Journal of Global Positioning Systems (2008)
Vol.7, No.2 : 125-132

Journal of Global Positioning Systems
2008

A Conceptual Framework for Server-Based GNSS Operations

Samsung Lim and Chris Rizos
University of New South Wales, Sydney, Australia.

Abstract

The diversification of Global Navigation Satellite Systems (e.g. the current and modernized GPS, the revitalized GLONASS, the planned Galileo and Compass), is an opportunity for engineers, surveyors and geodesists because of expected improvements in positioning accuracy, operational flexibility, redundancy, and quality assurance. Recent research activities include new algorithms for multiple frequency ambiguity resolution, software-based receivers for re-configurability, network-wide corrections for utilising redundancy, reversed real-time kinematic schemes for quality/accuracy improvement, and a wide range of rover-side applications. This paper discusses the integration of these "pieces" of work into a new framework and facilitates information and communication technologies in order to derive benefits from network infrastructure such as continuously operating reference stations and local/regional GPS networks. Operational models are proposed for precise point positioning and real-time kinematic services including "near real-time" applications, which require an optimal design to balance the computational overhead with data communication latency. The proposed framework is designed to be a comprehensive, server-based, and thin-client platform. It provides end-users with "out-of-the-box" services. End-users should be able to obtain extensive GNSS capabilities and high productivity without conventional constraints such as an expensive set of receivers, proprietary data formats, user-installed carrier phase processing software, incomplete interoperability, limited communication links, etc. The framework also adopts up-to-date database technologies and web technologies that enable servers to perform data management and spatial analysis, while end-users are able to syndicate data and create their own business models. The framework has been applied to Sydney Network (SydNET), a network of continuously operating reference stations located in Sydney, Australia. It is expected that the new framework will be versatile enough to cope with a diverse range of user performance requirements and the operational requirements for communications and positioning computations.

Keywords: GPS, GNSS, Server-based, Framework.

1. Introduction

By definition, a server-based thin-client real-time kinematic (RTK) requires a designated server to compute a rover's coordinates in the required reference system by taking advantage of existing GNSS reference network infrastructure, instead of broadcasting corrections or data to users and placing the onus of obtaining a final solution on clients and their equipment. Final (position) solutions for all real-time (logged) users could be simply computed as a by-product of the continuous network processes – all the time satisfying the quality and integrity criteria implemented at the network administrator level (Rizos & Cranenbroeck, 2006).

Note that improved accuracy and reliability of the user coordinates can be expected if GNSS data is processed in the network mode (e.g. as implemented in network-RTK schemes), rather than as individual baselines as is the case of standard RTK-type techniques (i.e. single-base RTK). In addition, precise ultra-rapid ephemerides produced by International GNSS Service (IGS) can be used in network-RTK instead of the broadcast ephemeris. For example, network-RTK software "SNAPper" developed by the School of Surveying and Spatial Information Systems at the University of New South Wales, Australia, is able to generate real-time network corrections, based on IGS ultra rapid orbits. SNAPper is functionally equivalent to Trimble® GPSNet/VRS™ (Trimble, 2007) or Leica® SpiderNet™ (Leica, 2007). After all, there exist already a number of web-based services for the generation of coordinates via the post-processing of data submitted to a server by the client user. What is suggested in this paper is therefore to extend this capability to real-time data processing.

A server-based approach reverses the data flow in conventional RTK by requiring the user to transmit their data to the main server – sometimes also referred to as "reverse RTK" or "remote RTK" or "inverted RTK". Note that there is still the need for two-way communications between the client (field user) and the server (computer centre). The server software can select the optimal combination of continuously operating reference stations (CORS), and compute the best possible position solution before returning the result to the field user. The user then receives not only raw coordinates, but also a value-added product such as positioning quality indicators. Service providers can now exercise control over the generated products and, as a result, place a true commercial value on the service.

