ESA/ESOC REAL TIME DATA PROCESSING

J. Pérez[(1)], L. Agrotis[(2)], J. Fernández[(1)], C. García[(1)], J. Dow[(3)]

(1) ESOC GMV, Robert-Bosch-Str. 5, D-64293 Darmstadt, Germany
(2) Symban Ltd, 22 Marshal's Drive. St Albans. Herts AL1 4RH, UK
(3) ESOC, Robert- Bosch-Str. 5, D-64293 Darmstadt, Germany

Abstract

ESOC has been operating most of the stations of the ESA/ESOC GPS tracking network in real time mode since 2003. So far, GPS measurements have been streamed from the remote sites to ESOC using internally developed tools and protocols. Data from some of the stations have been shared with the IGS Real Time Network since 2004. ESOC has also been involved in the development of a system for Real Time Navigation (RETINA). This is based on a sequential filter for the real time processing of the GPS and GLONASS observations and includes specially designed and built infrastructure elements for streaming the data, and archiving and retrieving data and results.

In 2006 the IGS UDPrelay software has been installed at ESOC to get data from the IGS network and also to deliver the complete ESA/ESOC network to the be used in the frame work of the IGS Real Time Working Group. An assessment of the strategy and results of the developed applications is made using the data from different sources including the current IGS Real Time Network.

Outline

The paper first describes the ESA/ESOC RT tracking network and the ESOC Real Time processing strategy. This is followed by a discussion of a sequential filter prototype for Real Time estimation of clock biases and ZTD[1], presenting two test cases with some preliminary results. The last part of the paper describes the design of a new ESOC system for REal TIme Navigation (RETINA) which is currently under development.

ESA/ESOC RT Tracking Network

ESOC has been supporting the IGS RT working group since 2004. There are currently 9 ESA sites[2] (see Figure 1) submitting 1Hz GPS data streams in Real Time and eventually the whole network[3] will be operating in RT.

The migration from the previous RT data management applications (RTGlocal & RTGremote, which had been developed by ESOC) to the new standard RT IGS software (Ashtechreader & UDPrelay) has been accomplished. Several problems showed up during the installation of these new packages because the original code,

[1] Zenith Tropospheric Delay
[2] KIRU, KOUR, MALI, MAS1, NNOR, PERT, VILL, REDU, CEBR
[3] The previous sites plus FAAA and KOU1

developed under LINUX and using GNU tools was not 100% compatible with the SUN environment at ESOC.

With the new software, the RT data streams cannot be directly routed from the remote sites but need to be transmitted to the control facilities at Darmstadt before the streams can be made publicly available. This is because the network shares resources with ESOC's operational network which has significant security constraints.

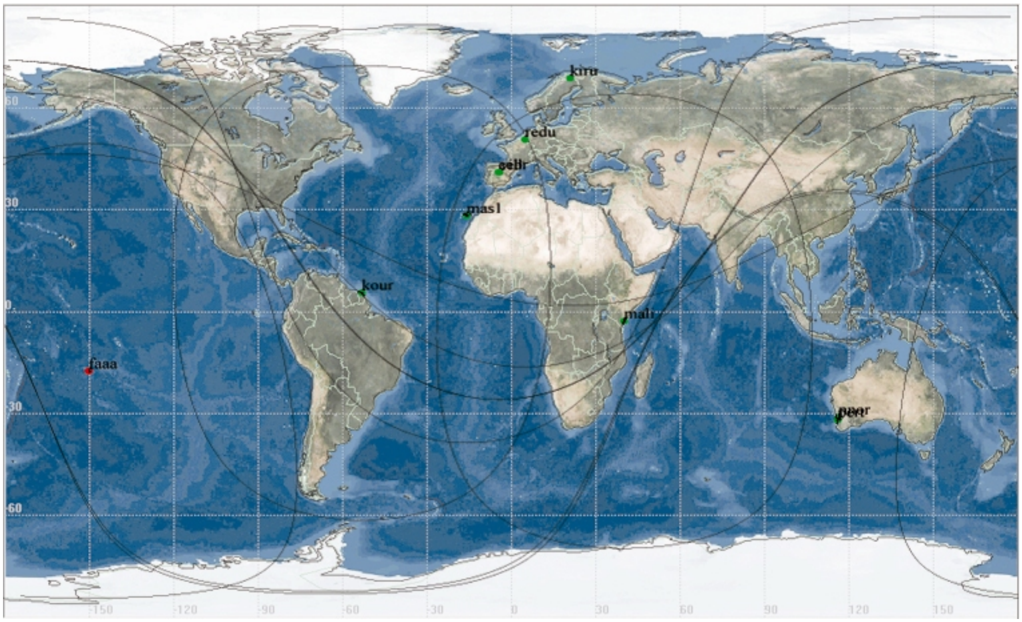

Figure 1.- ESA/ESOC RT GPS Tracking Network

ESOC Real Time Processing Strategy

The ESA/ESOC RT solution is based on a combined batch/sequential filtering approach (see Figure 2). Firstly, the orbits (predictions) and the precise geodetic coordinates are estimated by batch POD processes. The frequency of these batch runs is higher than the IGU (normally one run every hour). Then, those orbits and coordinates are used as inputs by the sequential clock bias estimator. Apart from the clock biases, the filter estimates tropospheric zenith delays for all the stations and phase ambiguities between each station-satellite pair.

