

2-2015

Real-time Monitoring of Tectonic Displacements in the Pacific Northwest through an Array of GPS Receivers

Răzvan Popovici

Răzvan Andonie

Walter M. Szeliga

Timothy I. Melbourne

Craig W. Scrivner

Follow this and additional works at: <https://digitalcommons.cwu.edu/cotsfac>



Part of the [Computer Sciences Commons](#), and the [Geophysics and Seismology Commons](#)

Real-time Monitoring of Tectonic Displacements in the Pacific Northwest through an Array of GPS Receivers

R. Popovici, R. Andonie, W.M. Szeliga, T.I. Melbourne, C.W. Scrivner

Răzvan Popovici

1. Altair Engineering Inc.

Troy, MI, USA

rp@miravtech.com

2. School of Engineering and Computer Science, Oakland University

Rochester, MI, USA

Răzvan Andonie*

1. Computer Science Department, Central Washington University

Ellensburg, WA, USA

2. Electronics and Computers Department, Transilvania University

Braşov, Romania

*Corresponding author:andonie@cwu.edu

Walter M. Szeliga, Timothy I. Melbourne, Craig W. Scrivner

Department of Geological Sciences, Central Washington University

Ellensburg, WA, USA

walter@geology.cwu.edu, tim@geology.cwu.edu, scrivner1@cwu.edu

Abstract: The Pacific Northwest Geodesic Array at Central Washington University collects telemetered streaming data from 450 GPS stations. These real-time data are used to monitor and mitigate natural hazards arising from earthquakes, volcanic eruptions, landslides, and coastal sea-level hazards in the Pacific Northwest. The displacement measurements are performed at millimeter-scale, and require stringent analysis and parameter estimation techniques. Recent improvements in both accuracy of positioning measurements and latency of terrestrial data communication have led to the ability to collect data with higher sampling rates, of up to 1 Hz. For seismic monitoring applications, this means 1350 separate position streams from stations located across 1200 km along the West Coast of North America must be able to be both visually observed and analyzed automatically. We aim to make the real-time information from GPS sensors easily available, including public access via interfaces for all intelligent devices with a connection to the Internet.

Our contribution is a dashboard application that monitors the real-time status of the network of GPS sensors. We are able to visualize individual and multiple sensors using similar time series scales. We are also able to visualize groups of sensors based on time-dependent statistical similarity, such as sensors with the the highest variance, in real-time. In addition to raw positioning data, users can also display derived quantities, such as the Allan variance or the second derivative of a data stream.

Keywords: Real time dashboard, signal analysis, data streaming, GPS, slow earthquake.

1 Introduction

The Pacific Northwest Geodesic Array (PANGA) Geodesy Laboratory at Central Washington University (CWU) has a primary scientific role to support high precision geodetic measurements using Global Positioning System (GPS) observations in order to characterize crustal deformation, plate tectonic motions, coastal and earthquake hazards, and other environmental science applications. Under contracts from the National Science Foundation, the National Aeronautics and Space Administration, the U.S. Geological Survey, and UNAVCO, Inc., the Laboratory

analyses all publicly shared GPS data within the Cascadia subduction zone [1] and greater Pacific Northwest. The Geodesy Laboratory analyzes data from roughly 1000 GPS stations that comprise the EarthScope Plate Boundary Observatory, whose stations span the Pacific-North American tectonic plate boundary from Alaska to Mexico. Fig. 1 is a map of all stations currently analyzed by CWU. In addition to serving as the Data Analysis Facility for the PANGA, the Geodesy Laboratory also supports field experiments on Cascades volcanoes and in mainland Mexico, Baja California, California, Idaho, Montana, Oregon, and Washington. CWU also operates a continuous GPS network in Nepal.

Data from 450 GPS stations are telemetered in real-time back to CWU, where they are processed, also in real-time, using both NASA Jet Propulsion Lab's RTG [2] software as well as Trimble's RTKNet Integrity Manager software to provide relative positioning of several mm resolution across the Cascadia subduction zone and its metropolitan regions. These real-time data are used to monitor and mitigate natural hazards arising from earthquakes, volcanic eruptions, landslides, and coastal sea-level hazards. In addition, they are also used to monitor man-made structures such as Seattle's sagging Alaska Way Viaduct, WA SR-520 and I-90 floating bridges and power-generation/drinking-water-supply dams throughout the Cascadia subduction zone, including those along the Columbia River.