In addition, the user does not have to learn complicated GNSS surveying techniques or software. Safeguards, and thus integrity, can also be easily implemented into such a service. For example, if the number of satellites is too low, the geometry is unfavourable, or the multipath effects too detrimental, a message can be sent back to the user warning them that the provided solution is not optimal and that it may not meet their specifications. With the critical processes of legal traceability and integrity looming on the horizon for positioning services, such a total quality assured coordinate service may become increasingly attractive. For example, Nippon GPS Solution has implemented an inverted RTK service in Japan (Kanzaki, 2006), and is marketing their service by promoting the quality assurance aspects of server-based RTK processing.

2. Base Station Selection

Although it has been proven that network-RTK is superior to single-base RTK (Rizos & Han, 2003), there is a need to switch off network-RTK and turn on single-base RTK in certain cases, such as when there is a communication fault between a reference station and a distributed server, or during maintenance of a base and/or a server, or in the event there is low quality data from a reference receiver (for whatever reason). The fault detection module of a distributed server detects such faults and then queries the main database server to obtain the information on the most suitable reference station for single-base RTK. The main database server stores the geometry and topology information for this purpose, and the local database server stores the data quality information for the reference stations. This database approach is faster and more reliable than a simple selection on the basis of which reference station is nearest to the rover.

In this paper, the state of New South Wales (NSW), in Australia, is used as an example to illustrate the reference station selection logic. The basic assumption is that 220 cities and towns across NSW are sampled as reference receiver sites. Distances from these sites are graphically shown in Figure 1. Yellow regions represent the distance within 10km of a reference station, so that single-base RTK is suitable. Green areas and yellow-green areas are within 50km from a reference station where server-based RTK is ideal. Light blue to green areas are within 100km from a reference station and therefore server-based network-RTK is preferred. There is a significant amount of blue coloured area, outside the 100km radius of a reference station. These rural areas are expected to suffer a lower accuracy. This distance distribution can also be interpreted as a proportional error distribution of RTK as far as the geometric correlation is concerned.

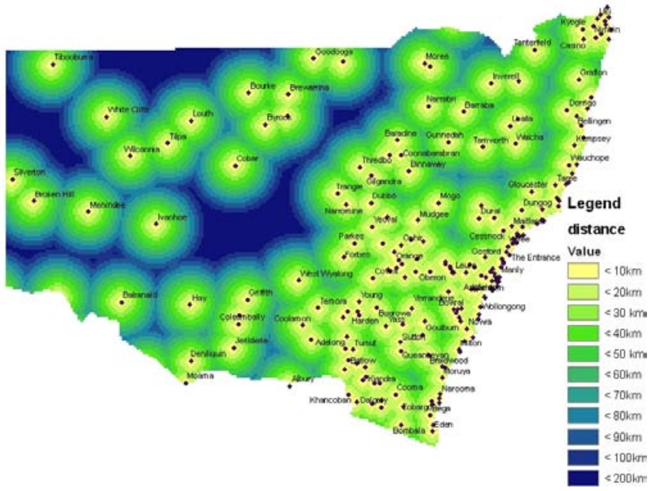

Fig. 1 Distance distribution of the assumed NSW network

Rather than calculating the distance from a rover on an *ad hoc* basis for the purpose of selecting the nearest reference station, Voronoi polygons in Figure 2 would be convenient. Voronoi polygons are obtained from the locations of the CORS so that each polygon contains only one reference station which is the nearest site from any location in the polygon. Therefore any rover within the polygon can perform single-base RTK with that particular reference station. Voronoi polygons can be obtained from Delaunay triangles and vice versa.

Fig. 2 Voronoi polygons of the assumed NSW network

Voronoi polygons must be stored in the main database server (see Section 3.4). Storing the geometry and topology of Voronoi polygons in the database is now feasible because of the spatial extensions of a Database Management System (DBMS). The benefit of storing such information in a database is that a query can be made as to whether a rover is located in the interior, or on the boundary, or exterior to a polygon, without the cost of computing the spatial relationship based on the coordinate information.