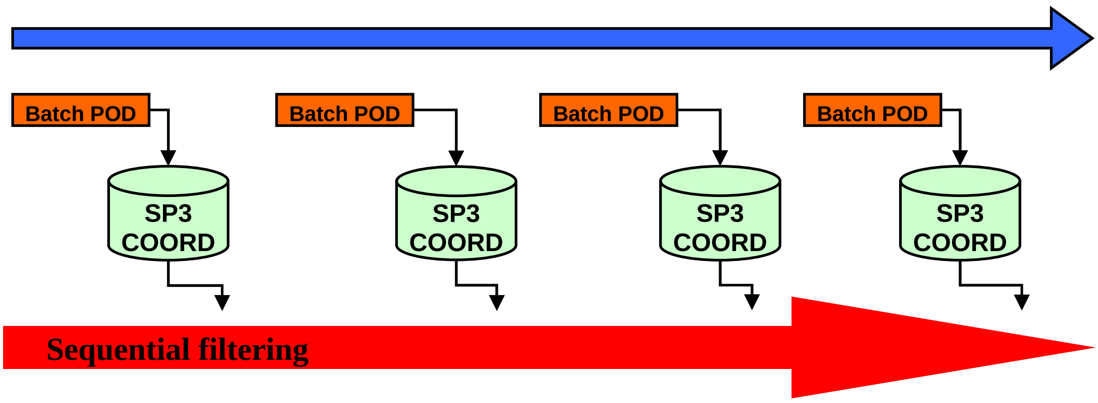

Figure 2.- Combined batchsequential filtering approach

This combined batch/sequential filtering is less complex than other schemes due to the fact that implementation of satellite dynamics models in the filter software is not needed. Also, the computation time is reduced because of using a priori ephemeris and coordinates. The number of parameters to be estimated every epoch is dominated by the need to estimate phase ambiguities, and this constrains the number of stations that can be used in the solution in order to keep up with the measurement frequency.

The s/c clock biases are the critical Real Time navigation products because unlike to the s/c orbits, which can be accurately propagated into the future over reasonably long periods, the clock biases cannot be predicted at the same level of accuracy[4].
An aditional benefit of the sequential GNSS data processing is that it is possible to monitor the GNSS SIS[5]. In addition to the navigation solution, users can be provided with up-to-date information about the status and availability of the constellation(s).

Sequential Filter for RT Estimation

The data is processed sequentially through the implementation of a Kalman filter, where the ionospheric free combinations of code and phase GNSS observables are processed. The filter has built-in algorithms for the detection of cycle slips and outliers in the data.

Due to the high CPU demand of the Kalman algorithm, some effort was spend on code optimisation in order to improve execution performance. This was partly achieved through the use of the open source “ATLAS” package for performing some of the low level algebra computations (http://math-atlas.sourceforge.net/). Improvements in execution performance will be an ongoing research activity.

The Kalman filter estimates the following parameters:

- The clock bias for all spacecrafts and stations in the system.
- An intersystem bias for every combined GPS/GLONASS receiver.
- A value of the zenith tropospheric delay for each site.
- The initial ambiguities for all the phase observations.

Two test cases are presented here to show the performance of the filter and to validate the approach.

Test Case 1

This first test case is an assessment of the maximum accuracy achievable by the current prototype where only the RT filter prototype is tested. The clock estimates are computed based on data from a globally distributed network of 52 IGS sites and using the ESA final orbits and coordinates. Finally, 30 seconds measurement sampling is used with an elevation cutoff of 10º. The test period used was 20/04/2006, (GPS week 1371, dow 4) and the solutions were compared with the ESA IGS Rapid and Final datasets. These are summarised in Table 1 and Figure 3 and Figure 4.

[4] IGS Urapid clk. vs. IGS Rapid clk ~3-6 ns
[5] Signal in Space

Comparison Dataset	**RT - IGR**	**ESA - IGR**
Overall RMS	0.240 ns	0.106 ns
Best Clock RMS	0.048 ns	0.025 ns

