Prediction of Errors and Improvement of Position Accuracy on Low Cost GPS Receiver with MLP Neural Network

M.R.Mosavi
Department of Electrical Engineering
Shahid Rajaee Teachers Training University
Lavizan, Tehran 16788, Iran
M.Mosavi@srttu.edu

K.Mohammadi
Department of Electrical Engineering
Iran University of Science and Technology
Narmak, Tehran 16844, Iran
Mohammadi@iust.ac.ir

M.H.Refan
Department of Electrical Engineering
Shahid Rajaee Teachers Training University
Lavizan, Tehran 16788, Iran
Refan@srttu.edu

M.Farrokhi
Department of Electrical Engineering
Iran University of Science and Technology
Narmak, Tehran 16844, Iran
Farrokhi@iust.ac.ir

Abstract: This paper presents a method to determine an accurate position by using a low cost GPS receiver and proposes a MLP neural network for better accuracy in GPS positioning. At first, we introduce the GPS system errors. Then measuring the errors of position components, real and dynamic patterns of the errors will be created and feed in to the neural network. The neural network is trained with such real data to predict the errors of later seconds. The steps design and implementation of the neural network are presented and the experimental results of the tests are stated with real data. These show position components errors decrease due to training of the neural network.

Key Words: Improve of position accuracy, Low cost GPS receiver, MLP Neural network

1.Introduction

Global Positioning System (GPS) has replaced prior positioning systems. It can cover all the earth by satellites to measure time, altitude, longitude and latitude in every desirable point [1,2].
Positioning began from 1950s and improved in 1970s. In 1980s, GPS became an operational positioning system. At first it was designed and used for military purposes. But later commercial applications have been increased. Nowadays commercial receivers take a great part in its market. GPS

receivers are used not only in navigation but also in topography, digital transmission systems and power networks [1,2].
The presented errors of GPS system can be categorized in three groups: **satellite errors, observer errors** and **receiver errors** [1,2].

- **Satellite errors:** The errors of satellite include mainly clock and orbit errors. Daily control of clock decreases its errors. Increasing the number of satellite and appropriate geometric configuration decreases the orbital errors. The related factor to this item is Dilution Of Position (DOP). The smaller is the factor, the less are errors.
- **Receiver errors:** It includes clock error, insufficient channels, limitation of registers length and so on.
- **Ionosphere and troposphere errors:** Passing through the atmospheric layers, cause a delay in propagation of transmitted signals from satellite. Since the thickness and density of layers are not stable, the delay in signals is variable and causes an error. This delay is proportional to $1/f^2$ so that a two frequency receiver can determine the errors to a great deal and compensate for them. Ionospheric delay is the most important error in GPS system. Table 1 shows the average errors introduced per satellite of GPS system in meter.

Table1. The average errors introduced of GPS system in meter [3,4]

Error Source	**Average (meters)**	**Time Constant**
Receiver Noise	0.4	-
Troposphere	0.5	More Than 1 Hour
Signal Multi Path	0.6	0.5 to 10 Minute
Satellite Clocks	1.5	-
Orbit Errors	2.5	More Than 1 Hour
Ionosphere	5	More Than 1 Hour
Selective Availability	30	$\approx$ 2 Minute

Because of above mentioned error sources, all GPS collected of points have a certain number of errors. This means that received data from GPS receiver will not reflect the real location. Therefore, users who wish to increase the accuracy of their GPS receiver must take steps to minimize the errors.
In this paper, an intelligent method to decrease the position errors in low cost GPS receiver is described. The theoretical background for better accuracy is based on the principle of neural network prediction scheme.