As for network-RTK, a combination of three or more reference stations can be selected to generate network corrections. However, three stations are sufficient and efficient in most cases. SNAPper uses three stations all the time, while GPSNet™ and SpiderNet™ have an option to choose more than three stations. To determine three reference stations, i.e. to form triangles, a Delaunay triangulation is more effective than any other triangulation. Delaunay triangles from locations of the CORS assure the condition that no reference station is located inside the circum-circle of any triangles. That is, Delaunay triangles maximise the minimum angle of triangles so that thin sliver triangles can be avoided, and therefore maximise the benefit of interpolating geometric correlations. Delaunay triangles and Voronoi polygons are geometrically paired off, i.e. Delaunay triangles can be defined by Voronoi polygons and vice versa.

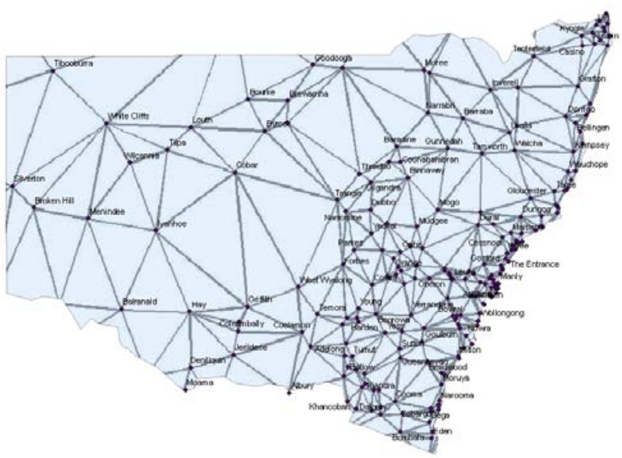

Fig 3 Delaunay triangles of the assumed NSW network

Figure 3 illustrates the Delaunay triangles for the assumed NSW CORS network. Again, the geometry and topology of Delaunay triangles can be stored in the main database server so that it is possible to identify if a rover is located in the interior, or on the boundary, or exterior to the triangle. The area of each triangle can be interpreted as a proportional error distribution of network-RTK unless error sources other than the geometric correlation are considered.

3. System Architecture

The system architecture of the proposed server-based thin-client RTK is indicated in Figure 4. The basic assumption is that distributed-computing is necessary to cope with simultaneous requests from hundreds of clients. Distributed-computing allows computers to efficiently communicate and individually process data, which is different from networked-computing.

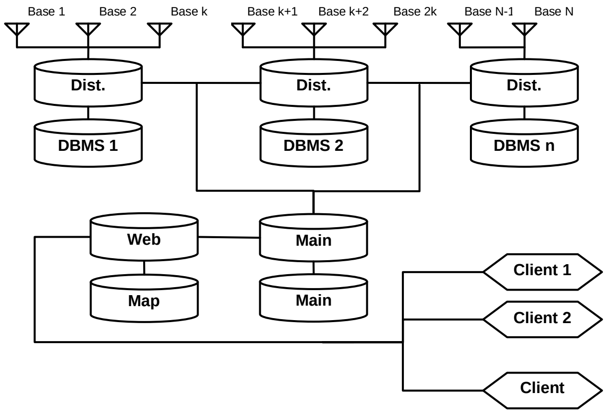

Fig. 4 System architecture of a distributed-computing based RTK service

4. Distributed Server

A distributed server receives data streams from k reference stations ($k \geq 3$) where k reference stations form as many Delaunay triangles as possible. Exclusive sets of reference stations are allocated to distributed servers, so that the last distributed server may have less than k reference stations unless the number of reference stations is divisible by k. For example, if a CORS network consists of Stations 1, 2, …, 6 and has 4 Delaunay triangles as depicted in Figure 5, then Stations must be allocated as described in Table 1.