Table 1. Statistics for Test Case 1

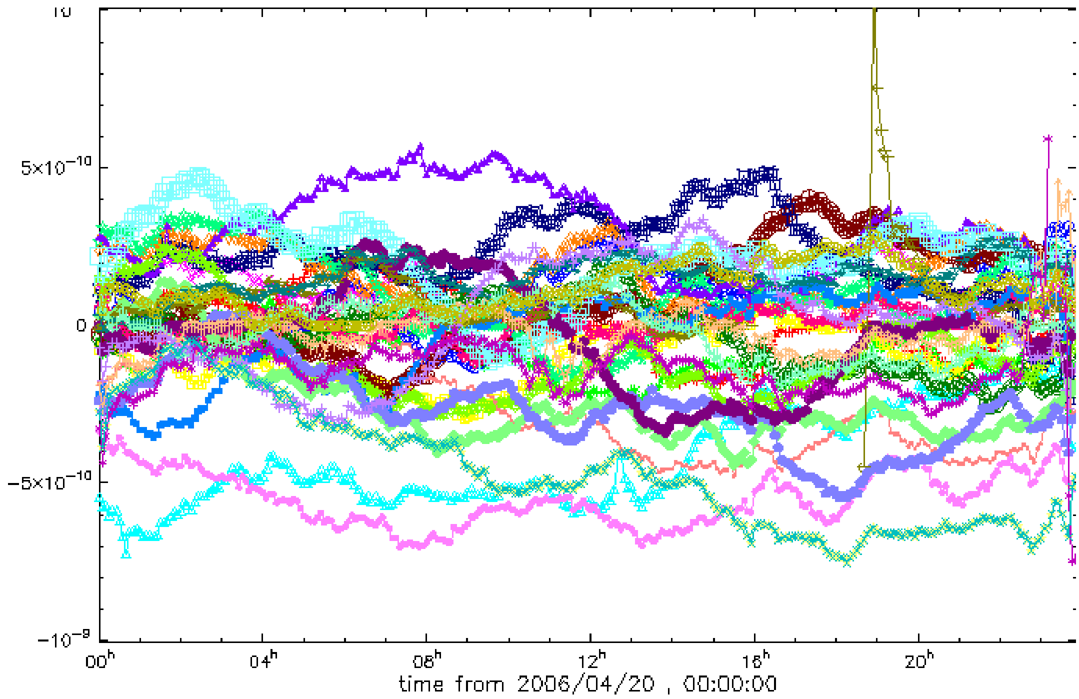

Figure 3. Filter vs. IGR s/c Clock Bias Comparison for 1371_4 (s)

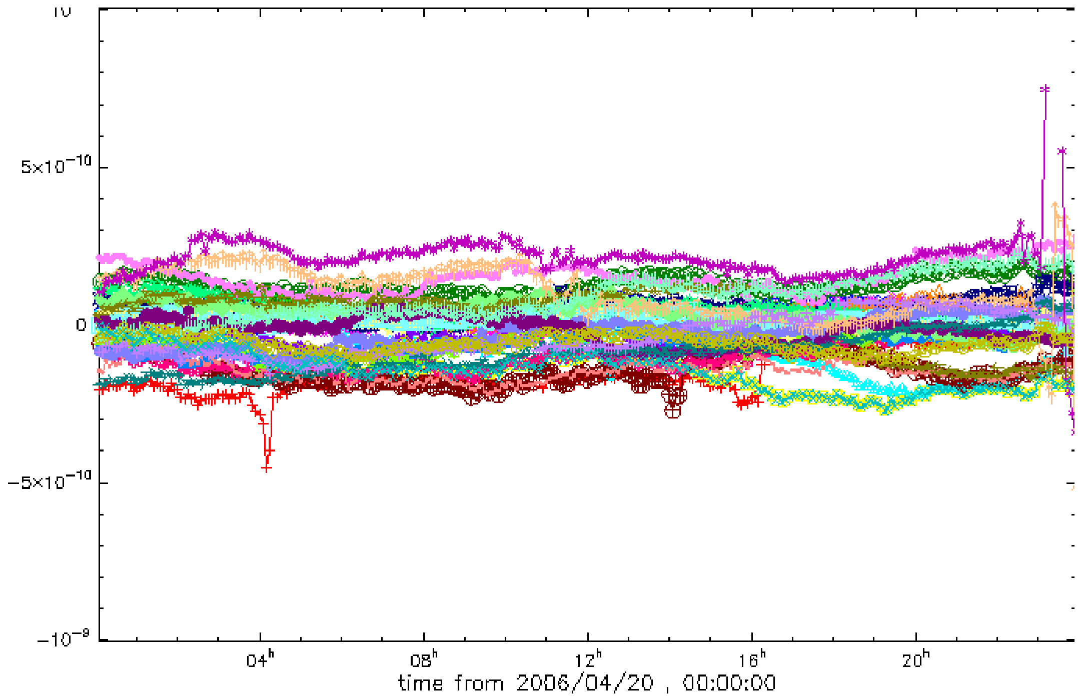

Figure 4. ESA Final vs. IGR s/c Clock Bias Comparison for 1371_4 (s)

Test Case 2

This test case is a Real Time emulation that shows the achievable performance in Real Time mode using currently available resources (Tracking network & software). The system was set up in the same way as in the planed operational environment.
The clock estimates were computed based on a tracking network of 32 IGS LEO sites (see Figure 5) and using frequent batch updates (orbits and coordinates were computed every hour). Measurement sampling was at a frequency of one second and an elevation cutoff of 5º was used, due to poor coverage in certain areas. The test period used is 24/03/2006, (GPS week 1367, dow 5).

