A Software Implementation of Multi-Sensor Data Analysis and GPS Integrity Assessment for Real-Time Monitoring Applications

Clement Ogaja
School of Surveying and Spatial Information Systems,
The University of New South Wales, Sydney, NSW 2052, Australia
E-mail: c.ogaja@student.unsw.edu.au

Abstract. Some recent developments in structural monitoring schemes have discussed the complementary benefits of integrating GPS positioning with other sensors. This leads to the challenge of having to deal with potentially vast amounts of data, requiring heavy computational and communication loads. Efficient data reduction techniques therefore become necessary tools for handling the large amounts of data generated. These techniques can be easily implemented in projects where the processing software runs on a centrally located control station PC, receiving real-time streams of data, and computing precise positions for all stations located on the target structure. In such schemes, integrity assessment also becomes an important task if the objective is to determine the actual response signature of the structure. This problem can be addressed by using statistical tools in which one or several parameters may change abruptly. In the first instance, biases or outliers due to sensor malfunction should be detected as soon as possible, and then the sensor isolated (and ultimately repaired) in order to guard against incorrect (or biased) deformation signals.

The system reported here has been developed for the purpose of achieving real-time computational efficiency, time-savings, and integrity assessment of GPS coordinate results. The solution, which has been provided in a simple software implementation, is described and examples of its operation are given.

Key words: Multi-sensor, integrity assessment, real-time monitoring, RTK-GPS

1 Introduction

Recent advances in Global Positioning System (GPS) processing technology have led to a new generation of software products that are capable of providing automated 3D position information in real-time. Such technology is now used in nearly all sectors requiring high precision positioning. The system described in this paper is aimed at supporting those applications concerned with the monitoring of engineering structures in view of the following requirements:

- Some deformation studies of large civil structures (Wong et al, 2001; Roberts et al, 2000) have indicated that the monitoring schemes become more heavily instrumented with multiple GPS units, in some instances integrated with additional sensors [eg. Accelerometers, Anemometers]. This leads to the challenge of having to deal with additional data, requiring more computational load and communication bandwidth. Data reduction techniques may be applied to handle the large amounts of data generated.
- In a deformation monitoring scheme, the system failure or malfunctioning can be caused either by satellite or GPS sub-system failures, or by GPS receiver failures, or the system can simply be plagued with long-term drift problems associated with most non-GPS

sensors. The system operators ought to have, in addition to the actual deformation parameters, real-time or near real-time knowledge concerning the system health and status.

The proposed system can address the two requirements mentioned above, assuming only the GPS receiver faults in the latter case. The basis of this system is to analyse streams of data from GPS receivers and other sensors located on the target structure. The data is assumed to have been transferred via modem, wireless radio or network connection to a centralised personal computer running the real-time processing software. The data streams are processed within the PC using the data reduction and integrity assessment algorithms specifically tailored for real-time operation.

In this article, the proposed software system has been implemented and tested off-line using data stored in the local hard disk of the PC. The reader should therefore note that some challenges involved in the actual interfacing with the GPS processing engines are yet to be addressed. Such challenges include online data access, data rate synchronisation and improving the overall system user-friendliness.

2 Outline of the Proposed System

The general flow of the developed system is as shown in Figure 1. Its main objective is to analyse continuous data streams and to compare outputs with specified criteria in order to extract deformation signals as well as to characterise the behaviour of the monitoring system or sub-system.