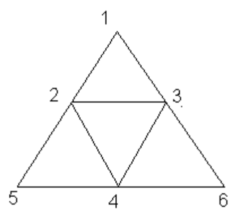

Fig. 5 Sample delaunay triangles

Table 1. Sample allocation of reference stations

k	Distributed Server 1	Distributed Server 2
3	1, 2, 3	4, 5, 6
4	1, 2, 3, 4	5, 6
5	1, 2, 3, 4, 5	6
6	1, 2, 3, 4, 5, 6	

Table 2. Sample allocation of Delaunay triangles

k	Distributed Server 1	Distributed Server 2
3	Δ1-2-3 Δ2-3-(4)*	Δ(2)-4-5 Δ(3)-4-6
4	Δ1-2-3 Δ2-3-4	Δ(2)-(4)-5 Δ(3)-(4)-6
5	Δ1-2-3 Δ2-3-4 Δ2-4-5	Δ(3)-(4)-6
6	Δ1-2-3 Δ2-3-4 Δ2-4-5 Δ3-4-6	

*Reference stations in brackets are received from other distributed servers.

If a distributed server does not include a reference station that forms a Delaunay triangle, then the distributed server must receive data streams from other distributed servers. Table 2 illustrates the situation. For example, in case of $k = 3$, Distributed Server 1 receives data streams directly from Stations 1, 2, and 3; and receives data streams indirectly from Station 4 via Distributed Server 2. Then server-side network-RTK in the area of two Delaunay triangles Δ1-2-3 and Δ2-3-4 can be serviced by Distributed Server 1. On the other hand, Distributed Server 2 receives data streams directly from Stations 4, 5, and 6; and receives data streams indirectly from Stations 2 and 3 via Distributed Server 1. As a result, all distributed servers can service server-side network-RTK for all possible Delaunay triangles. Note that the case of $k = 3$ is the most well-balanced distribution in terms of data transmission and reception or the computational workload.

Networked Transport of RTCM via Internet Protocol (NTRIP) is the chosen protocol for GNSS data transfer in this system. Each distributed server is equipped with an NTRIP caster so that the distributed server can broadcast data streams received from the allocated reference stations. Although the web server is the first contact point for all clients, the main server has the authority to permit advanced or privileged client access to a distributed server. Such a client can obtain direct data streams and perform traditional network-RTK (or single-base RTK), i.e. their own network algorithm can be applied to the data streams. It should be noted that an NTRIP client can be any Hypertext Transfer Protocol (HTTP) retrieval program, e.g. PHP, JavaScript, cURL, Wget, etc.

A normal client is expected to request the web server (and the web server to request the main server) to perform a server-based RTK computation. The main server determines if either single-base RTK or network-RTK is appropriate, depending on the rover requirement and the data quality of the reference stations. Then the main server assigns a distributed server to the client so that the distributed server performs a server-based RTK solution. The distributed server also parses "raw" data streams via NTRIP and inserts the data into the database of the local database server.

As for server-based network-RTK, the distributed server obtains precise ephemerides from the main database server and generates network corrections by interpolating residuals (or an extrapolation if the rover is not located within any of the triangles). Researchers have proposed a range of interpolation techniques: Linear Combination Model (LCM), Distance-Based Linear Interpolation Method (DBLIM), Linear Interpolation Method (LIM), Low-order Surface Model (LSM), and Least Squares Collocation Method (LSCM). Dai (2002) and Fotopoulus & Cannon (2001) reviewed these algorithms. Basically, these algorithms assume that errors are all spatially correlated, and therefore they are simply a variant of inverse distance weight (IDW) interpolation.

In order to improve the interpolation algorithm and to take into account the non-spatial correlation, the Kriging technique is a worthwhile alternative. Kriging is similar to regression analysis. Kriging is based on the assumption that the parameters being interpolated should minimise the estimation variance by applying an empirical covariance model. Kriging is only useful when the number of reference stations is large (≥ 50) so that the spatial correlation between reference stations can represent the empirical covariance model. Therefore the main server is ideal for computing Kriging parameters. Then each distributed server can use the parameters to perform Kriging for a client. This guarantees more uniformly distributed accuracy within the network than the traditional IDW interpolation because the parameters are obtained from network-wide observations. There are many types of Kriging, however, Ordinary Kriging is suitable for this application since there are enough observations to estimate the variogram, and for the state-wide area coverage of the NSW CORS network, the mean residuals apparently show unknown constant trend.
In summary, a distributed server should be able to perform the set of functions listed in Table 3.