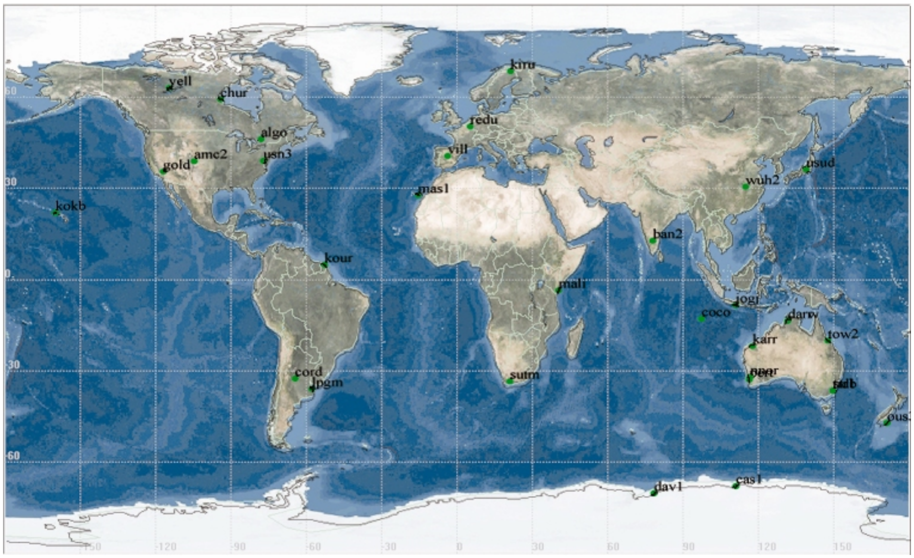

Figure 5. Test case 2 – Selected network

The following figures show comparisons with IGS final products. These are summarised in Table 2.

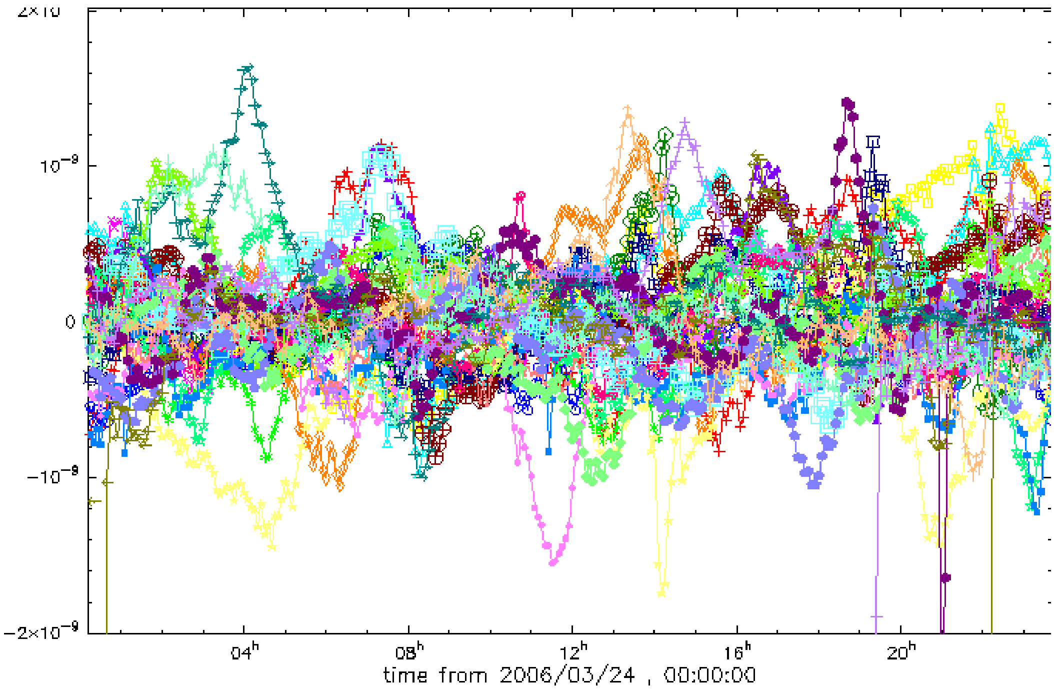

Figure 6. RT vs. IGS s/c Clock Bias Comparison for 1367_5 (s)