2.The Receiver and Data Collection

To achieve information of position and implementing an operational system, a low cost GPS engine manufactured by Rockwell Company was used. This miniature receiver with a small volume is appropriate for a vast range of Original Equipment Manufacturer (OEM) products. OEM receiver provides the possibility of improving software by presenting raw data [5,6].
This receiver has 5 parallel channels. It can track up to 9 satellites simultaneously. This receiver supports approved and improved NMEA-0183 protocol. It can receive differential RTCM messages to improve the accuracy of positioning in differential mode. It's serial port can receive and transmit NMEA or binary data with the rate of 4800 or 9600 bit per second. The binary protocol provides more detailed information compare with NMEA protocol [5,6].
To study the function of receiver, the GPS receiver was installed and set up in a fixed position. There are several binary messages provided by Micro tracker Low Power (MLP). One famous and general purpose of these messages is message No.103, which is available on the first output port as default, when we configure the receiver in binary mode [5,6].
In order to setup the receiver, data collection and connecting to PC, a hardware designed and implemented. Fig.1 shows the hardware structure. The output data were collected for many months

and were saved by a Pentium III computer with 450MHZ speed.

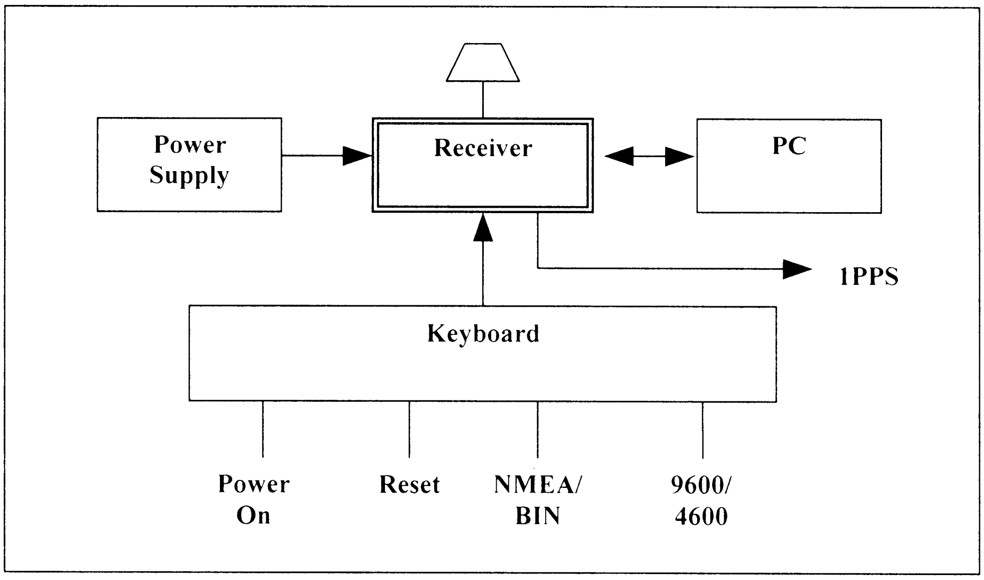

Fig.1. Hardware structure

3.Position Components Errors

Since MLP is a low cost nonmilitary receiver, its measurement errors are notable, too (60 Meter RMS 3D when S/A is off) [5,6]. To study the receiver data and achieving the errors, the data of position were studied in World Geodetic System–1984 (WGS–84). Therefore the x,y and z magnitude in the 103 binary message were collected and were saved in separate files with 1 second period.
We focus on variation of x,y and z position in studying neural network [7]. A software was composed for this purpose. By calculating the average of each quantity in file length, the software provides difference of the instantaneous magnitude of each point with its corresponding quantity average according to equations (1) to (6) [8,9] and saves them in other files.

$$Ax = (\sum_{i=1}^{i=n} x_i)/n \tag{1}$$

$$dx_i = x_i - Ax \tag{2}$$

$$Ay = (\sum_{i=1}^{i=n} y_i)/n \tag{3}$$

$$dy_i = y_i - Ay \tag{4}$$

$$Az = (\sum_{i=1}^{i=n} z_i)/n \tag{5}$$

$$dz_i = z_i - Az \tag{6}$$

Where x_i, y_i, z_i are instantaneous magnitude of x,y,z and Ax, Ay, Az are the average magnitude of x,y,z and dx_i, dy_i, dz_i are instantaneous error magnitude of x,y,z respectively. n is number of samples.
We developed softwares (xgraph, ygraph and zgraph) to drawing the dx , dy and dz graphs. A sample from data collection for almost 24 hours is shown in Fig.2.