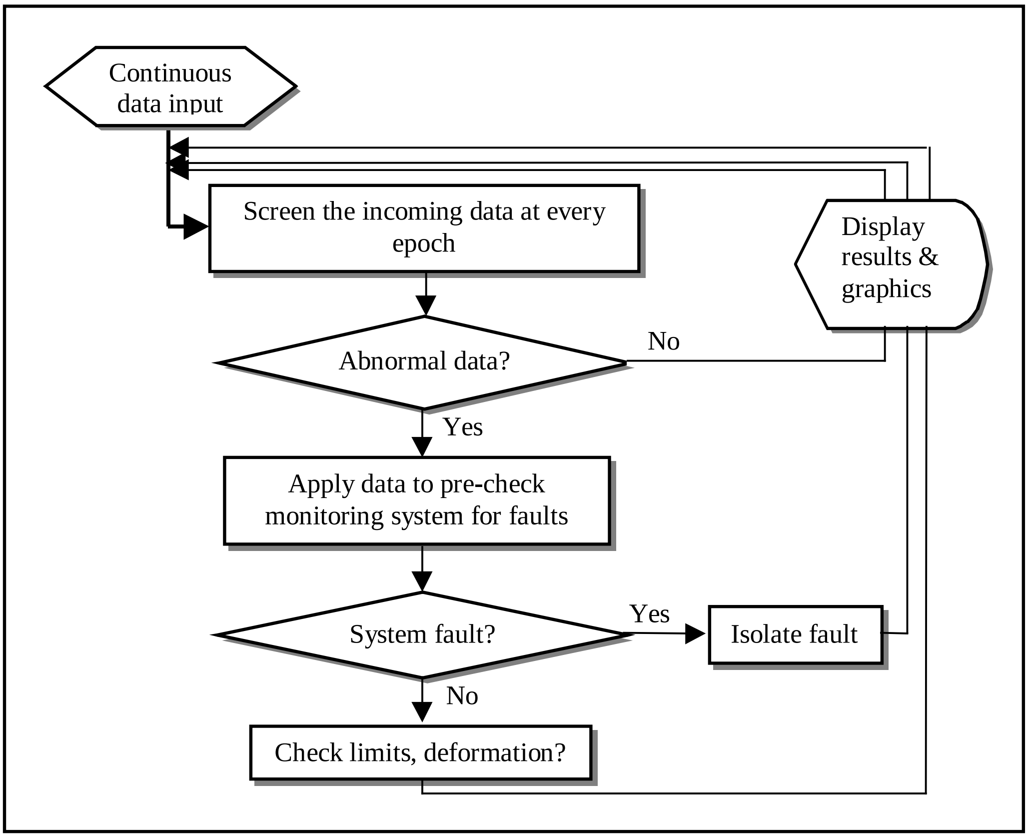

Fig.1: The Concept of a Real-Time Monitoring System

Two main software modules have been designed to implement the mathematical algorithms of GPS data integrity assessment and multi-sensor data reduction. Both modules include functions for online analysis that detect abnormal observations as quickly as possible, and either provide clues on the possible cause (e.g. sensor faults), or simply issue an alert if a warning limit is exceeded. The tasks that cannot be carried out online can then be handled by off-line analysis. The validity of the analysed results and the efficiency of the monitoring system are both crucial.

In the following sections the mathematical principles of multi-sensor data reduction and GPS data integrity assessment are outlined, followed by a brief description of the two system modules that have been developed.

2.1 The Mathematical Principles

Multi-Sensor Data Reduction Algorithm

In measurement systems where many sensor types are used to collect information with a common objective, the resulting data would have some degree of correlation if measured simultaneously. In such cases, a representative statistic can then be used for a quick overview of the system status. The use of the Q-statistic (see e.g., Ogaja et al, 2001) is considered in this article. For its implementation, the reference means vector $\overline{X}$ and the reference covariance matrix S of all the variables should be known apriori. If some "clean" historical data are available, these parameters can be estimated.

The following procedure outlines the computation process for the Q-statistic:

- estimate the reference covariance matrix:

$$S_m = \frac{1}{m-1}\sum_{i=1}^{m}(X_i - \overline{X}_m)(X_i - \overline{X}_m)^{\mathrm{T}} \tag{1}$$

where the vector X_i is the i^{th} epoch observation of p variables from m reference samples and $\overline{X}_i$ is the mean vector containing the means of each variable. S_m is a $p \times p$ matrix. If the reference data are not available apriori, the reference parameters can be computed online when the process is at a "clean" start-up stage. In that case, define a running sample means vector:

$$\overline{X}_n = \overline{X}_{n-1} + \frac{\left(X_n - \overline{X}_{n-1}\right)}{n} \tag{2}$$

of the first n observations $\{X_1, X_2, ..., X_n\}$, to be updated sequentially and in real-time.

- using the estimated reference covariance matrix (Eq. 1), the measured variables are continuously transformed into the Q-statistic according to:

$$Q_f = (X_f - \overline{X}_n)^{\mathrm{T}} S_m^{-1} (X_f - \overline{X}_n) \tag{3}$$

where X_f denotes the p-dimensional vector of future observations on the p variables.

GPS Data Integrity Assessment Algorithm

The principle of GPS data integrity assessment will be explained using the concept shown in Figure 2. The basic idea is that a combination of three GPS 'rover' stations can be used to monitor deformation as well as to monitor GPS system faults, assuming that only one receiver can fail at any one time.

With the aid of Figure 2, let dX_0, dY_0, dZ_0 represent the reference or the mean baseline vector solution, and dX_t, dY_t, dZ_t represent the instantaneous baseline vector solution at time t.