Table 3. Distributed server functions

Function	Task
NTRIPClient1	To receive data streams from multiple reference stations via NTRIP
NTRIPClient2	To receive data streams from another distributed servers via NTRIP
NTRIPCaster	To broadcast multiple data streams via NTRIP

FaultDetection	To detect system faults such as a communication fault between a reference station and the distributed server
DataParser	To parse multiple data streams (Raw, RTCM, RT-IGS)
InsertDB	To insert the data into databases
ServerRTK	To perform server-based single-base RTK
ServerNetworkRTK	To perform server-based network-RTK
IDWInterpolator	To interpolate double differenced residuals (LCM, DBLIM, LIM, LSM, LSCM)
Kriging	To interpolate double differenced residuals (Ordinary Kriging)
NetworkAR	Network ambiguity resolution
NetworkResidual	To compute double-differenced residuals

5. Local Database Server

A distributed server can also serve as a local database server, or the two can be separately implemented. A distributed server parses data streams coming from reference stations and inserts the data into a database while the corresponding local database server monitors the quality of the data. The network administrator would want the quality check, the quality assurance, detection of cycle slips and outliers, monitoring high frequency variations in the double-differenced residuals, etc. A local database server sends such information to the main database server so that the network administrator can perform web-based monitoring by accessing the main database server only.

A client who wants post-processed positioning solutions can request the web server to obtain Receiver INdependent EXchange (RINEX) files. Then the web server retrieves RINEX files from a local database server via the main server. Figure 6 shows the webpage for extracting RINEX files from SydNET, which is a network of continuously operating reference stations located in Sydney, Australia (Department of Lands, 2006).

The main purpose of a local database server is to perform DB functions. A set of necessary DB functions is listed in Table 4.

Table 4. Local DB server functions

Function	Task
StoreDB/RetriveDB	To store and retrieve the parsed data into databases
MonitorDataQuality	To monitor the quality of data from reference stations
TransmitDataQuality	To transmit the monitored information to the main database server
RetrieveRINEX	To retrieve RINEX files from databases

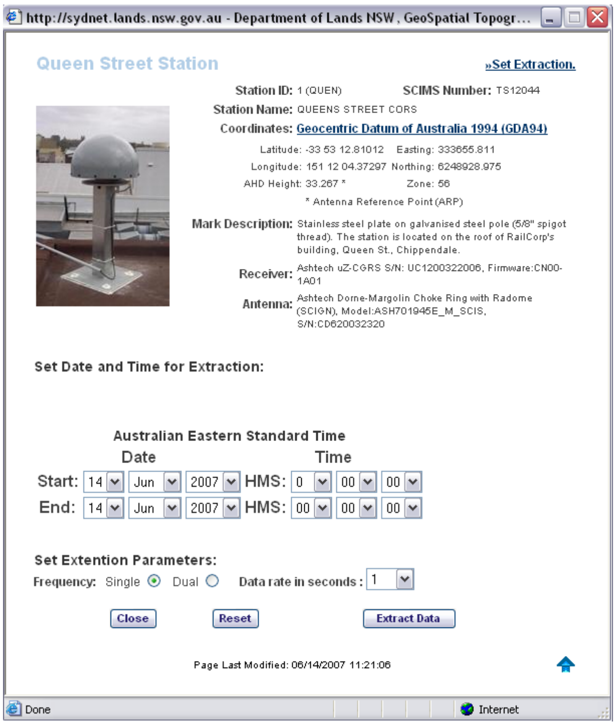

Fig. 6 Web extraction for SydNET RINEX files

6. Main Server

The main server distributes computing resources to the distributed servers. Once a client requests server-based RTK solutions, the main server queries the main database server for the most suitable Delaunay triangle and allocates a distributed server to the client. The main server calculates the network-wide parameters, e.g. orbit errors and atmospheric parameters from the network-wide observations, empirical covariance parameters for Kriging, and so on. The main server determines whether precise IGS orbits or the network orbits must be used, unless there is a special request from the client. The set of main server functions is listed in Table 5.

