Open-source Precise Point Positioning (PPP) with Ginan v2

Simon McCLUSky
Senior GNSS Scientist
simon.mcclusky@ga.gov.au
Locate 23 11 May 2023
Overview

• What is Ginan
• How it works
• Current capabilities
• Ginan in operations
• Case Study – “Ginan in a box”
• Project timeline – What’s next
The Ginan Name

Ginan comes from the Wardaman people of Northern Territory

Is a Wardaman word for a red dilly-bag filled with songs of knowledge

Is the fifth-brightest star in the Southern Cross

The Southern Cross helped the First Australians to navigate
Positioning Australia (PA)
National Positioning Infrastructure Capability (NPIC)

“Accurate and reliable positioning for everyone”
Ginan - Geoscience Australia’s (GA) GNSS Analysis Centre Software

- Part of GA’s Positioning Australia (PA) National Positioning Infrastructure Capability (NPIC)
- Open-source software toolkit for precise positioning and navigation
- Multi-GNSS data processing and analysis capability
- Undifferenced, State Space Representation (SSR) using Precise Point Positioning (PPP) methodology
- Capable of delivering precise positioning products and services for post processed and real-time applications
- Enables centimetre level accuracy positioning in areas with mobile phone/internet coverage
Methodology - PPP (SSR) vs RTK (OSR)

\[ P_{r,f} = \rho_r + c \left( \delta_T q - \delta_T s \right) + \delta_T \rho_r + \mu_f I_r + \mu_f I_f + \delta r_f + \delta f + \delta s + \varepsilon_r \]

- **P** - Observed range between satellite & receiver
- **ρ** - Geometric distance between satellite and receiver *
- **c** - Speed of light
- **δτ** - Receiver and satellite clock offsets
- **T** - Troposphere delay between satellite and receiver
- **μ** - Frequency dependent ionosphere delay factor
- **I** - Ionosphere delay between satellite and receiver
- **d** - Receiver and satellite hardware signal biases
- **O** - Satellite orbit errors
- **ε** - Range measurement noise

s, r, f, q – satellite, receiver, frequency, constellation indices

* The geometric range ρ is what is used for positioning
More precisely known ρ’s == more precise positioning
Ginan – Functional Architecture

- **Configuration:**
  - Standard Yet Another Markup Language (YAML)

- **Input:**
  - File based for Post Processing (PP) or Streams for Real Time (RT) processing

- **Observations:**
  - Always Un-Differenced (UD)
  - Combined Ionosphere Free (IF) form, or Un-Combined (UC)
  - Dual frequency (IF), or Multi-frequency UC
  - Multi-constellation: GPS, GLO, GAL, QZS, BDS (SBAS in dev)

- **Measurement model:**
  - Positions, Clocks, Phase/Code biases, Troposphere, Ionosphere, PCO, PCV, phase windup, Antenna Ecc, Tides, Relativity, …….

- **Filtering and Estimation:**
  - Robust Kalman filter
  - Flexible full GNSS observation model State estimation
  - Backwards Smoothing (Fixed Lag and Full RTS)

- **Output:** Industry standard file products or RTCM3 stream based

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Data Input</th>
<th>Modelling + Prediction</th>
<th>Filtering + Estimation</th>
<th>Error Handling</th>
<th>Augmentation</th>
<th>Products Output</th>
<th>Scripts + Visualisation</th>
</tr>
</thead>
</table>

© Commonwealth of Australia (Geoscience Australia) 2023.
## Ginan - Development & Operations timeline

<table>
<thead>
<tr>
<th>Ginan v1</th>
<th>Ginan v2.0</th>
<th>Ginan v3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>Alpha</td>
<td>Alpha</td>
</tr>
<tr>
<td>February</td>
<td>Beta</td>
<td>Beta</td>
</tr>
<tr>
<td>March</td>
<td>v2.0</td>
<td>v3.0</td>
</tr>
</tbody>
</table>

### Ginan v1
- Ginan v1.5.4 source code: Available GitHub, supported
- Ginan v1.5.4 correction products: Available GitHub, no longer supported

### Ginan v2.0
- Ginan v2.0 source code: Available GitHub, supported
- Ginan v2.0 to operations: Available GA data repository, NTrip, supported
- Ginan v2.0 correction products: [Phase 1 and 2 messages] (Available GitHub, supported)

### Ginan v3.0
- Ginan v3.0 source code: Available GitHub, supported
- Ginan v3.0 to operations: Available GA data repository, NTrip, supported
- Ginan v3.0 correction products: [Phase 1 and 2 messages] (Available GitHub, supported)
Ginan: v1 vs v2