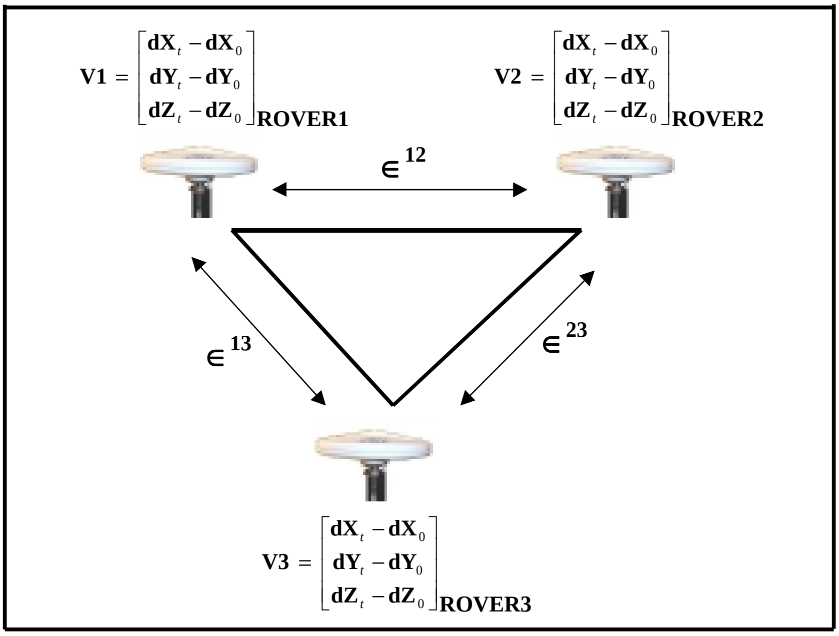

Fig. 2: GPS Measurement Model

Use all the combinations of receiver pairs to generate between-receiver differences of the residuals. Thus:

$$\in^{12};\in^{13};\in^{23} = [\mathbf{V1} - \mathbf{V2};\ \mathbf{V1} - \mathbf{V3};\ \mathbf{V2} - \mathbf{V3}] \tag{4}$$

If a persistent bias is present in the measurements of one of the 'rovers', then it would also exist in two of the between-receiver differenced residuals. Each component of these residuals is tested for a change-point using the cumulative sum scheme (Figure 3):

$$\begin{aligned} S_t^{*} &= \left(S_{t-1}^{*} + \in_t - \frac{\bar{\delta}}{2} \right)^{+} \\ S_t^{\bullet} &= \left(S_{t-1}^{\bullet} - \in_t - \frac{\bar{\delta}}{2} \right)^{+} \\ S_0^{\bullet} &= S_0^{*} = 0 \\ \bar{\delta} &= |\mu_1 - \mu_0| \end{aligned} \tag{5}$$

where $\bar{\delta} > 0$ is the assumed change size (tuning parameter), μ_0 is the initial mean and μ_1 is the mean after change. For a more detailed explanation of this scheme the reader can refer to any standard literature on quality control charts (see e.g. Basseville & Nikiforov, 1993). The related work with this scheme for online shift detection in GPS coordinate results can also be found in Mertikas (2001).

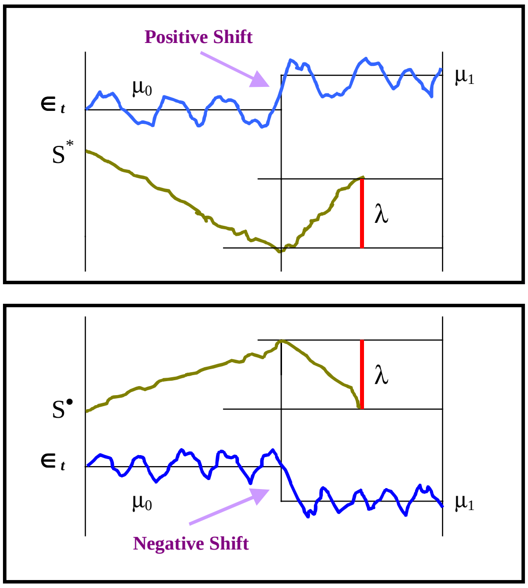

Fig. 3: Cumulative Sum Scheme

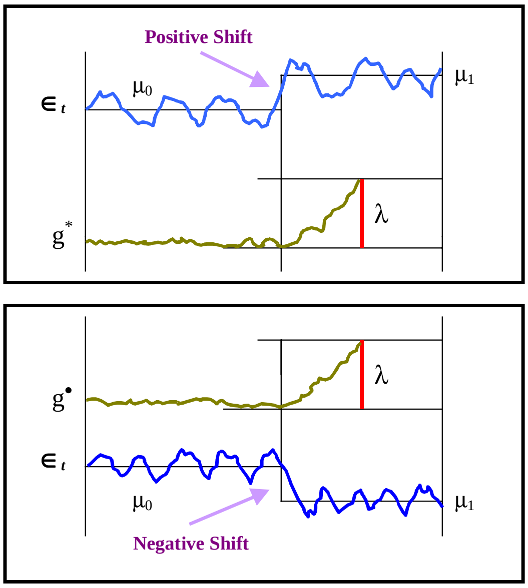

Fig. 4: Modified Cumulative Sum Scheme

Next, transform the cumulative sums of Figure 3 into two parallel convenient detectors (Figure 4) which may be described as follows: detect a change in the mean at the first time *n* and issue

$$\mathit{alarm}\ \text{if} \begin{cases} g_n^* = S_n^* - \min\limits_{0 \le t \le n} S_t^* \ge \lambda \\ g_n^\bullet = \max\limits_{0 \le t \le n} S_t^\bullet - S_n^\bullet \ge \lambda \end{cases} \tag{6}$$

Note, however, that the transformations in Eq. 6 are only necessary if the generated functions are to be plotted in real-time, as is the case in this article.