4.Linear Correlation between Position Components

We collected position data for a fix point at 17 days and saved them to 17 files. Based on the equations (7) to (9), linear correlation between x,y and y,z and z,x for 17 data files were calculated. The obtained results of correlation coefficient were recorded in Table 2.

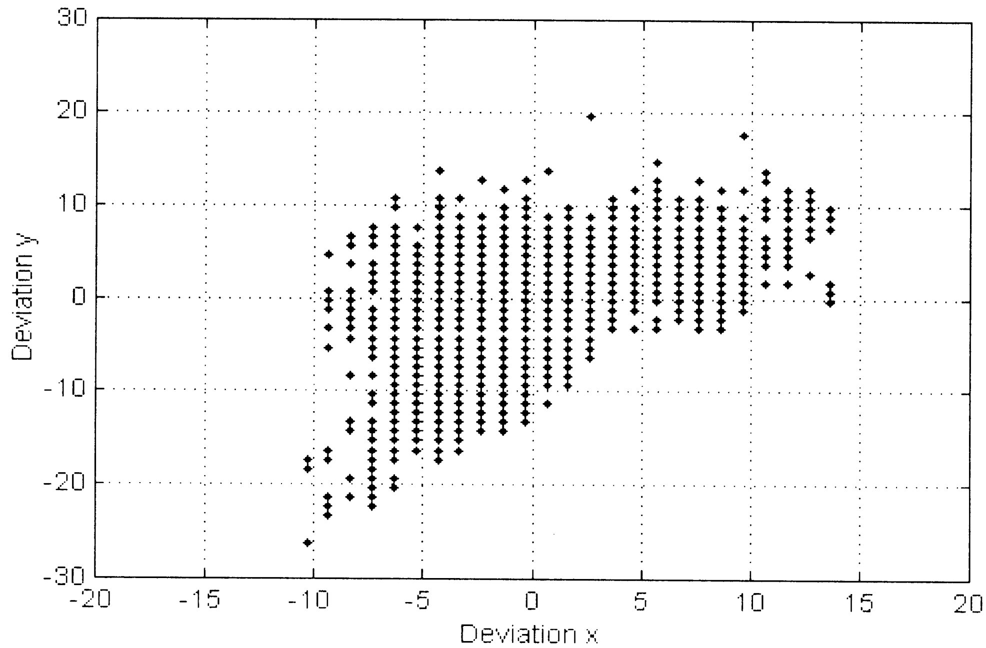

Fig.2. Graph of x,y and z errors for almost 24 hours

$$r_{xy} = [n \sum_{i=1}^{i=n} x_i y_i - (\sum_{i=1}^{i=n} x_i)(\sum_{i=1}^{i=n} y_i)] / \{[n \sum_{i=1}^{i=n} x_i^{\ 2} - (\sum_{i=1}^{i=n} x_i)^2][n \sum_{i=1}^{i=n} y_i^{\ 2} - (\sum_{i=1}^{i=n} y_i)^2]\}^{0.5} \quad (7)$$

$$r_{xz} = [n \sum_{i=1}^{i=n} x_i z_i - (\sum_{i=1}^{i=n} x_i)(\sum_{i=1}^{i=n} z_i)] / \{[n \sum_{i=1}^{i=n} x_i^{\ 2} - (\sum_{i=1}^{i=n} x_i)^2][n \sum_{i=1}^{i=n} z_i^{\ 2} - (\sum_{i=1}^{i=n} z_i)^2]\}^{0.5} \quad (8)$$

$$r_{yz} = [n \sum_{i=1}^{i=n} z_i y_i - (\sum_{i=1}^{i=n} z_i)(\sum_{i=1}^{i=n} y_i)] / \{[n \sum_{i=1}^{i=n} z_i^{\ 2} - (\sum_{i=1}^{i=n} z_i)^2][n \sum_{i=1}^{i=n} y_i^{\ 2} - (\sum_{i=1}^{i=n} y_i)^2]\}^{0.5} \quad (9)$$

Where r_{xy}, r_{xz} and r_{yz} are linear correlation between x,y and y,z and z,x , respectively.