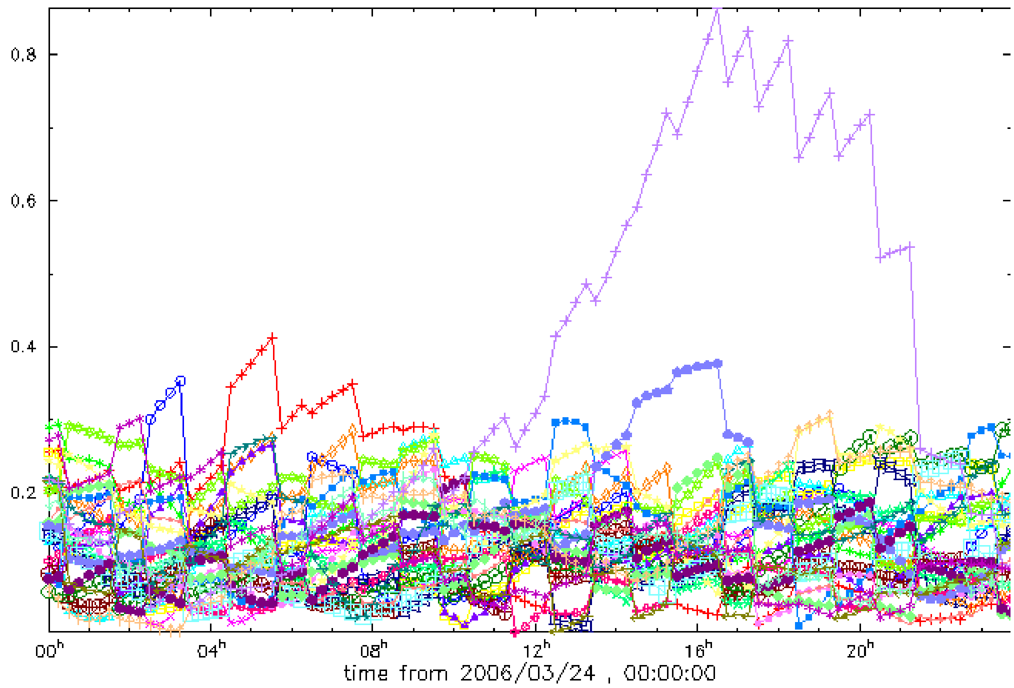

Figure 7. Propagated Orbits vs. IGS Orbits for 1367_5 (m)

Clock Statistics	Overall RMS:	0.424 ns
	Best Clock RMS:	0.199 ns
Orbit Statistics	Overall RMS:	9.7 cm
	Best s/c RMS:	5.4 cm

Table 2. Statistics for Test Case 2

PPP Results

Finally, a further assessment on the RT products quality based on static PPP was performed. The precise coordinates for site WTZR (Germany) have been computed with NAPEOS (new ESOC POD software) using the products (orbits and clocks) from the two test cases and the results have been compared with the coordinates computed using the ESA final products.

In test case 1, where the orbits were estimated (not predicted), the results are:

Table 3. Residuals in local system (mm)

Latitude	Longitude	Radial
6.0 N	-20.9 E	-31.7 U

In test case 2, where the orbits where predicted in the RT emulation, the results are:

Table 4. Residuals in local system (mm)

Latitude	Longitude	Radial
-7.7 N	-35.8 E	72.7 U

RETINA

RETINA stands for System for Real Time Navigation and it is a suite of algorithmic, infrastructure and visualisation elements to enable the processing, distribution and archiving of both Real Time and Near Real Time GNSS data and results. The RETINA system is currently being developed for ESOC by UK-based Symban Ltd.

The software has both generic and GNSS-specific elements. The objective of RETINA is to integrate batch and Real Time processing under a single system with centralised facilities for monitoring and control of the Real Time tasks and batch processes.

RETINA Real Time processing concepts and techniques borrow heavily from those used in Satellite Control systems where ESOC has many years of experience. In particular, GNSS data and results are treated in much the same way as spacecraft Telemetry Streams and application messages and alarms are displayed and acknowledged from a central console.

The RETINA subsystems belong to one of three functional categories:

Infrastructure Software is written in C++. The main components are middleware elements for History Filing and Event Logging and a Job Scheduling application. All middleware elements have C++, Java and FORTRAN interfaces.

Algorithmic Software is written in FORTRAN 90 and includes heritage FORTRAN77 code. It incorporates applications for Real Time and Batch data processing and estimation and for generation of products and comparison statistics between results sets.

Visualisation Software is entirely written in Java for portability. It includes Real Time graphical and alphanumeric display applications and the RETINA Graphical User Interface.

Infrastructure Software

Infrastructure software comprises middleware, server applications and utilities for:

- Filing and Archiving
- Event Logging and Alarm Management
- Job Scheduling

Filing and Archiving software is the main tool used for streaming and archiving of Real Time data. The software provides a middleware layer to the RETINA applications with API calls to support:

- Filing data to (time) key indexed History Files
- Update of existing HF records
- Retrieving ‘live’ (recent) data
- Retrieving archived (old) data (backward and forward directions)
- Derived retrieval by parameter name e.g. phase bias between spacecraft *x* and receiver *y*

History File Open and Access servers are responsible for relaying the data to external users via TCP, using the same API calls as local users who read directly from the History Files. An archiving demon runs in the background to automatically archive data from the 'live' History Files after a user-configurable time interval. Retrieval from archive and 'live' History Files appear seamless to the user.

The use of the Filing and Archiving infrastructure in RETINA is illustrated by the data flow diagram in Figure 8.