If two change-points are detected simultaneously by Eq. 6, then the isolation of the faulty receiver is realised by majority voting. The receiver that is common to the two between-receiver residuals with the change-points is the faulty receiver.

To obtain the residuals in Eq. 4, the mean or reference of a selected baseline vector component [i.e. dX_0 or dY_0 or dZ_0] common to all the receivers (Figure 4) is estimated in real-time by determining the running mean using Eq. 2 (see also Mertikas, 2001, for an example). The running mean is denoted μ_0 before a change is declared and μ_1 after a change is detected (Figs. 3 & 4). A threshold λ is tuned for minimum detection delay and a desired rate of false alarms. An output error which is lower than the minimum level of $\bar{\delta}$ does not lead to a sustained increase in the cumulative sum and hence an alarm does not occur. A reset interval is also defined to reset the algorithm in the case of detection.

2.2 The Software Modules

For the two main tasks described earlier, a software package has been designed under the Windows operating system, with a graphical-user-interface (GUI). The module is named "Real-Time System Monitor" (RTSM). The interface is designed to provide both visual and graphical information on the system status. By using a framework of Visual C++ 6.0 classes for real-time plotting, the two algorithms described in the previous section have been implemented in C++ programming code. In the present article, two separate modules of the RTSM are discussed.

RTSM Module I implements the integrity assessment algorithm as well as performing additional functions at the same time. The flow of the process involved in this module is as shown in Figure 5.

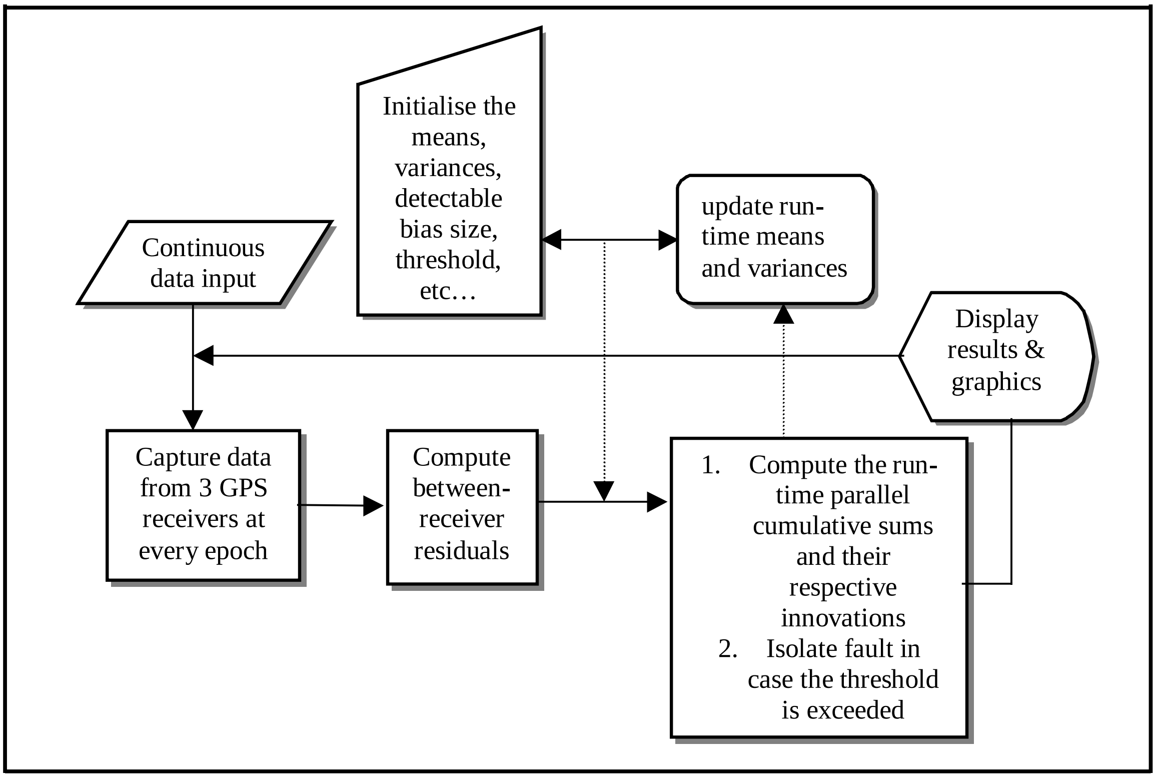

Fig.5: The Concept of Module I of RTSM

The main functions of Module I of RTSM include:

- To access the positioning residuals for multiple receivers from a GPS processing engine and/or auxiliary sensor data, and to perform statistical calculations in real-time.
- To check the consistency between multiple 'rover' receivers in real-time, assuming there is an independent check on the GPS base station(s).
- To display the data analysis results graphically in real-time.
- To perform alarm handling in case of abnormal observations or system sensor faults.