Table 5. Main server functions

Function	Task
SelectRTK	To distribute computing resources
DownloadIGS	To download precise IGS orbits
InsertIGS	To insert IGS orbits into the main database server
NetworkOrbits	To compute orbit errors based on the network-wide observations
NetworkAtmosphere	To compute atmospheric errors based on the network-wide observations
NetworkParameters	To compute the network-wide parameters e.g. Kriging parameters

7. Main Database Server

In this proposed framework, the main database server requires Open Geospatial Consortium (OGC) Simple Feature Specification for Structured Query Language (SQL). OGC has published the specification in 1997, in order to propose a conceptual method for a SQL DBMS to deal with spatial data. Most DBMSs have implemented the spatial extensions recommended by OGC (Open Geospatial Consortium, 2006). For example, Oracle Spatial and Oracle Locator comply with the OGC specification. MySQL has implemented a spatial extension to follow the specification. PostgreSQL has a spatial module known as PostGIS. Therefore, unlike a traditional DBMS it is possible to store the geometry and topology of geographical features within a database.

Spatial operators and functions of a DBMS are powerful because spatial relationships can be retrieved rather than computed. This approach reduces the computational overhead significantly. For example, some spatial functions available include:

- *Area* returns the area of a polygon
- *Contains* indicates if Feature 1 completely contains Feature 2
- *Crosses* indicates if Feature 1 spatially crosses Feature 2
- *Disjoint* indicates if Feature 1 spatially disjoints from Feature 2 (i.e. does not intersect with each other)
- *Distance* returns the shortest distance between two points
- *Intersects* indicates if Feature 1 intersects Feature 2
- *Overlaps* indicates if Feature 1 overlaps Feature 2
- *Related* indicates if a given spatial relationship between Features 1 and 2 exists
- *Touches* indicates if Feature 1 touches Feature 2
- *Within* indicates if Feature 1 is within Feature 2

The main database server stores the coordinates, Voronoi polygons and Delaunay triangles of reference stations. Voronoi polygons are used to assign the nearest reference station for single-base RTK, while Delaunay triangles are used for network-RTK. The nearest reference station or the triangle that contains the rover is selected upon a client's request for server-side RTK. The main database server also stores the network-wide parameters: orbit errors, atmospheric parameters, and Kriging parameters. The set of main database server functions is listed in Table 6.

Table 6. Main DB server functions

Function	Task
StoreDataQuality/ RetrieveDataQuality	To store and retrieve the monitored information such as data quality checks, data quality assurance parameters, detected cycle slips and outliers, high frequency variations of double-differenced residuals, abnormal multipath effects, etc.
StoreIGS/ RetrieveIGS	To store and retrieve precise IGS orbits
StoreNetworkOrbits/ RetrieveNetworkOrbits	To store and retrieve the network-wide orbit errors
StoreNetworkAtmo/ RetrieveNetworkAtmo	To store and retrieve the network-wide atmospheric errors
StoreNetworkParam/ RetrieveNetworkParam	To store and retrieve the network-wide parameters, e.g. Kriging parameters
StoreFeature/ RetrieveFeature	To store and retrieve geographical features such as points, lines, and polygons
StoreReference/ RetrieveReference	To store and retrieve the geometry and topology of reference stations
StoreVoronoi/ RetrieveVoronoi	To store and retrieve the geometry and topology of Voronoi polygons
StoreDelaunay/ RetrieveDelaunay	To store and retrieve the geometry and topology of Delaunay triangles

8. Web Server

The web server is the portal for clients and the network administrator. Ordinary clients simply access the web server and submit their data streams via NTRIP in order to activate server-based RTK. Authorised clients request

RINEX files for their post-processing or request data streams from the server so that they can perform traditional RTK. JavaScript and eXtensible Markup Language (XML) play an important role in the web server because a web-based Application Programming Interface (API) for Asynchronous JavaScript and XML (AJAX) must be used by clients. Advanced clients or the network administrator can utilise the API for their web-programming, similar to programming with the Google Maps API (Lim, 2005). AJAX is not a new technology, but a new paradigm that uses JavaScript, XML, Document Object Model (DOM), and HTTP Request.