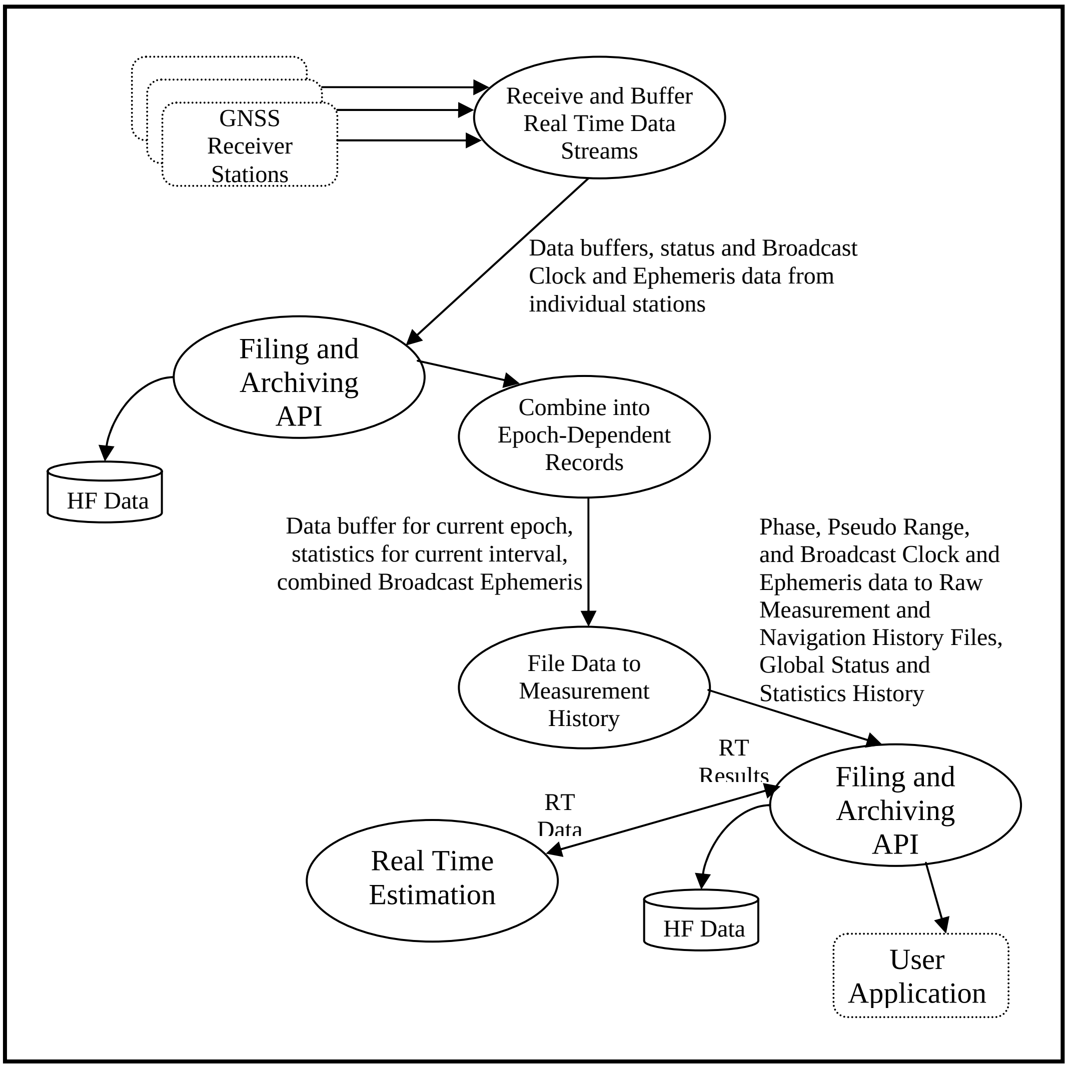

Figure 8. RETINA Real Time Data Flow

Event Logging and Alarm Management software includes a Logging Server accessible to all network nodes, for filing log messages to a Log History File, and an API layer for allowing applications to log error, warning and information messages. A Log Retriever API provides the interface to allow an Events and Alarms Display application access to the messages, enabling the RETINA operator to monitor the system and to acknowledge alarms.

The RETINA Job Scheduler is used for automating and monitoring the execution of both Batch and Real Time tasks. It is used for scheduling Batch jobs and sequences and for monitoring and restarting automatically the Real Time Resident Tasks. Jobs are submitted on any machine on the network using redundant Job Scheduling Demons. A Job Submission Demon on each node in the network communicates with the prime Job Scheduling Demon and is responsible for launching the applications on the local node. All functions are configured and monitored centrally via a Java Job Scheduling Display application. The User can intervene to cancel, abort, suspend or reschedule selected jobs and can also view messages and output from a selected job.

Algorithmic Software

Algorithmic software uses the infrastructure API and relies on the capabilities of the Job Scheduler for automated operation. The software includes applications for:

- Real Time Data Processing
- Real Time Estimation
- Batch Run Management
- Batch Data Preprocessing
- Batch Orbit Determination
- Batch Product and Statistics generation

The Real Time data processing concept is shown in Figure 8. The software provides combined measurement buffers and associated ephemeris data to the Real Time Estimation using the History File mechanism.