The data streams from these 450 receivers is continuously downloaded, analyzed, archived and disseminated, as part of the existent geophysics and tectonics research programs within the Department of Geological Sciences, CWU. These tectonic displacement measurements are performed at millimeter-scale, and requires stringent analysis and parameter estimation techniques. The Geodesy Laboratory uses NASA's GIPSY OASIS (GPS Inferred Positioning SYstem, Orbital Analysis and SIMulation Software) software to translate GPS satellite phase observables into position time series, and in-house parameter estimation and modeling software to quantify crustal deformation caused by plate tectonics, earthquakes, landslides and volcanic eruptions.

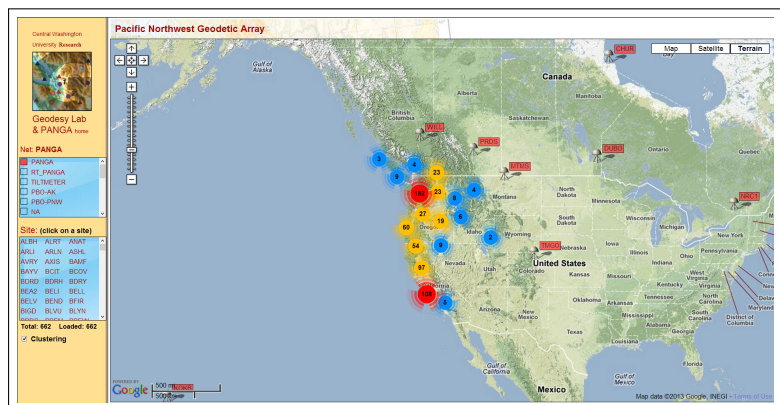


Figure 1: Panga sensor placement.

The position of a GPS station is estimated from a combination of satellite range and carrier phase measurements, with the position accuracy being heavily dependent on the accuracy of the satellite ephemerides, stability of the Cesium time standards on board the orbiting satellites, and the realism of models for both the tropospheric water content and electron density of the ionosphere. In the early days of GPS positioning, when none of these variables were particularly well-determined, long-duration observations were necessary to properly constrain models of each error source, with typically a single position estimated daily. These daily solutions have enabled the study of long term crustal deformation and even earthquake source processes, but only well after the observations were recorded.

At it's inception, the GPS receivers networks were designed to monitor tectonic displacements

with a coarse-grained temporal resolution, with data reductions to one measurement per day. However, over the intervening two decades, real- or near-real time estimates of ephemerides and real-time estimations of satellite clock behavior and atmospheric delays have enabled positioning estimates to be made at rates as rapid as once per second. Moreover, the widespread implementation of real-time data telemetry systems have enabled these high-rate position estimates to be computed nearly in real-time. This, combined with increased precision in all aspects of GPS position estimations, essentially transforms a continuously-operating, real-time networks of GPS receivers into a seismic network capable of detecting the strong ground motion that accompanies large earthquakes in real-time. For example, Fig. 2, shows data from one component of motion at a single site due to the well known M9 2011 Tōhoku earthquake in Japan. This time series is comparable to data from traditional seismic instrumentation for the event but is free from data artifacts common to seismic instruments, such as "clipping" of a time series due to physical limitations of the instrument response. Thus real-time high-sample-rate GPS solutions can provide useful data to determine the magnitude of large earthquakes in the immediate quake aftermath when the size of the event is still being determined and emergency response is being organized.

Ground deformation due to a major earthquake leads to a sudden change in the positions of sensors across a wide zone. One outstanding problem in monitoring any large network of continuously operating instruments is facile observation and analysis of its data streams. In the case of the Pacific Northwest Geodetic Array, the 450 GPS stations each output three continuous data channels: latitude, longitude and vertical position. For seismic monitoring applications, this means 1350 separate position streams from stations located across 1200 km along the West Coast of North America must be able to be both visually observed and analyzed automatically. One of the characteristics of seismic events is spatial coherence of the observed earthquake deformation, which requires that station behavior be monitored in spatially-clustered groups.