Table 2. x,y,z correlation coefficients for 17 days

File name	Rxy	Rxz	Ryz
Data1.bin	0.54	0.43	0.46
Data2.bin	0.62	0.41	0.50
Data3.bin	0.63	0.47	0.58
Data4.bin	0.58	0.42	0.52
Data5.bin	0.57	0.41	0.55
Data6.bin	0.63	0.41	0.47
Data7.bin	0.57	0.34	0.47
Data8.bin	0.60	0.43	0.61
Data9.bin	0.57	0.46	0.47
Data10.bin	0.58	0.39	0.55
Data11.bin	0.51	0.42	0.51
Data12.bin	0.63	0.44	0.52
Data13.bin	0.48	0.43	0.52
Data14.bin	0.55	0.50	0.49
Data15.bin	0.59	0.44	0.54
Data16.bin	0.56	0.40	0.43
Data17.bin	0.61	0.35	0.42
Average Correlation Coefficients	0.58	0.42	0.51

The results of these statistical studying show that the average correlation coefficients are: $r_{xy} = 0.58$, $r_{xz} = 0.42$ and $r_{yz} = 0.51$. Hence, the results show linear correlation between x,y and y,z and x,z.

5.Neural Network Prediction

Patterns of positioning errors (dx,dy and dz) are fed into neural network in this special application. This neural network is trained by such patterns. It find the ability to predict later errors [10,11,12]. Because of linear correlation between x,y and y,z and also x,z , the neural network should be of such kind that their input variables include patterns of positioning errors conjointly. Hence the structure of this neural network is such in a way that input variables include conjoint patterns of positioning errors (due to linear correlation x,y and y,z and x,z) and output variables predict errors of x component ($\hat{d}x(n+1)$), of y component ($\hat{d}y(n+1)$) and of z component ($\hat{d}z(n+1)$). The structure of this topology is shown in Fig.3.

In forming of the neural network with P equal 60 (one-minute patterns), we considered 180 inputs. The first 60 were due to x component errors (dx). The second 60 were due to y component errors (dy) and the third 60 were due to z component errors (dz). In fact the first 60 inputs show x component errors, during dx(n-60) to dx(n). The second 60 inputs show y component errors, during dy(n-60) to dy(n). The third 60 inputs show z component errors, during dz(n–60) to dz(n). The outputs of the neural network predict position component errors in $\hat{d}x(n+1)$, $\hat{d}y(n+1)$ and $\hat{d}z(n+1)$. The number of the hidden layer neurons obtains in terms of the experience equation [7]:

$$N \ge \log_2^T \tag{10}$$

Where, N is the number of the hidden layer neurons and T is the number of training samples. The number of total samples in this paper was considered 1000. Thus, $N = \log_2^{1000}$, that is ten hidden layer neurons are suitable as is shown in Fig.3.