The central goal is to allow any Windows PC to run the software system without user intervention except when an alert condition is reached. With the help of object-oriented programming, the various modules have been carefully structured to optimise memory usage and to permit easy code re-use. The implemented algorithms are robust against both positive and negative observational or system biases.

RTSM Module II combines both the integrity assessment algorithm and the data reduction algorithm. The concept of this module is illustrated in Figure 6. Its main functions include:

- To access both GPS data and auxiliary multi-sensor data continuously, and to perform statistical calculations in real-time.
- To perform online multi-sensor data reduction.
- To check the consistency between multiple 'rover' receivers in real-time, assuming there is an independent check on the GPS base station(s).
- To display the data analysis results graphically in real-time.
- To perform alarm handling in case of abnormal observations or system sensor faults.

Since the two modules share some common functions, a dialogue has been designed to switch between them in order to avoid code duplication.

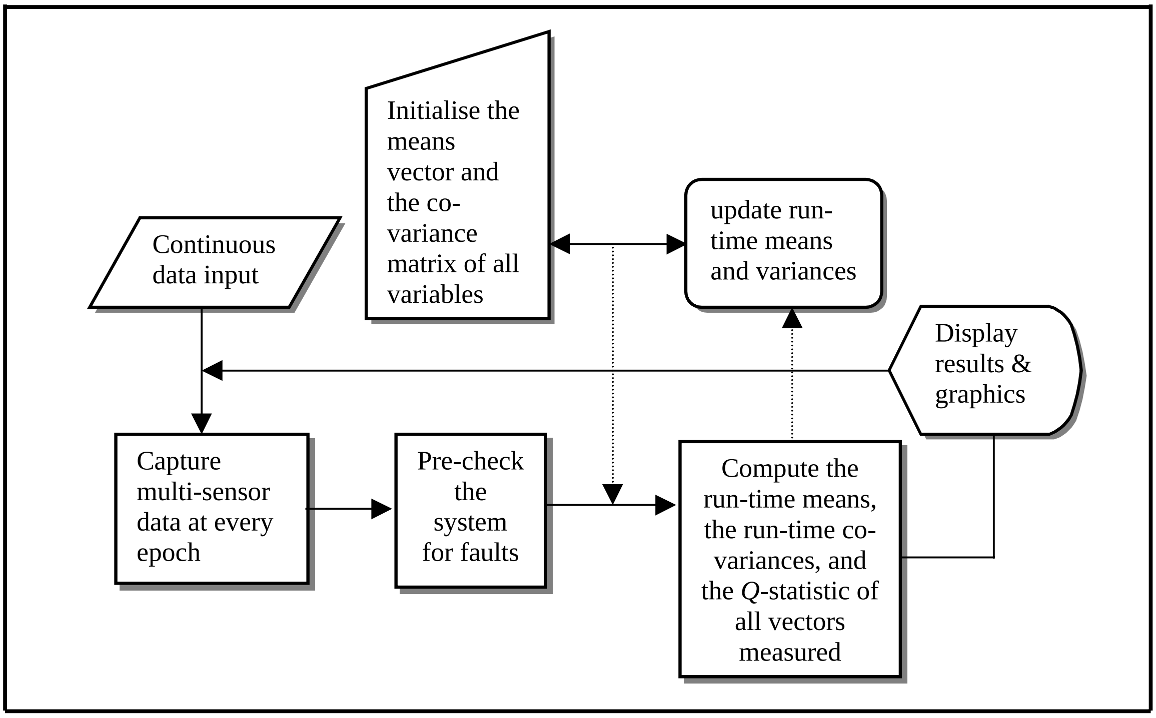

Fig.6: The Concept of Module II of RTSM

3 Examples

The implemented system has been tested to verify its performance. Two examples are discussed.

Example1 – Bias Simulation and Detection

In the first example, a test has been carried out using data collected on the rooftop of the Electrical Engineering Building of the University of New South Wales (UNSW) on 26 November 2001. Figure 7 shows the experimental set up in which a total of four dual-frequency NovAtel GPS receivers were used to collect data at 1 second epoch rate for more than twenty minutes. One receiver was designated as the base station, while the remaining three were the rover stations. The data was collected using specially developed UNSW software (RTFour). All four receivers were connected to a single PC via 4 RS232 serial cables joined together by a PCMCIA card. The data from the three rovers and the base station were logged simultaneously in binary format and post-processed using baseline software, also developed at UNSW. For the purpose of testing the RTSM, a 20-minute span of the baseline residuals for all the rovers were saved to the local harddisk (Figure 8).