There are two conflicting factors which motivate our present work. One is the potentially valuable information which can be extracted from the GPS sensors' data stream for detecting major earthquakes. Obviously, such data is valuable when extracted and processed in real-time. This takes us to the second factor, which is the challenge posed by mining big datasets of streaming data: we are more concerned about the abundance, not the lack of data. The recent advancements in data collection, such as data streams from sensors, exceed the ability of the data scientists to really put data in context of the questions and extract usable knowledge.

Our final goal is to make the real-time information from GPS sensors easily available. This includes wider public access via interfaces for all intelligent devices with a connection to the Internet. Practically, a geologist with mobile phone access, should have real-time access to streaming GPS position data. When combined with other measurements and information, this would help him to detect a major earthquake.

Our contribution is a dashboard application that monitors the real-time status of a network of GPS sensors. We are able to visualize individual and multiple sensors using similar time series scales. We are also able to visualize groups of sensors based on time-dependent statistical similarity, such as sensors with the the highest variance, in real-time. In addition to raw positioning data, users can also display derived quantities, such as the Allan variance or the second derivative of a data stream.

Section 2 describes our interface to the PANGA data stream. Section 3 introduces our main contribution, the dashboard application, whereas in Section 4 we discuss our results. In Section 5, we conclude with final remarks and open problems.

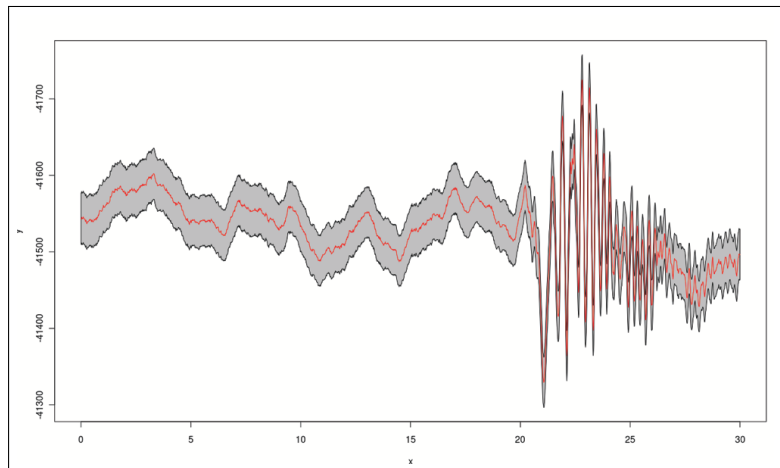


Figure 2: Longitudinal movement of a GPS receiver located in Japan, during the Tōhoku earthquake. The read line is the reading and the gray area is one standard deviation border around the reading.

2 The PANGA data output stream interface

The dashboard input data is the output of the PANGA network, and it consists of a list of streams with active/inactive status. Each stream contains a set of samples, each containing the North/West/Elevation displacements, as well as quality statistics. Although not used by the dashboard, it is possible to compute the correlation between each pair of any of the three dimensions with displacement. We first will describe the PANGA data structure and our interface to the resulted output data stream.

The PANGA network aims to precisely measure the displacement of multiple locations organized in a network. Physically, the network is composed of a number of GPS receivers, which receive signals from satellites and regularly send that data to a processing center using a cellular radio network. In our data model, such a receiver is called a 'site'. Each site provides one or more streams of processing solutions, which are displacement values and quality information. Depending on the type of, the streams can contains raw solutions, or they are processed further using a Kalman filter. Some streams provide data points every second, other streams provide an aggregation of the data over a longer time span. Each stream contain a record payload storing the time stamp, as well as the variations of the position with respect of north, east and elevation relative to a reference position. It is possible for a stream to miss some points in time and it is not guaranteed that all the data is available at every sampling time stamp. Fig. 3 depicts the conceptual entities, described above, as well as the querying mechanism.

A large data repository¹ is available, as well as the diverse real time data streams we focus on. The challenge is to visualize and exploit the historical data to increase the accuracy of current readings or to efficiently categorize a major earthquake. Literature examples, such as [3], show that big data can overwhelm the analyst, either if the right algorithm is not in place, or if the nature of the data does not address, directly or indirectly, the research questions.