The Real Time Estimation application can operate in either a Real Time or a Retrieval mode. It is also possible to start from a point in the past and continue in Real Time mode when all the past data has been processed. The software uses the History File API to read the streamed combined raw measurements from Real Time Data Processing, as well as predicted orbit data from the Batch Estimation. It writes out Real Time results obtained using a Kalman Filter to History Files, including:

- S/c and receiver clock biases and Zenith Tropospheric Delays
- Phase biases and processed measurements

The software allows configuration changes without interruption, with utilities for the operator to schedule the addition or removal of satellites or receivers or updates to the station coordinates.

Batch Software is scheduled to run at regular frequent intervals (e.g. every 1-2 hours) as a Batch Sequence, using the facilities of the Job Scheduler. The first task in the sequence is a Batch Run Management task, which prepares the data and configures the software for the run. This is followed by Batch Data Preprocessing and the Orbit Determination software, which computes precise orbits and clocks. A Batch Products and Statistics task generates standard products, updates the Orbit History File used by the Real Time Estimation and generates comparison statistics between Batch and Real Time results. A further invocation of this software with a delay of several days allows the automated comparisons between RETINA and IGS results. Summary Results are filed to History Files while product files are compressed and archived.

Visualisation Software

Visualisation software is written exclusively in Java and includes applications for:

- Real Time Graphics
- Events and Alarms Management
- Job Scheduling

- ➢ GUI panels for Application software configuration, manual execution and inspection of results
- ➢ GNSS database administration

The Real Time Graphics display (see Figure 9) is the main application for Real Time visualization of the data filed to History Files. It is both a graphical and an alphanumeric display application with Real Time update. It can plot data from History Files as well as from ASCII files. For Real Time plots it includes auto-scrolling and auto-scaling features for hands-free operation, ensuring that new data is always displayed within the visible part of the plotting canvas. It also allows display of historical information using scroll bars and retrievals within user-specified time limits. A data popup window gives access to all the numerical values of the plotted variables and includes options for exporting the data to ASCII file.

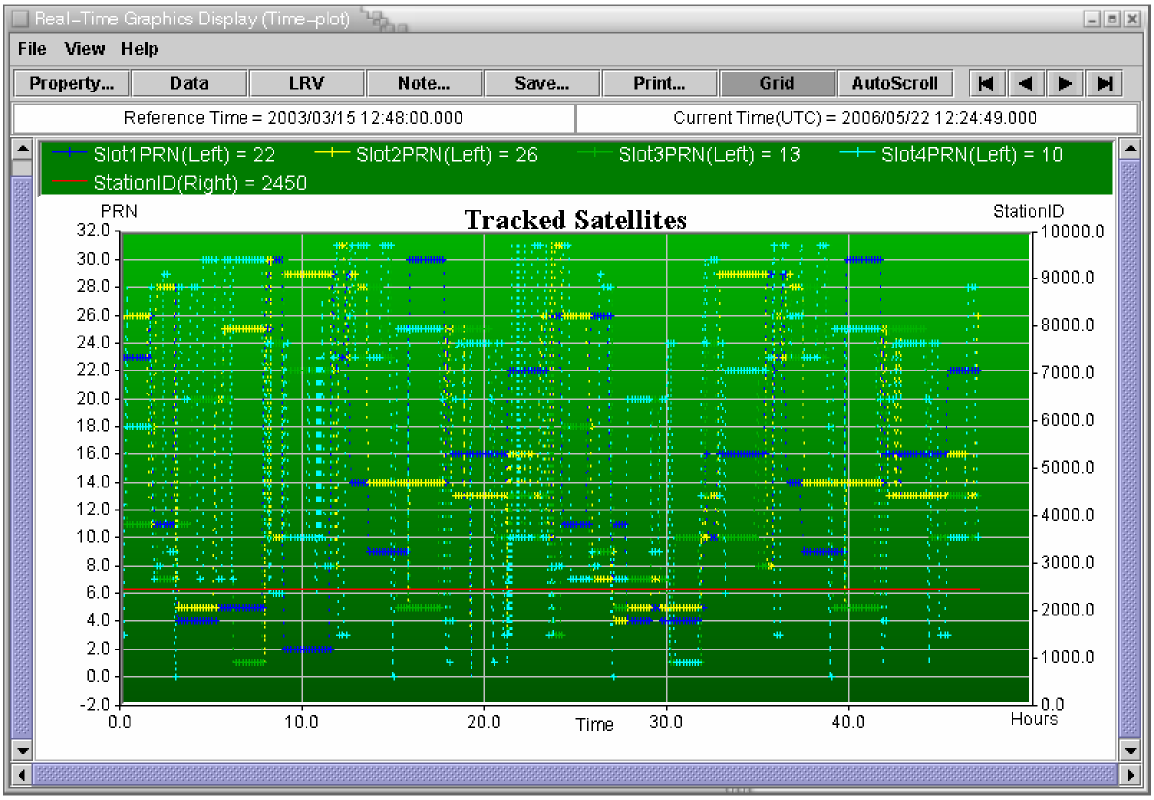

Figure 9. Real Time Graphics Display

Figure 10 shows the Integrated RETINA Graphical User interface. It is composed of a number of Java applications that are integrated under a single desktop. A configuration button enables the user to specify online or offline scenarios in which to run the applications. This allows the definition of separate environments to ensure that runs can be performed both inside and outside the operational environment. The top menu bar launches the individual application configuration and manual execution GUI panels and also allows editing of the GNSS database and invocation of the Real Time plots.