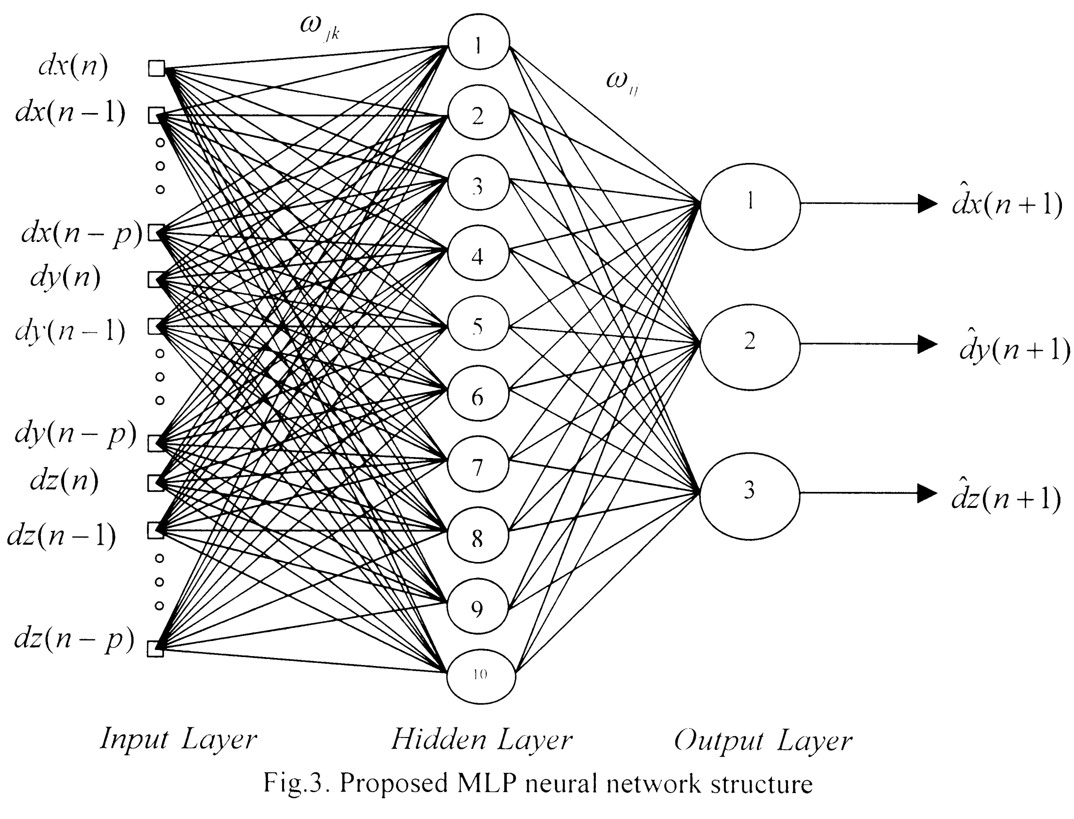

Fig.3. Proposed MLP neural network structure

To train this neural network, the patterns are determined by single transfer from a 60 chosen window for each pattern of positioning errors dx, dy and dz.
Back-Propagation (BP) algorithm is used to train of the neural network. The decision function is a nonlinear sigmoid function, that is:

$$\sigma(x) = 1/(1+e^{-x}) \tag{11}$$

5.1.The Forward Calculation Process

The purpose of training neural networks is to determine the weights of between layers. In other words, training occurs by arranging the weights and threshold magnitudes repeatedly. After each repetition, networks become more aware of its surrounding. Learning can be under supervision, without it or conjointly with it (a part under supervision and a part without it) [13,14].
The forward calculation process is calculating output state of every layer neuron so that real output of network will be obtained. The calculating equations are as follows:
Input of the j-th hidden layer neurons, V_j:

$$V_j = \sum_{k=1}^{k=180} w_{jk}x_k - \theta_j \tag{12}$$

Output of the j-th hidden layer neurons, O_j:

$$O_j = \sigma(V_j) = \sigma(\sum_{k=1}^{k=180} w_{jk}x_k - \theta_j) \tag{13}$$

Input of the i-th output layer neurons, V_i:

$$V_i = \sum_{j=1}^{j=10} w_{ij}O_j - \theta_i \tag{14}$$

Output of the i-th output layer neurons, O_i:

$$O_i = \sigma(V_i) = \sigma(\sum_{j=1}^{j=10} w_{ij}O_j - \theta_i) \tag{15}$$

Where w_{jk} is the weight from input layer to hidden layer, w_{ij} is the weight from hidden layer to output layer, x_k is input value of input layer, θ_j and θ_i are separately thresholds value of hidden layer and output layer neurons.