The system can be interrogated using a RESTful Web interface² that returns data serialized in JSON. RESTful (Representational State Transfer) [4], although known for more a decade, is a recently embraced concept in the architecture of web services that excludes the presence of an established session between client and server, as opposite to the classical dynamic web

¹http://www.geodesy.cwu.edu/data_ftp_pub/data/

²<http://www.panga.org/realtime/data/api>

approach of a shared state encompassed by the session. The advantages of the new approach are the possibility to spread the server requests over a number of cluster nodes, as well as simplicity in the implementation of both server and client, since it is eliminated the need to implement system states and shared session. JSON (JavaScript Object Notation) [5] is an emerging standard for structured data encoding, designed to replace XML in lightweight implementations, such as browsers and mobile applications. Initiated by JavaScript, it can be used under most of the modern platforms, including (in our case) Java.

The interface is composed of a set of query verbs and parameters. We use the verb `sitelist` to acquire the list of the sites available. For each site, the response provides site attributes and the streams associated with the site, including the status (active or inactive) of the stream.

The `records` verb provides access to the position data for a stream. The stream name is given as a required parameter. The response is composed of the current set of positions for the stream, the time stamp of the last record and current server time. The last record time stamp can be used as an option parameter to constrain the next query for that stream. If the time stamp of the last record of the previous query is included, then the response will only include records newer than those received previously. In this way, the client maintains the state necessary to avoid the unnecessary download of records that were already contained within a previous interrogation.

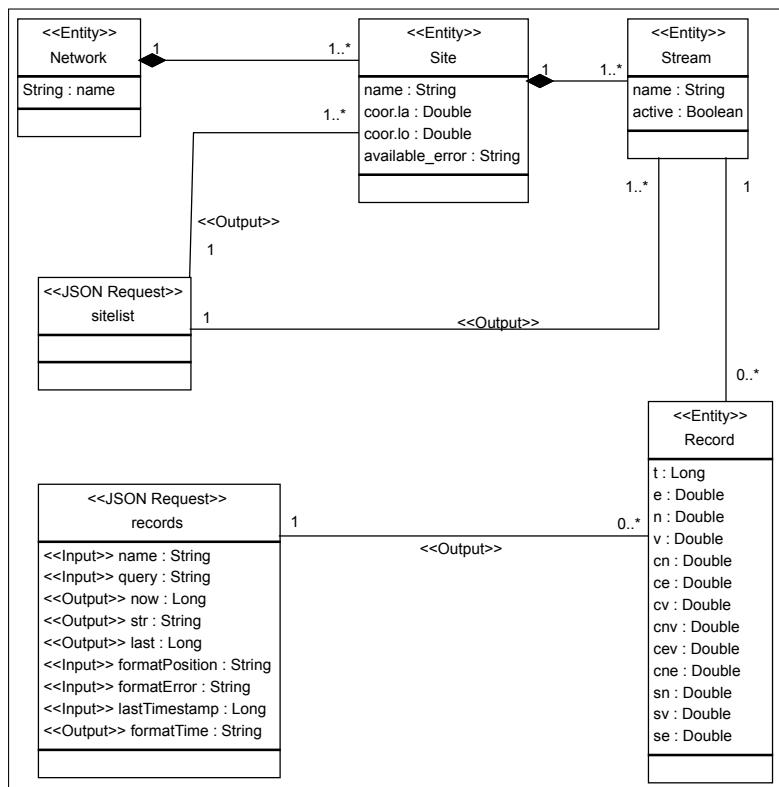


Figure 3: PANGA data model and web interrogation structure.

In the UML diagram of Fig. 3, conceptual classes are labeled with "Entity" stereotype, so that they differentiate of JSON requests. Within a request, each attribute can be either input or output. The associations of JSON requests are also of "Output" stereotype, since they are part of the reply.

Multiple networks could be theoretically represented in the system. This is the reason why we introduce the "Network" class. However, multiple networks may have different data models and different querying systems. For example, the network that captured the Tōhoku earthquake [6],

only provides the standard deviation value for each of the three dimensions of displacement, as a measure of precision, rather than the full covariance of the solution.

For each data stream PANGA provides multiple "queues", with different sampling intervals. This is a convenience for charting applications, which do not need to perform analysis, but for our application we only request data from at the full sample rate.

