Handling the Data Rate Difference between GPS and INS Using AFS

Salam A. Ismaeel
Computer Man College for Computer Studies, Khartoum/Sudan
Email: salam.ismaeel@gmial.com

Abstract —**The inertial navigation system (INS) is a very fast system and produces data at a high data rate while the global positioning system (GPS) receiver is slower than the INS. Hence, there is a gap between the two systems reading data. For GPS/INS system integration, synchronization must be provided between them, to make it possible to compare the reading data of both systems.**
In this paper, the sampling rate difference between GPS and INS is solved by predicting or extrapolating the missing reading data of the GPS to be compatible with those of the INS data using adaptive fuzzy system (AFS).
Index Terms — **GPS, Integrated Navigation, Adaptive Fuzzy System**.

I. INTRODUCTION

The combination of the global positioning system (GPS) and inertial navigation system (INS) has become increasingly common in the past few years, because the characteristics of GPS and INS are complementary.
Some articles overcome this problem by choosing the GPS and INS systems with the same sampling rate as in reference [1], or this rate difference can be solved by predicting or extrapolating the missing reading data of the GPS to be compatible with those of the INS data. In reference [2] Kalman filter was used to predict the sampling between instants. While in reference [3], three methods were used which are, Newton, Spline, and artificial neural network (ANN).
This paper looks at ways to handle the difference in data rate between the two systems in order to make the integration possible using adaptive fuzzy system (AFS).
The following algorithm produces the general extrapolation process for GPS data prediction [3]:
Step 1: Read the GPS data from the GPS receiver.
Step 2: Calculate the number of samples between time intervals of the GPS data to be extrapolated depending on the data rate of the INS and GPS systems (i.e. if the INS and GPS data rate is 10 and 1 Hz, respectively, then the INS and GPS provide data at each 0.1 and 1 second respectively). Then the number of sampling instants to be predicted is 9 points between each two reading samples.
Step 3: Use one of the extrapolation methods to extrapolate the GPS data.
Step 4: Use the extrapolated GPS data with the INS data to compare and manipulate them in the integrated GPS/INS system.

II. ADAPTIVE FUZZY SYSTEM STRUCTURE

The equation which represents a fuzzy logic system with center average defuzzifier, product interface rule, non-singleton fuzzifier, and bell-shaped membership function is [4]:

$$f(\underline{x}) = \frac{\sum_{j=1}^{M} y_j \left[\prod_{i=1}^{n} \exp\left[-\left(\frac{x_i - m_{ij}}{\sigma_{ij}} \right)^2 \right] \right]}{\sum_{j=1}^{M} \left[\prod_{i=1}^{n} \exp\left[-\left(\frac{x_i - m_{ij}}{\sigma_{ij}} \right)^2 \right] \right]} \quad (1)$$

where

- $f(\underline{x})$: Fuzzy logic system output, which represent a function to n input variables x
- x_i : Input variable in the input universe of discourse
- y_j : Center of fuzzy set F_j, which is, a point in the universe of discourse V when membership function (μ_{Fj} (y)) achieves its maximum value, and μ_{Fj} (y) is given by a product interface engine
- M : The number of fuzzy rules
- N : The number of input variables
- m_i, σ_i : The center and width of the bell-shaped function of the i^{th} input variable, respectively.

This equation can be implemented on a Forward Neural Network (FNN). This connectionist model combines the approximate reasoning of fuzzy logic into a five layer neural network structure [4].
Based on the error back propagation algorithm for multi-input single-output (MISO) system, the goal is to determine a fuzzy logic system $f(\underline{x})$ in the form of eq. (1), which minimizes the error function:

$$\mathrm{E(k)} = \frac{1}{2}(\mathrm{f}(\mathrm{x}(\mathrm{k})) - \mathrm{d}(\mathrm{k}))^2 \quad (2)$$

where
d(k) is the desired output at time k.
According to equation (1), if the number of rules is M, then the problem becomes training the parameters y_j, m_{ij}, and σ_{ij} such that E(k) is minimized. And based on the back propagation training algorithm the iterative equations for training the parameters y_j, m_{ij}, and σ_{ij} are:

$$y_j(k+1) = y_j(k) - \eta\left(f(\underline{x}(k)) - d(k)\right)\frac{1}{D} z_j \quad (3)$$

$$m_{ij}(k+1) = m_{ij}(k) - 2\eta \frac{z_j}{D}\left(f(\underline{x}(k)) - d(k)\right) \cdot \left(y_j(k) - f(\underline{x}(k))\right) \cdot \left(\frac{x_i^2(k) - m_{ij}}{(\sigma_{ij})^2} \right) \quad (4)$$

$$\sigma_{ij}(k+1)=\sigma_{ij}(k)-2\eta\frac{z_j}{D}\left(f(\underline{x}(k))-d(k)\right)\cdot\left(y_j(k)-f(\underline{x}(k))\right)\cdot\left(\frac{(x_i^2(k)-m_{ij})^2}{(\sigma_{ij})^3}\right) \tag{5}$$

where

$$z_j=\prod_{i=1}^{n}\exp\left[-\left(\frac{x_i^2-m_{ij}}{\sigma_{ij}}\right)^2\right] \tag{6}$$

D: The denominator of equation (1).
η : The learning rate.
Equations (3), (4), and (5) perform an error back propagation procedure.

III. Handling the Data Rate Difference using AFS

As mentioned in previous sections, this paper uses AFS to solve the difference in data rate between GPS and INS, i.e. to predict the GPS data at intermediate times.

Adaptive fuzzy system that developed to extrapolate GPS data for different rate systems shown in figure (1), where the proposed system have 2 inputs (one input for trajectory and the other input for the time instant), and one output (that predicted value of the GPS data).

Each component of position and velocity has its own network. To start the training, the networks need to be initialized with the number of epochs, the value of the learning rate, the number of fuzzy rules (M), and the parameters (m, y, and σ). These initial values are selected by trial-and-error. Appropriate selection of the initial values ensures good performance of the networks and converging to a minimum error value.

In this paper, the training phase was carried out after initializing all position and velocity networks with learning rate = 0.6, number of rules (M) = 40, m = [-1, 1], y = [-2, 2], σ = [0.01, 3.01], number of epochs = 1000. Utilizing the learned parameters (m, y, and σ), the following two strategies were used to extrapolate the GPS data.

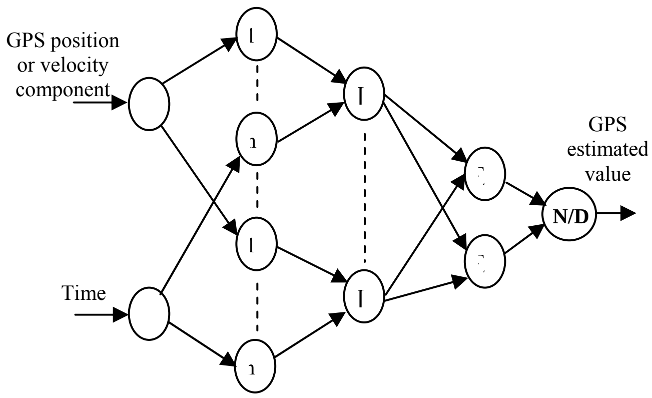

Fig.1. The Architecture of an AFS network for each component of position and velocity.

A. Extrapolation based on the two preceding seconds

In this implementation, the extrapolation is made based on the data taken from the true trajectory at the two preceding seconds according to the following pseudo-code:

```
Initialization
Load true trajectory (1 to n samples; let
n=600)
Load learned parameters m, y, and σ
Specify time (T; let T = 60 sec.)
P1 = true trajectory from 1 to 581 (step 10)
P2 = true trajectory from 11 to 591 (step 10)
Time1 = 1 to 58
Time2 = 2 to 59
Time3 = 2 to 60 (step 0.1)
N = number of inputs (let N = 5)
M = number of rules
Extrapolation process
begin
k = 0
for loop = 1 to 58
X (1) = P1 (loop); X (2) = P2 (loop)
X (3) = Time1 (loop) X (4) = Time2 (loop)
 for jj = 0.1 to 0.9 (step 0.1)
  k = k+1
  X (5) = Time2 (loop)+ jj (Time3 = Time2+jj)
  for j = 1 to M
   for i = 1 to N
    forward computation
   end
  end
  Compute Y_model (k)
 end
k = k+1
Y_model (k) = P2 (loop+1)
end
End
```