The left hand of the desktop is occupied by the Job Scheduler GUI, with facilities for monitoring and controlling Batch and Real Time task execution. Schedule, Job and

Sequences definition dialogues are used for specifying the nodes and times when jobs or sequences are expected to start.

The right hand pane within the RETINA desktop shows the Events and Alarms window. This displays filtered messages from all applications and includes utilities to allow the operator to receive and acknowledge alarms as well as to inspect historical log entries.

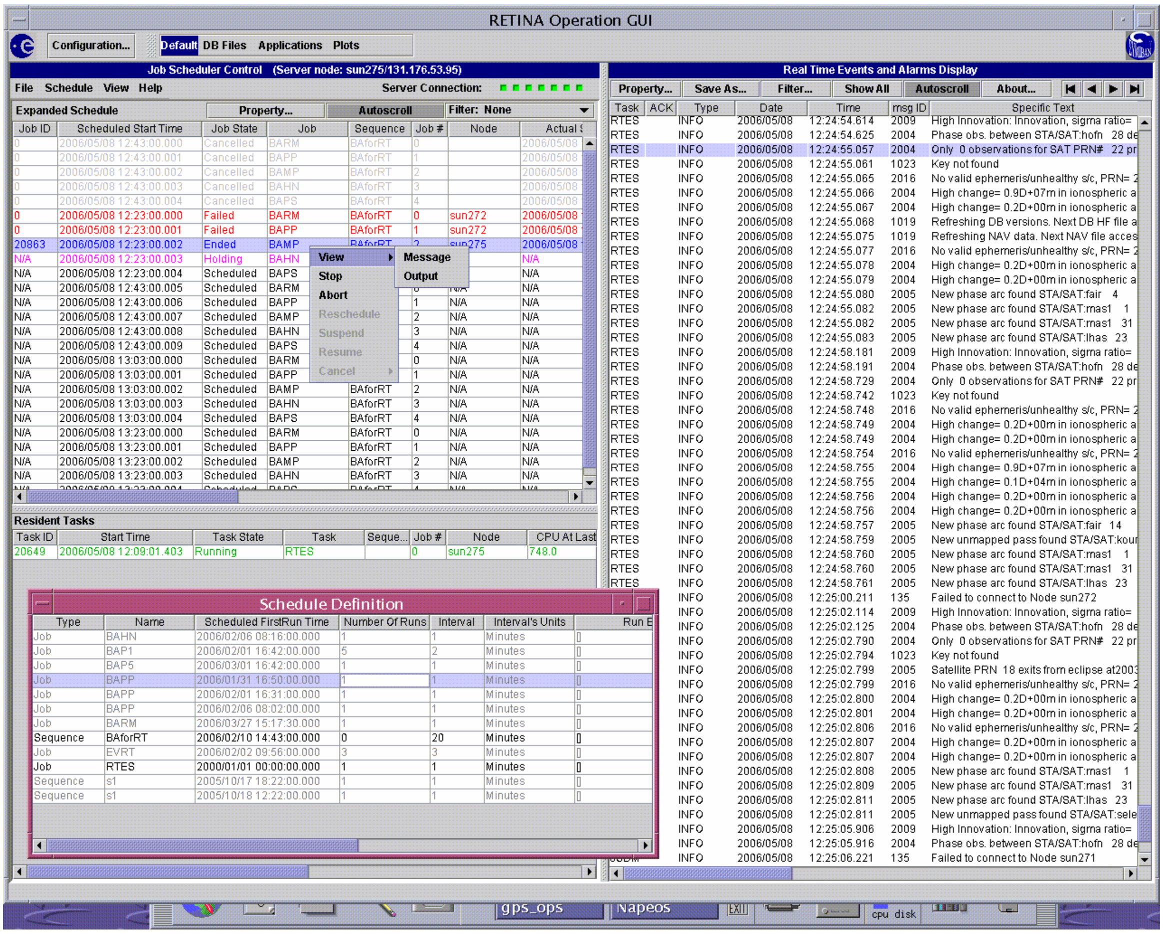

Figure 10. RETINA Integrated GUI

Conclusions

ESA/ESOC has adopted the latest RTIGS standard software and it is providing RT data streams from most station in its network within the framework of IGS Real Time Working Group.

A software prototype of a Real Time processing application has demonstrated that a combined batch/sequential filter approach can be used for obtaining decimetre level positioning accuracies.

This software is one of the applications that are being developed within the RETINA system, which provides the basic infrastructure for operational Real Time GNSS processing. Within this framework, the development of new applications can be easily incorporated as needs arise. RETINA also allows Batch and Real Time software to co-exist, with utilities both for automated operations and for manual intervention.