3 The dashboard application

Our dashboard is oriented on stream processing, which facilitates the visualization of PANGA data. The application is J2EE compliant, developed in two layers: the server layer consists of a EJB singleton that downloads PANGA data, stimulated by a timer, while the second layer is a web front end, which connects to the first layer and receives the instantaneous state of the data. This way, independently of the number of the web dashboard, the PANGA server is only queried once. As shown in Fig. 4, the implementation is divided into three logical layers: the interface layer, the J2EE middle layer, and the data layer. Since the data layer is highly volatile and it does not need to be persisted into a database, it is not implemented as an EJB entity. The J2EE layer is responsible for scheduling of the downloads of PANGA data, as well as for the storage of user configuration settings within the user session.

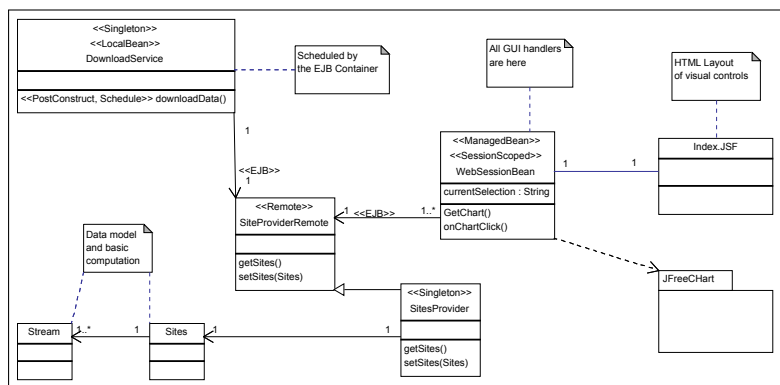


Figure 4: Interface-centric class diagram of EJB components.

We run our application with the Oracle Glassfish application server [7], but it can be deployed most of the J2EE application servers. Opposite to most of the J2EE applications, we don't mandate the existence of a back-end database, since we don't persist any data outside the user session scope. For future analysis, we save the JSON content downloaded by the application in a local directory. Since this infringes the J2EE requirements, the option is only designated to be used in a testing environment and shall be subject of scrutiny in case of deploying the application on a cluster.

The application runs on both Windows and Linux server (with untested possibility to run on Apple computers). We have tested all major browsers: Microsoft Internet Explorer, Mozilla Firefox, Chrome, and Safari, with different screen resolutions. The best resolution to visualize the dashboard is High Definition Television (HDTV) 1920 × 1080, resolution common to most of the TV screens available on the market. It is possible to access the dashboard with mobile devices, through the browser capability, as shown in Fig. 5.

The user interface layer is based on the PrimeFaces library [8], using Google Maps [9] for spatial display of sensors (see Fig. 5). The selected sensors are displayed yellow, the active ones green and the inactive transmitters are black. We extend the capabilities of the control, provided by PrimeFaces, to allow replacement of map icons without the need for full reload of

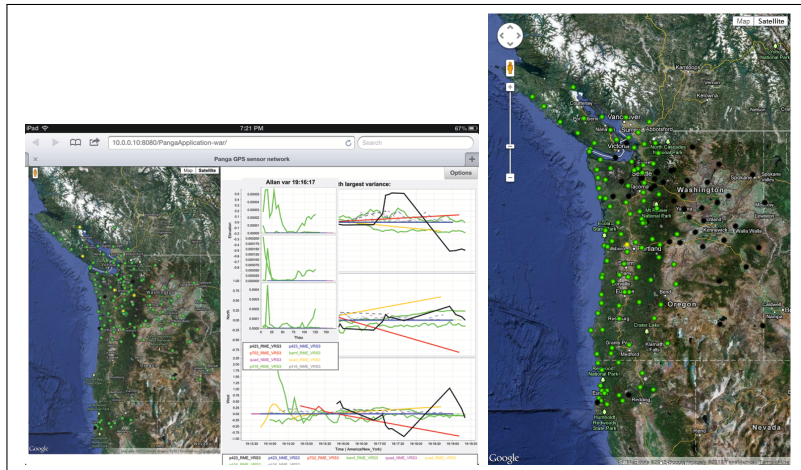


Figure 5: left: Screenshot of the dashboard on Apple iPad; right: Map of the sensors.

the control's content. This way, we avoid flickering the map if the data has been changed. We use the JFreeCharts library [10] to render the charts. The rendering occurs on the server side, within the servlet container, JFreeCharts is able to render SVG content to a Java stream, by receiving a Graphical Context object compatible with Java AWT drawing system. Using this technique we render the main chart and Allan variance chart, shown in Fig. 6.