B. Extrapolation based on the two preceding extrapolated samples

The extrapolation is made based on the two preceding extrapolated samples according to the following pseudo-code:

```
Initialization
Load true trajectory (1 to n samples; let n =
600)
Load learned parameters m, y, and σ
Specify time (T; let T = 60 sec.)
P1 = true trajectory from 10 to 590
P2 = true trajectory from 11 to 591
Time1 = 1.9 to 58.9 (step 0.1)
Time2 = 2 to 59 (step 0.1)
Time3 = 2.1 to 59.1 (step 0.1)
N = number of inputs (let N = 5)
M = number of rules
Extrapolation process
begin
for loop = 1 to 570
X (1) = P1 (loop); X (2) = P2 (loop)
X (3) = Time1 (loop); X (4) = Time2 (loop)
X (5) = Time2 (loop)+0.1(Time3 = Time2+0.1)
for j = 1 to M
 for i = 1 to N
  forward computation
 end
end
Compute Y_model (loop)
P1 (loop+1) = P2 (loop)
P2 (loop+1) = Y_model (loop)
end
End
```

Table I shows the results obtained from the two strategies. Figures (2) and (3) compare between the true trajectory and the extrapolated trajectory resulted from implementing the two strategies for all components.

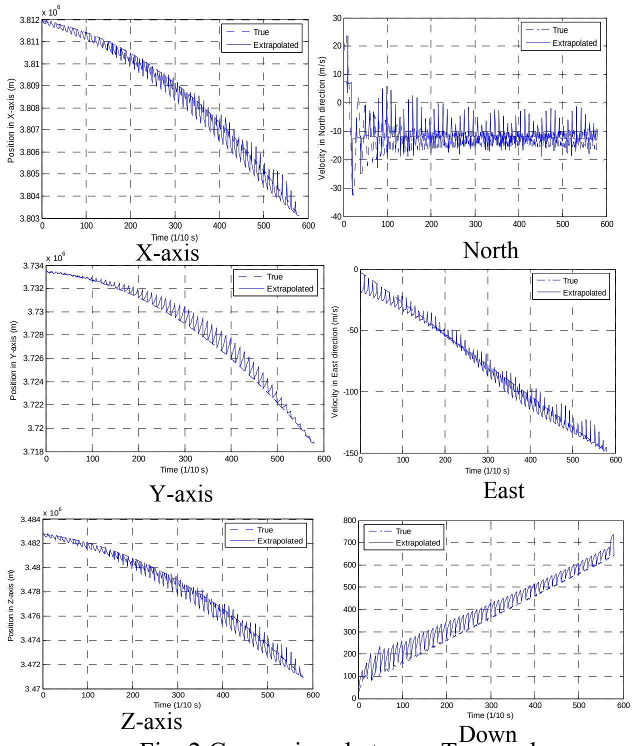

Fig. 2.Comparison between True and Extrapolated trajectory obtained from strategy

IV. Conclusions

From the results, the following conclusions can be drawn:

1. The AFS gives the solution to the problem of difference in sampling rates in a short time interval.
2. In general, strategy B produces better results, in terms of the STD and mean values, than the other one since it uses the data of the nearest samples to the extrapolated one.
3. Both AFS and ANN require prior knowledge of the trajectory. To solve this problem a database must be built for the selected trajectories to be used. However, the ANN gives better results than the AFS; on the other hand, the AFS has an advantage over Levenberg-Marquardt algorithm, used in [3], in terms of the capacity required in memory to implement the programs.