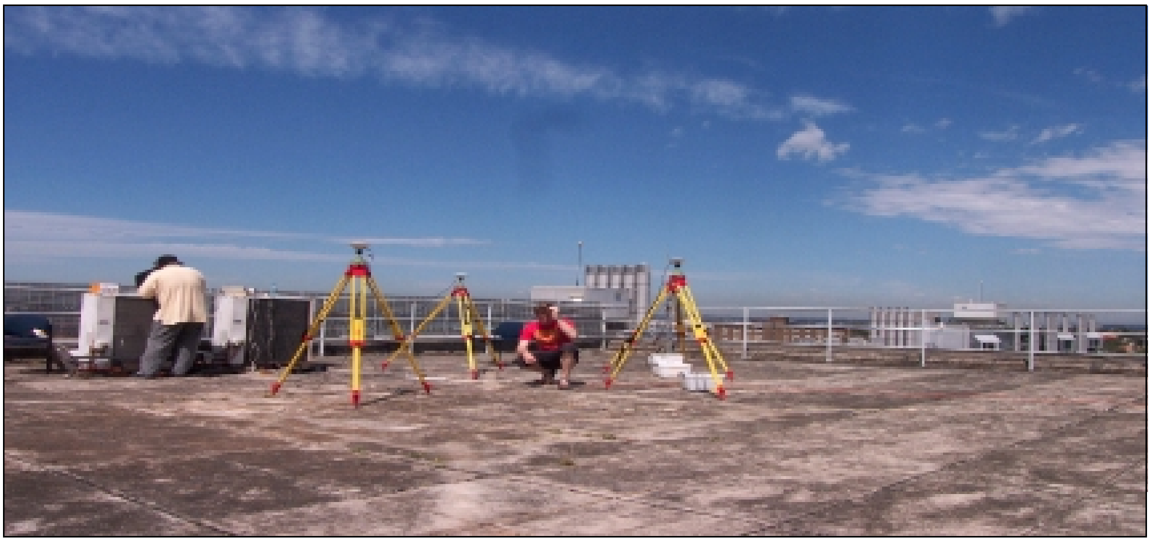

Fig. 7 Experimental Set Up at UNSW (26 November 2001)

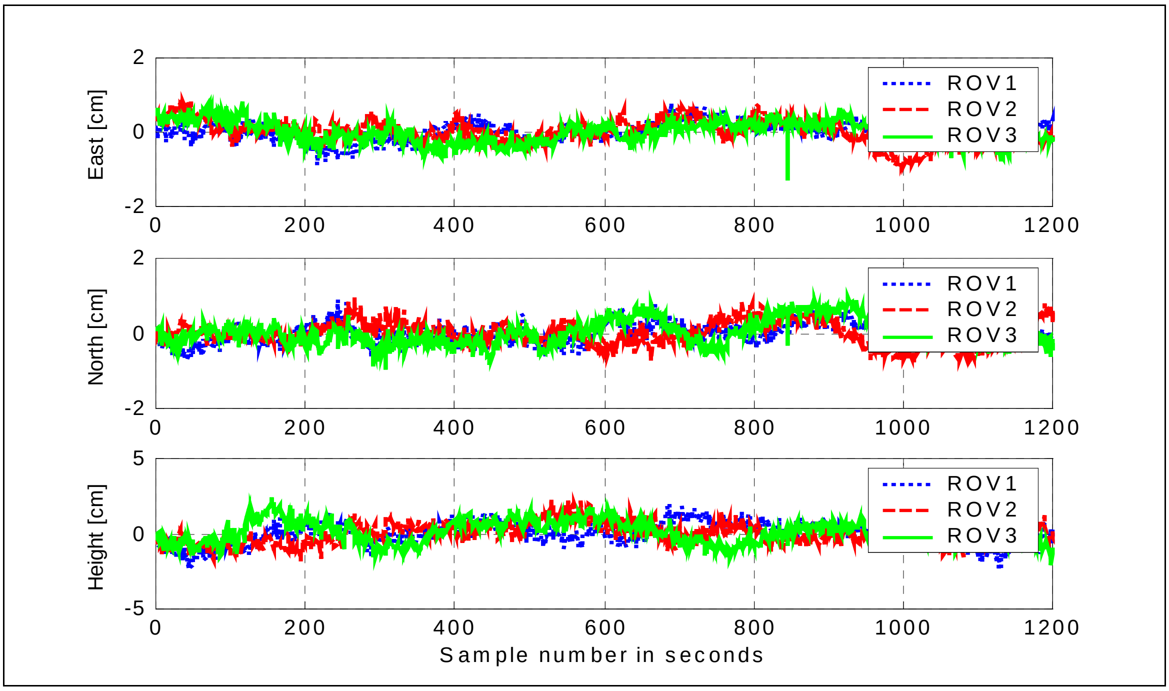

Fig. 8 Time series of coordinate solutions (UNSW – 26 November 2001)

A bias of constant magnitude was introduced into the residuals of each of the vector components obtained from one of the rovers, identified in this example as ROVER1. The standard deviations of each of its components were first computed and the bias introduced to each component of the order of two standard deviations, with a known onset time. Thus from the standard deviation values of ±0.26cm, ±0.25 and ±0.72cm, the quantities +0.52cm, –0.50cm and –1.44cm were introduced as biases to the easting, northing and height components of ROVER1. The positive and negative signs of the biases have been assigned arbitrarily. Figure 9 is the RTSM interface of the residuals. This interface emulates the online process in which the bias is detected and ROVER1 is isolated as the responsible sensor (as shown by the red flag and the triggered message box).