The PrimeFaces library was not designed with real time charts in mind. Particularly, it is difficult to refresh the content of a chart without performing a full control refresh; even it technologies like Ajax allow the discriminative refresh of specific controls on a page, refreshing the image will cause it to disappear on the duration of the download, and then it reappears again. Since a regular HTTP web transaction takes between 20-300 milliseconds, there are at least a few frames when the image is missing. For this reason, we use a technique similar to the double buffering: we load the content of the new chart in a hidden IMG element, then, once load is completed, the content of the hidden image is moved to the visible image control. Since this last operation is performed on the user browser, its duration is in sub-millisecond range, hence no flickering occurs.

A similar challenge has been observed for the map control. When a new set of sensor data is available, the Google Map control requires a complete refresh, in order to accommodate the new changes in the way envisioned by the authors of PrimeFaces. We had to access the control at JavaScript level, remove the existing markups layers and add the newly available points. This way, we retain the position of latitude and longitude, as well as the zoom level of the map, established by the user on the local browser.

Fig. 7 shows a sensor plot, as well as the numeric second derivative of the signal, and the local Allan variance.

4 Discussion

We used the second derivative and the Allan variance [11] to characterize the GPS data streams. The Allan variance is a statistical measure of the stability of a series of observations over various periods. This statistic was initially developed to investigate the stability of the atomic oscillators that comprise most high-precision frequency standards [12, 13]. The Allan variance is uniquely suited to identifying not only the frequency-dependency of noise in a time series, but over what periods that noise operates.

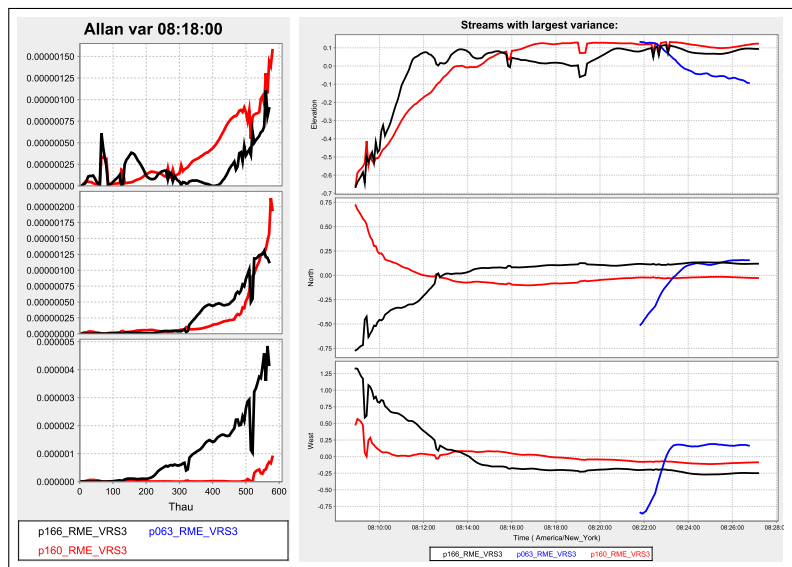


Figure 6: Allan variance chart.

