-Time Cross Validation," *IET Radar, Sonar and Navigation,* V1(5), pp. 388–396, October 2007

[5] Semeniuk, L., and A. Noureldin, "Bridging GPS Outages Using Neural Network Estimates of INS Position and Velocity Errors," *Measurement Science and Technology,* IoP, V17 (9), pp. 2782–2798, September 2006

[6] Sharaf, R., and A. Noureldin, "Sensor Integration for Satellite Based Vehicular Navigation Using Neural Networks," *IEEE Transactions on Neural Networks,* V18 (2), pp. 589–594, March 2007 **IG**

LEPINSY CHANTHALANSY AND ABOELMAGD NOURELDIN

LEPINSY CHANTHALANSY is a master of applied science student at the Royal Military College (RMC) of Canada and is a member of the Navigation and Instrumentation Research Laboratory at RMC. LCdr Chanthalansy received his B.Eng. degree in computer engineering from RMC and his B.Sc. (General) from the University of Waterloo. He is a naval combat systems engineering officer in the Canadian Forces.

ABOELMAGD NOURELDIN is a cross-appointment associate professor at the departments of electrical and computer engineering in Queen's University and the Royal Military College (RMC) of Canada. His research is related to artificial intelligence, digital signal processing, and estimation with emphasis on their applications to INS/GPS. Dr. Noureldin holds a B.Sc. degree in electrical engineering , and M.Sc. in engineering physics from Cairo University and a Ph.D. in electrical and computer engineering from the University of Calgary. He is the director of the Navigation and Instrumentation Research Laboratory at RMC.