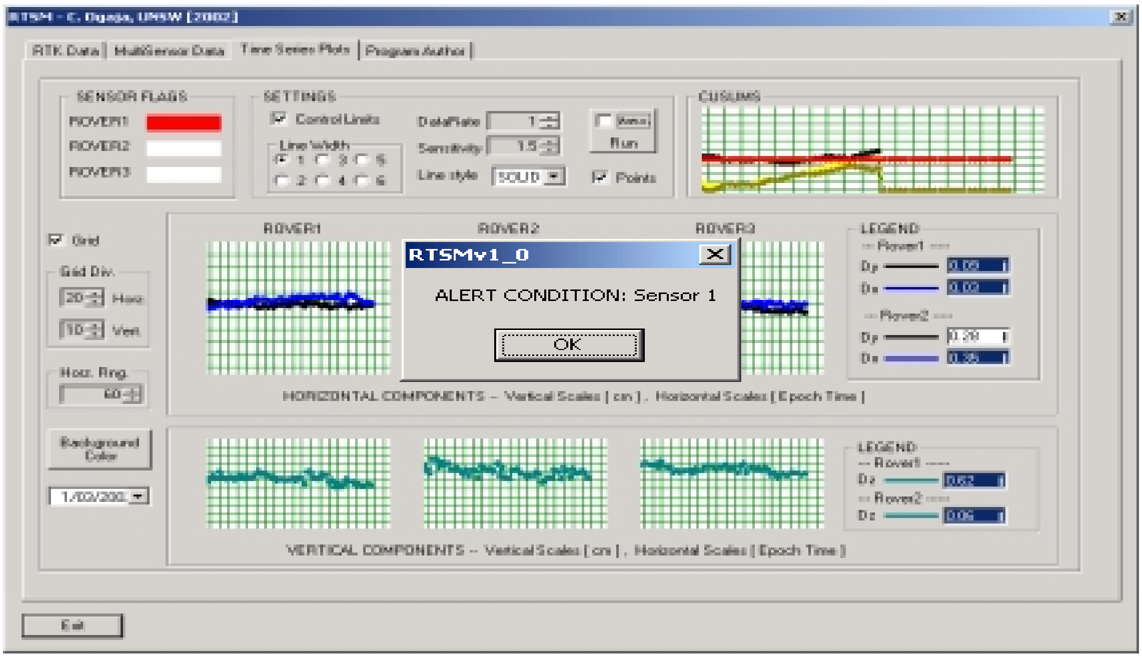

Fig. 9 Screen Image of integrity test

Example 2 – Multi-Sensor Data

A second example is a test of the RTSM as applied to GPS and auxiliary sensor data. In this case the data used were collected using GPS receivers, accelerometers, anemometers and a temperature sensor. The experiment was carried out on the rooftop of a tall building in Singapore during the month of February 2000 (Figure 10). Notice the two anomalous responses for wind and acceleration. Figure 11 is the RTSM display of the *Q*-statistic variable that represents all of the data in a multivariate sense. This figure shows the two out-of-control events detected after they exceeded the multivariate warning limit.