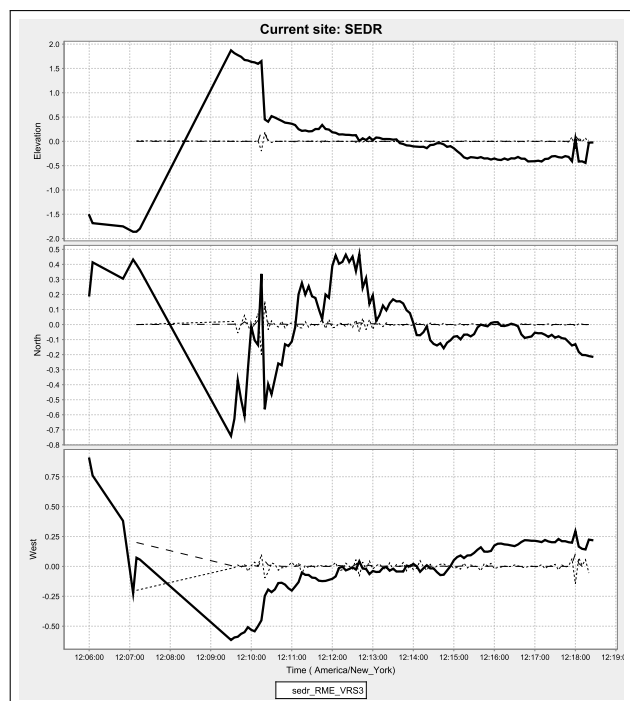


Figure 7: Chart of sensor plot together with Allan variance and second derivative.

GPS position time series are often composed of daily aggregates of higher-rate observations. This aggregation is typically performed to improve the noise characteristics of the resulting geodetic time series. Still, these daily aggregate time series are known to be contaminated by colored noise [14]. In the past decade, time series composed of the higher-rate observations themselves have begun to provide crucial measurements of absolute ground deformation, both static and dynamic, due to earthquakes [15]. However, the noise characteristics of position time series at sampling intervals shorter than one day are poorly known and are likely to be strongly dependent on the subtleties of each GPS processing methodology [16]. The characterization of the noise content of these high-rate GPS measurements is essential for their future use in the study of earthquake deformation [17].

Previous authors [16, 18, 19] have used the Allan variance statistic to study the noise characteristics of various geodetic time series, such as Earth Orientation Parameters, GPS position time series, and VLBI baseline time series. With the advent of streaming of near-real-time, high-rate GPS time series, we are provided with the opportunity to study not only the frequency characteristics of high-rate geodetic time series, but their evolution with time. This ability to monitor the evolution of the noise characteristic of each individual instrument in a network has implications not only for the observation of near-real-time earthquake deformation and thus earthquake early warning, but also for instrument and network state-of-health.

To compute the second derivative, we first determine the sampling rate of data (different streams may have a different sampling rate) and then we apply eq. (1):

$$f''(x_1) = \frac{f(x_0) - 2f(x_1) + f(x_2)}{h^2} \quad (1)$$

where

$$h = x_2 - x_1 = x_1 - x_0$$

In our case, if any of $f(x)|x \in \{x_0, x_1, x_2\}$ does not exist, the second derivative for the point x_1 is not being computed.

The Allan variance is computed by eq. (2), where $\sigma_y^2(x, \tau)$ is the computed Allan variance, y is the signal function, and τ is the time interval considered to compute the function.

$$\sigma_y^2(x, \tau) = \frac{(y(x - \tau) - 2y(x) + y(x + \tau))^2}{2\tau^2} \quad (2)$$

Since we took the value of τ to be equal with the sampling rate, we observe that:

$$\sigma_y^2(x, \tau) = \frac{(\tau y'')^2}{2} \quad (3)$$

According to eq. (3), the Allan variance and the second derivative are equivalent in meaning. More specifically, the Allan variance is always positive and it differs from the second derivative by a constant.

We plot eq. (2) by choosing a fixed x and letting the τ vary. The fixed x is provided by the user, as a result of a mouse click in the chart. Based on chart scaling, we depict the variation of time coordinate x in Fig. 6.

5 Conclusion and Open Problems

Our application runs on both Windows and Linux server, on all major browsers. Through the web interface, we made PANGA data accessible to a plurality of remote users. To cover

a wider range of users, we plan to implement the same dashboard as a native Android and Apple iOS (iPhone and iPad) application, so that availability of PANGA data to be truly ubiquitous.

One challenge to producing a fast and reliable classification technique of earthquakes is the lack of training data. Major earthquakes large enough to produce significant ground deformation across a large region are rare and at this time the only set of data of with high sampling rates from a large and densely-instrumented GPS network is from the 2011 Tōhoku event sequence. Therefore, it is of major interest to record the displacement data measurement in order to be used for future data mining applications.