The objective of the proposed framework is that clients do not need to learn complicated GNSS algorithms or software. Precise point positioning and RTK services including "near real-time" applications are performed server-side and delivered to clients upon their HTTP Request. For example, near real-time structural deformation can be monitored on a client's website simply by sending an HTTP Request to the web server and receiving the output. The visualisation of the output can be interfaced by the web-based API. If a base map is necessary for the visualisation purpose, the client can query the web map server. This is currently feasible because of the AJAX technology. Furthermore, the surveying result, e.g. monitoring of the structural deformation, can be syndicated to anyone using the RSS technology. Likewise, the network administrator can develop tools for web-based monitoring of the CORS network.

9. Concluding Remarks

A new framework for server-based thin-client RTK services is proposed where distributed-computing is essential. The proposed framework is intended to extend the capability of the server to real-time data processing by integrating information and communication technologies and CORS network infrastructure. An optimal design that balances the computational overhead with communication latency is described.

The proposal provides end-users with "out-of-the-box" services, i.e. end-users obtain extensive GNSS capabilities and high productivity by overcoming the conventional constraints of an expensive set of GNSS receivers, proprietary data formats, user-installed carrier phase processing software, incomplete interoperability, limited communication links, and so on. The framework also utilises database and web technologies which enable servers to perform data management and spatial analysis, while end-users are able to syndicate data and create their own business models for data and result dissemination.
In order to demonstrate the usability of the framework, a prototype web-based data quality monitoring system has been developed on the Java platform. The monitoring system is able to display statistics of a local GNSS network and from network-RTK software in real time.

Acknowledgements

This paper is a revised version of the authors' paper presented at the Institute of Navigation Conference in 2007 (Lim & Rizos, 2007). The research presented in this paper was carried out with support from the Cooperative Research Centre for Spatial Information Project 1.4 "Integrating electricity, telecommunications and government infrastructure to deliver precise positioning services in regional areas", 2007-2010.

References

Dai, L. (2002), ***Augmentation of GPS with GLONASS and Pseudolite Signals for Carrier Phase-Based Kinematic Positioning***, PhD Thesis, School of Surveying & Spatial Information Systems, The University of New South Wales, Australia.

Department of Lands (2006), ***Online SydNET Data Access***, http://sydnet.lands.nsw.gov.au/, accessed 4 July 2007.

Fotopoulus, G. and Cannon, M.E. (2001), ***An Overview of Multi-Reference Stations Methods for Cm-Level Positioning***, GPS Solutions, 4(3): 1-10.

Kanzaki, M. (2006), ***Inverted RTK System and its Applications in Japan***, 12th IAIN Congress & 2006 Int. Symp. on GPS/GNSS, Jeju, Korea, 18-20 October, 455-458.

Leica (2007), ***Leica GPS Spidernet,*** http://www.leica-geosystems.com/corporate/en/products/gps_systems/lgs_4591.htm, accessed 16 August 2007.

Lim, S. (2005), ***Lecture Notes on Web 2.0 and AJAX***, School of Surveying & Spatial Information Systems, The University of New South Wales, Australia.

Lim, S. and Rizos, C. (2007). ***A New Framework for Server-Based and Thin-Client GNSS Operations for High Accuracy Applications in Surveying and Navigation***, U.S. ION GNSS, Fort Worth, Texas, 25-28 September, CD-ROM procs.

Open Geospatial Consortium (2006), ***OpenGIS Web Map Server Implementation Specification***, Ref. OGC 06-042.

Rizos, C., & Cranenbroeck, J.van. (2006), ***Alternatives to current GPS-RTK services***. 19th Int. Tech. Meeting of the Satellite Division of the U.S. Inst. of Navigation, Fort Worth, Texas, 26-29 September, 1219-1225.

Rizos, C., & Han, S. (2003), ***Reference station network based RTK systems - Concepts & progress***, Wuhan University Journal of Nature Sciences, 8(2B), 566-574.

Trimble (2007), ***Trimble Virtual Reference Systems***, http://www.trimble.com/vrs.shtml, accessed 16 August 2007.