5.2.The Learning Process

The weights of the network are adjusted by following equations:
Adjusted value of the weight input layer to hidden layer, Δw_{jk}:

$$\Delta w_{jk} = \eta.\delta_j.x_k = \eta.o_j.(1-o_j).\sum_{i=1}^{i=3} \delta_i w_{ij}.x_k \tag{16}$$

Adjusted value of the weight from hidden layer to output layer, Δw_{ij}:

$$\Delta w_{ij} = \eta.\delta_i.o_j = \eta.e_i.o_i.(1-o_i).o_j = \eta.(d_i - o_i).o_i.(1-o_i).o_j \tag{17}$$

Adjusted value of hidden threshold, $\Delta\theta_j$:

$$\Delta\theta_j = -\eta.\delta_j = -\eta.o_j.(1-o_j).\sum_{i=1}^{i=3} \delta_i w_{ij} \tag{18}$$

Adjusted value of output threshold, $\Delta\theta_i$:

$$\Delta\theta_i = -\eta.\delta_i = -\eta.e_i.o_i.(1-o_i) = -\eta.(d_i - o_i).o_i.(1-o_i) \tag{19}$$

Where, η is factor of learning.

6.Experimental Results

After training of the neural network, to study the efficiency of trained neural network, we used 3000 test data that were collected on the building of Computer Control and Fuzzy Logic Research Lab in the Iran University of Science and Technology. Their statistical significance results from this test data are shown in Table 3.

Table 3. Average, Variance and Standard Deviation of positioning error components

Statistical Significance Characteristic	X Position Component	Y Position Component	Z Position Component
Error Average [m]	-1.2036	1.1433	1.2553
Error Variance $[m^2]$	7.1037	10.3422	8.2826
Error standard deviation [m]	2.6652	3.2159	2.8779

According to Table3, the error average in predicting for x is -1.2036 [m], for y is 1.1433 [m] and for z is 1.2553 [m]. Error variance in predicting for x is $s_x^2 = 7.1037\ [m^2]$, for y is $s_y^2 = 10.3422\ [m^2]$ and for z is $s_z^2 = 8.2826\ [m^2]$. Also according to Table3, standard deviations of errors in predicting for position components of x,y and z are orderly: $s_x = 2.6652\ [m]$, $s_y = 3.2159\ [m]$ and $s_z = 2.8779\ [m]$, which shows decreasing of errors due to training neural network.

7.Conclusions

This paper has described how the positioning accuracy of a low cost GPS receiver can be greatly improved with a MLP neural network. The neural network was trained to predict the errors of later seconds according to the measurement or information available. The validity of the proposed MLP neural network was confirmed by experimental results on implemented unit (Fig.4) in this paper. The results show that the position component errors decrease due to training of the neural network. So that position measurement errors were reduced to less than 2 meters, while it was about 60 meters before neural network prediction.

References:

[1] Paul Zarchan, "Global Positioning System: Theory and Applications", Volume I, American Institute of Aeronautics and Astronautics, 1996.

[2] B.Hofmann-Wellenhof, H.Lichtenegger and J.Collins, "Global Positioning System: Theory and Practice", Third Revised Edition, Springer-Verlag Wien New York, April 1994.

[3] Sien-Chong Wu and William G. Melbourne, "An Optimal GPS Data Processing Technique for Precise Positioning", IEEE Transactions on Geoscience and Remote Sensing, Vol. 31, No. 1, Jan. 1993, pp.146-152.

[4] M.H.Refan and K.Mohammadi, "Point Averaging of the Position Components, before and after S/A IS Turned off", The Asian GPS Conference 2001, India, pp.53-58.

[5] "Microtracker LP Operations Manual", Rockwell International Corporation, GPS-16, January 10, 1994.

[6] "Microtracker LP Designer's Guide", Rockwell International Corporation, GPS-22, January 1, 1995.

[7] Xian-Jun Gao and Yi-Song Dai Ke Wang, "A Study on the Self-Difference GPS Positioning by Dynamic and Fictitious Datum Station", IEEE Conference on Vehicle Electronics, Vol.1, 1999, pp.16-18.

[8] M.R.Mosavi, K.Mohammadi and M.H.Refan, "Fuzzy Processing on GPS Data to Improve

Positioning Accuracy, before and after S/A Is Turned off", The Asian GPS Conference, October 2002, India, pp.117-120.

[9] M.R.Mosavi, K.Mohammadi and M.H.Refan, "Time Varying ARMA Processing on Low Cost GPS Receiver Data to Improve the Position Accuracy", The Asian GPS Conference, October 2002, India, pp.125-128.