Another challenge is that there are significant noise and data quality issues with GPS data streams. Some problems are common with those of seismic networks, such as telecommunications glitches and outages. Others are unique to GPS data acquisition and processing. For example, some atmospheric phenomenon can perturb GPS signals. Correlation with seismographs and atmospheric sensors may be key elements in a system to identify earthquakes and perform a rapid computation of an event's magnitude.

It is an open problem to implement a classifier for these big noisy data GPS streams. Such a classifier should be able to monitor and disseminate in real-time especially slow landslides, hard to detect by seismographs, arising from earthquakes, volcanic eruptions, and other hazards.

Bibliography

- [1] A.C. Aguiar, T.I. Melbourne, and C.W. Scrivner (2009), Moment release rate of Cascadia tremor constrained by GPS, *Journal of Geophysical Research*, Jul. 2009, 114:1-11.
- [2] W. Bertiger, S.D. Desai, B.Haines, N. Harvey, A.W. Moore, S. Owen, J.P. Weiss (2010), Single receiver phase ambiguity resolution with GPS data, *Journal of Geodesy*, May 2010, 84:327–337.
- [3] B. Franks (2012), *Taming The Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics*, Wiley and SAS Business Series, Wiley.
- [4] R.T. Fielding (2000), *Architectural styles and the design of network-based software architectures*, Ph.D. dissertation.
- [5] D. Crockford (2006), The application/json Media Type for JavaScript Object Notation (JSON), IETF, RFC 4627, 7.
- [6] M. Simons, S.E. Minson, A. Sladen, F. Ortega, J. Jiang, S.E. Owen, L. Meng, J.-P. Ampuero, S. Wei, R. Chu, D. V. Helmberger, H. Kanamori, E. Hetland, A. W. Moore, and F. H. Webb (2011), The 2011 magnitude 9.0 Tohoku-Oki earthquake: Mosaicking the megathrust from seconds to centuries, *Science*, Jun. 2011, 332(6036):1421–1425.
- [7] A. Goncalves (2010), *Beginning Java EE 6 with GlassFish 3 (Expert's Voice in Java Technology)*, Apress.
- [8] O. Varaksin, M. Caliskan (2013), *PrimeFaces Cookbook*, Packt Publishing.
- [9] G. Svennerberg (2010), *Beginning Google Maps API 3 (Expert's Voice in Web Development)*, Apress.
- [10] S. Weidler (2008), *Statistiken für Warehouse Management Systeme: Entwicklung eines Statistiktools mit Hilfe des JAVA-Frameworks JFreeChart*, Vdm Verlag Dr. Müller.

- [11] D.W. Allan, J.H. Shoaf, D. Halford (1974), Statistics of Time and Frequency Data Analysis, in *Time and Frequency: Theory and Fundamentals*, B. E. Blair (Ed.).
- [12] D. Allan (1965), *Statistics of Atomic Frequency Standards*, University of Colorado.
- [13] H. Hellwig (1972), I. for Basic Standards (U.S.). Time, and F. Division, *Frequency standards and clocks: a tutorial introduction*, ser. NBS technical note, Dept. of Commerce, National Bureau of Standards, Institute for Basic Standards, Time and Frequency Division.
- [14] A. Mao, C.G.A. Harrison, T.H. Dixon (1999), Noise in GPS coordinate time series, *Journal of Geophysical Research: Solid Earth*, 104(B2):2797–2816.
- [15] K. Larson, GPS seismology (2009), *Journal of Geodesy*, 83(3-4):227–233.
- [16] Z. Malkin, A. Voinov (2001), Preliminary results of processing euref network observations using a non-fiducial strategy, *Physics and Chemistry of the Earth*, Part A: Solid Earth and Geodesy, 26(68):579–583.
- [17] P. Elósegui, J. L. Davis, D. Oberlander, R. Baena, G. Ekström (2006), Accuracy of high-rate GPS for seismology, *Geophysical Research Letters*, 33(11):1-4.
- [18] C. Roberts, P. Morgan, C. Rizos (2002), Allan variance applied to time series baseline results for gps-based deformation monitoring applications, in *Proceedings of the 2nd symposium on geodesy for geotechnical and structural applications, Berlin, May 2002*, 299–311.
- [19] Z. Malkin (2011), Study of astronomical and geodetic series using the allan variance, *Kinematics and Physics of Celestial Bodies*, 27(1):42–49.