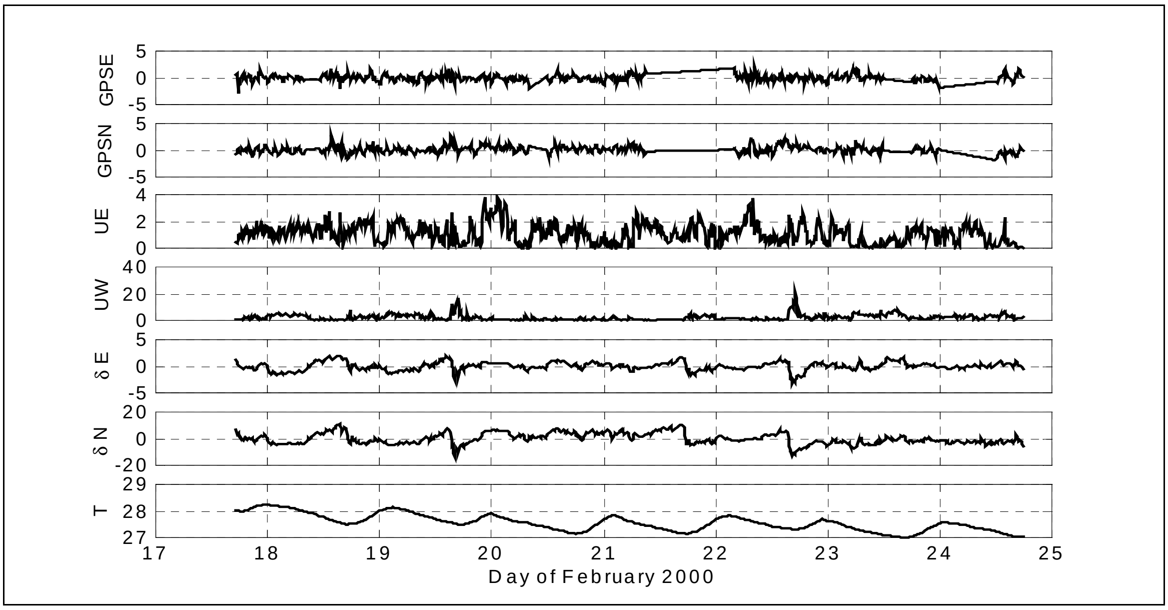

Fig. 10 Simultaneous GPS positioning, wind, accelerometer and temperature sensor data. Units: GPSN [cm], GPSE [cm], UE [m/s], UW [m/s], δE [mm/s^2], δN [mm/s^2], T [^{o}C].

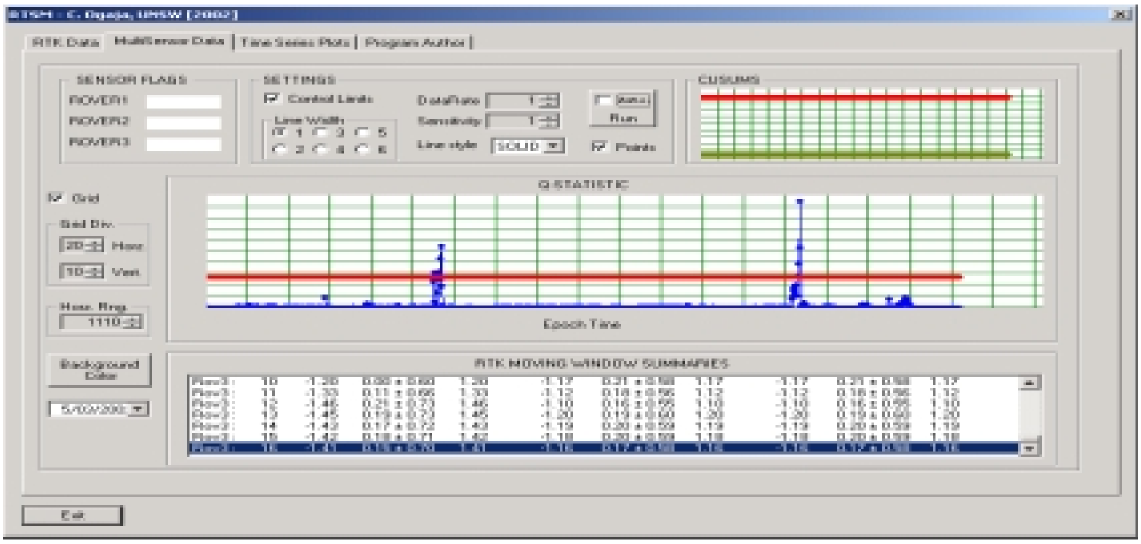

Fig.11: Screen image of multi-sensor signal

4 Concluding Remarks

Experience with the simple software system described here has shown that it is can be used to apply multi-sensor data reduction and consistency check of a GPS measurement system in real-time. It can support studies in which a centrally located computer is used as a shared resource serving multiple purposes in addition to the real-time processing of raw GPS data collected from multiple receivers. It is well suited for long-term real-time monitoring by GPS in which positions are not needed at the remote GPS sites, and where powerful processing engine is resident on the PC. Further development of this system is underway.

Acknowledgements

The author wishes to acknowledge the support of his supervisors, Prof. Chris Rizos and Dr. Jinling Wang, during this study. He is supported in his PhD studies by the International Postgraduate Research Scholarship (IPRS) funded by the University of New South Wales (UNSW). Thanks to Mike Moore and Jun Zhang, both of the Satellite Navigation and Positioning (SNAP) group of School of Surveying and Spatial Information Systems at the UNSW, for their collaborative efforts in the experiment of 26 November 2001.

References

Basseville M. & V. Nikiforov. *Detection of Abrupt Changes – Theory and Applications*. New Jersey, USA: Prentice Hall, 1993. 441pp.

Mertikas S. Automatic and on-line detect-ion of small but persistent shifts in GPS station coordinates and statistical process control. *GPS Solutions*, 2001, 5(1), 39-50.

Ogaja C., J. Wang, C. Rizos & J. Brownjohn (2001). Multivariate monitoring with GPS observations and auxiliary multi-sensor data. *To be published in GPS Solutions*.

Roberts G.W., X. Meng & A.H. Dodson. Structural dynamic and deflection monitoring using integrated GPS and triaxial accelerometers. *13th Int. Tech. Meeting of the Satellite Division of the U.S. Inst. of Navigation GPS ION 2000*, Salt Lake City, Utah, 19-22 September, 2000, 59-68.

Wong K., K. Man & W. Chan. Monitoring Hong Kong's bridges: Real-Time Kinematic spans the gap. *GPS World*, 2001, 12(7), 10-18.