[10] Jizhang Sang, Kurt Kubik and Lianggang Zhang, "Prediction of DGPS Corrections with Neural Networks", First International Conference on Knowledge-Based Intelligent Electronic Systems, May 1997, Australia, pp.355-361.

[11] Surender K.Gupta, Foong Sui Jin and Khoo Aik Leong, "An Intelligent Small Area DGPS", IEEE Conference on Position, Location and Navigation, 1996, pp.169-173.

[12] S.K.Gupta, M.K.Low and C.W.Tan, "A New Approach to Simulate GPS Measurements", IEEE Conference on Position, Location and Navigation, 1998, pp.236-242.

[13] Simon Haykin, "Neural Network", Macmillon, 1994.

[14] Thomas H.Kerr, "Critique of Some Neural Network Architectures and Claims for Control and Estimation", IEEE Transactions on Aerospace and Electronic Systems, Vol.34, No.2, April 1998, pp.406-419.

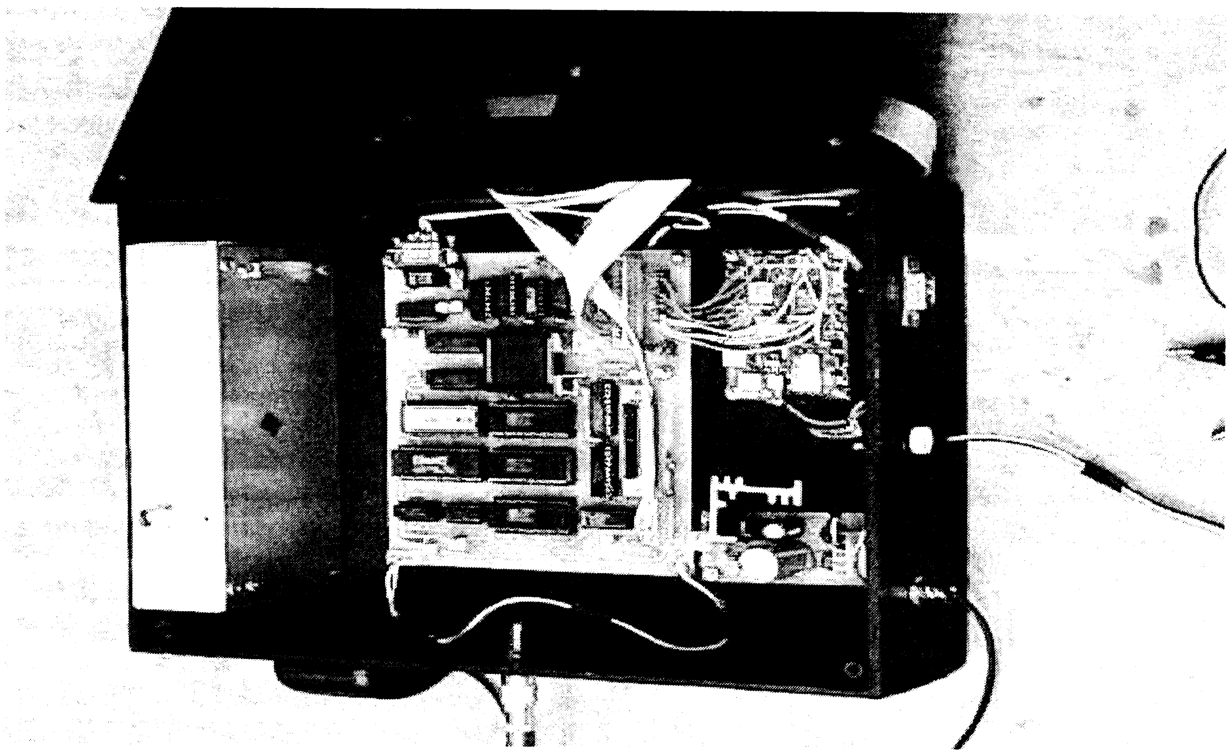

Fig.4. Implemented unit in this paper