❖ Unified User and Network operation modes (One Observation Model & Filter)
❖ More GNSS constellations – Full Multi-Constellation capability (Ex SBAS)
❖ Better internal frequency indexing (complete Multi-Frequency capability)
❖ UnDifferenced / UnCombined (UDUC) processing (v1 was Combined IF only)
❖ CPP integrated and coupled Precise Orbit Determination (POD) capability
❖ More robust data handling in filter cycle slip and outlier detection and removal
❖ Complete RTCM3 phase 1 and Phase 2 message decoding and encoding
❖ SLR data handling fully implemented
❖ Model & Performance improvements
Ginan Performance: v1 vs v2

Dual frequency uncombined PPP vs IF PPP in Ginan (ie. v1 vs v2)

<table>
<thead>
<tr>
<th></th>
<th>v1</th>
<th>v2</th>
</tr>
</thead>
<tbody>
<tr>
<td>99.9%</td>
<td>95.0%</td>
<td>68.0%</td>
</tr>
</tbody>
</table>

Dual Frequency uncombined PPP (AR) – Multi constellation

Dual Frequency uncombined PPP – Multi constellation (v2)
Operational Ginan system is built on top of the infrastructure used to maintain GA’s NPIC data repository

A Kubernetes (k8s) computing “cluster” via Amazon (AWS EKS) is used to run Ginan in individual "pods"

Resources on the cluster are controlled by Terraform (Infrastructure-as-code)
File-based Products

The files we output are created by Ginan five times per day:

❖ **Daily** for the **Rapid** Orbits, Clocks and CORS Positions
❖ **6-hourly** for the **Ultra-rapid** Orbits and CORS Positions

The process is split into **individual steps** or tasks controlled by **Argo**

**Argo** is an orchestration / workflow engine used within Kubernetes to create the necessary resources to run the "Ginan jobs"
Real-Time Services

We run Ginan continuously (multiple instances) to:

1. Produce an **RTCM correction stream** broadcasting **1059 and 1060** messages

2. Output **real-time PPP results** from a handful of CORS stations

   • Both of these are modules directly described by Terraform code
     • No scheduling / orchestration needed for real-time like with files
Ginan Information

Source code is available at GitHub and we support native builds on:

❖ Linux
❖ Mac
❖ Windows (via WSL – Windows Subsystem for Linux)

Docker Image

Playlist of Installation Videos

Links available at the Link Tree: https://linktr.ee/ginan_GA
Ginan Use Cases

16 use cases identified to date + 1: Cal-Val

- Giving precise positioning to the Internet of Things (IoT) for new applications
- As an aid to teaching GNSS technology for space navigation and surveying courses
- As a toolkit to help solve complex position, navigation and timing research challenges
- Work to maintain and improve Australia’s geodetic datums
- Improve the performance of crustal movement and earthquake monitoring
- To monitor the performance of networks of continuously operating reference stations (CORS)
- Detection of geohazards such as tsunamis, cyclones and space weather events through ionospheric disturbance monitoring
- On-selling correction products and streams with value-adding services
- Calculating precise positions for general surveying, mapping and spatial related purposes
- Making the user platform part of systems to bring precise positioning to mobile consumer devices e.g. phones, tablets. (Android, iOS)
- Running the user platform as embedded software for autonomous vehicles (land, sea, air, space)
- Precise orbit determination for LEO satellite fleets
- Ground truthing objects to assist with space situational awareness
- An alternative source of positioning data to check the performance of SouthPAN
- As part of a system to monitor the performance of GNSS signals over the South Pacific
- Use products such as the ZTD file to improve weather forecasting and climate change monitoring

Ginan open source software and correction products and streams

Academia

Industry

Geoscience

Government
Ginan in a Box
Embedded software for kinematic positioning and Navigation

Ginan embedded configuration
Ginan in a box

- Ginan running on a Raspberry Pi Arm64 processor
- Septentrio Mosaic Multi-GNSS receiver
Albert Park, Melbourne - Ginan in a Box kinematic positioning test

Ginan Multi-GNSS PPP wrt reference differential RTK trajectory

Horz RMS = 0.021m
Vert RMS = 0.109m

*After removing the stationary convergence data
Ginan v3 – What’s Next

❖ Compact SSR / IGS SSR correction formats
❖ PPP-RTK capability
❖ LEO obit estimation capability
❖ Multi-solution combination capability
❖ Web / Graphical User Interface
❖ Much more testing and validation