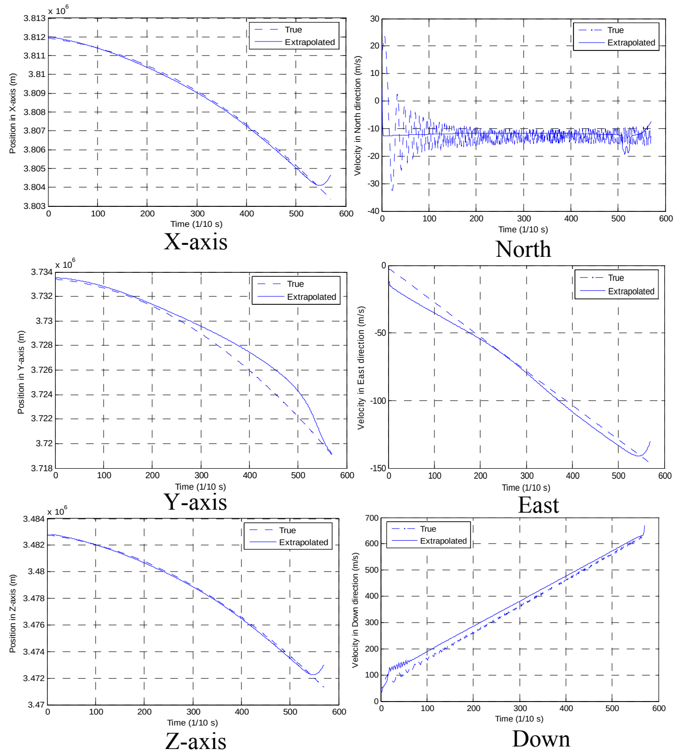

Fig.3. Comparison between True and Extrapolated trajectory obtained from strategy B for all

References

[1] Mayhew D. M., "Multi-rate sensor Fusion for GPS Navigation using Kalman Filtering", M.Sc. Thesis. Virginia Polytechnic Institute and state University, Electrical Engineering Dept., 1999.

[2] Ismaeel S. A., "Design of Kalman Filter for Augmenting GPS to INS Systems", Ph.D. Thesis. Computer Engineering Dept., College of Engineering, Al-Nahrain University, 2003.

[3] Salam A. Ismaeel and Ahmed M. Hassan, “Handling the Data Rate Difference between INS and GPS System,” International Arab Conference on Information Technology (ACIT’2006: Dec 19-21, 2006, Yarmouk University, Jordan), 2006.

[4] Salam A. Ismaeel and K. M. Al-Jebory, “Adaptive Fuzzy System Modeling,” Eng. Technology, vol. 20, no. 4, pp. 201-212, 2001.

TABLE I

Performance of Extrapolation using AFS

		Position			Velocity		
		X-axis	Y-axis	Z-axis	North	East	Down
Strategy A	MSE (m)	9.6207e-006	1.1786e-005	1.2299e-005	0.9082	0.0023	0.0049
	STD (m)	274.2244	430.1082	388.4952	5.3266	4.9506	25.0437
	Mean (m)	230.9971	-369.3932	323.8015	-1.4358	3.2623	-47.4992
	Elapsed Time (s)	0.4220	0.4220	0.4220	0.4370	0.4380	0.4060
	Prediction Time (s)	7.2759e-004	7.2759e-004	7.2759e-004	7.5345e-004	7.5517e-004	7.0000e-004
Strategy B	MSE (m)	9.6207e-006	1.1786e-005	1.2299e-005	0.9082	0.0023	0.0049
	STD (m)	161.9012	721.8531	209.9449	6.1647	4.0788	9.8063
	Mean (m)	34.5334	-757.7204	41.7194	-0.1635	4.2271	-19.4626
	Elapsed Time (s)	0.4370	0.4220	0.5320	0.4220	0.4220	0.4220
	Prediction Time (s)	7.6667e-004	7.4035e-004	9.3333e-004	7.4035e-004	7.4035e-004	7.4035e-004