Australian Government
Geoscience Australia

GINAN

The GNSS Science Manual

Ginan documentation

March, 2022

Document approvals

Role	Name	Signature	Date
Prepared:	Aaron Hammond	–	18-3-2022
Prepared:	Ken Harima	–	18-3-2022
Approved:	Simon McClusky	–	18-3-2022

Document history

Version	Dated	Author	Notes
Beta	18-3-2022	Aaron Hammond, Ken Harima	Ginan Beta release
–	–	–	–

Contents

1 Introduction

1.1 The Positioning Australia Program

The Australian Government is making a significant investment in the Positioning Australia program through Geoscience Australia. The program contains three major projects:

- The commercial procurement and operation of a Satellite Based Augmentation System (SBAS) called SouthPAN which will enhance positioning across the region through the provision of extra GNSS signals and data delivered from a geostationary satellite.
- The enhancement of the National Positioning Infrastructure Capability (NPIC) which will see upgrades to and an expansion of the Global Navigation Satellite System (GNSS) Continuously Operating Reference Station (CORS) network across the South Pacific and Antarctica.
- Ginan is an open source Precise Point Positioning (PPP) toolkit. It can produce PPP position correction products and, operating in another mode, use GNSS observations and those correction products to determine positions with an accuracy in the centimetre range.

The program is summarised in figure 1 below.

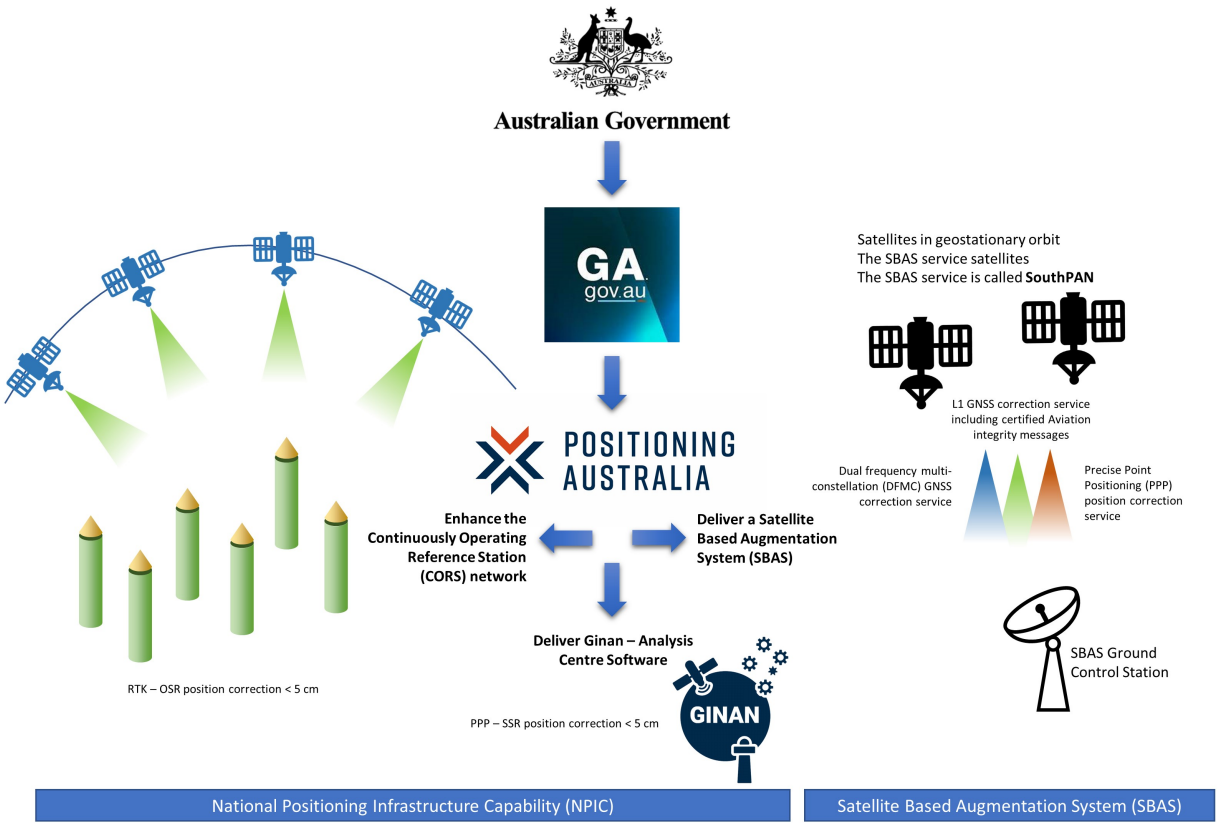

Figure 1.1: The three main projects in the Positioning Australia program.

1.2 Ginan - Analysis Centre Software

Ginan, is being rolled out in a phased approach and will offer products in four distinct categories:

- The software itself. Ginan is open-source software that GA has hosted on a GitHub repository.
- Standard precise point positioning (PPP) product files. An operational version of Ginan, maintained by Geoscience Australia (GA), will produce on a 24 X 7 basis, a range of standard PPP product files including, for example, a precise orbits and clocks file in SP3 format.
- Precise point positioning correction messages. An operational version of Ginan, maintained by GA, will stream over the internet on a 24 X 7 basis, a range of PPP correction messages in the RTCM3 message format.
- New PPP products and applications yet to be defined. The Ginan toolkit gives GA the ability to offer new PPP products, yet to be defined, but which, in collaboration with users, may spawn new applications and commercial opportunities.

Ginan is summarised in figure 2 below.

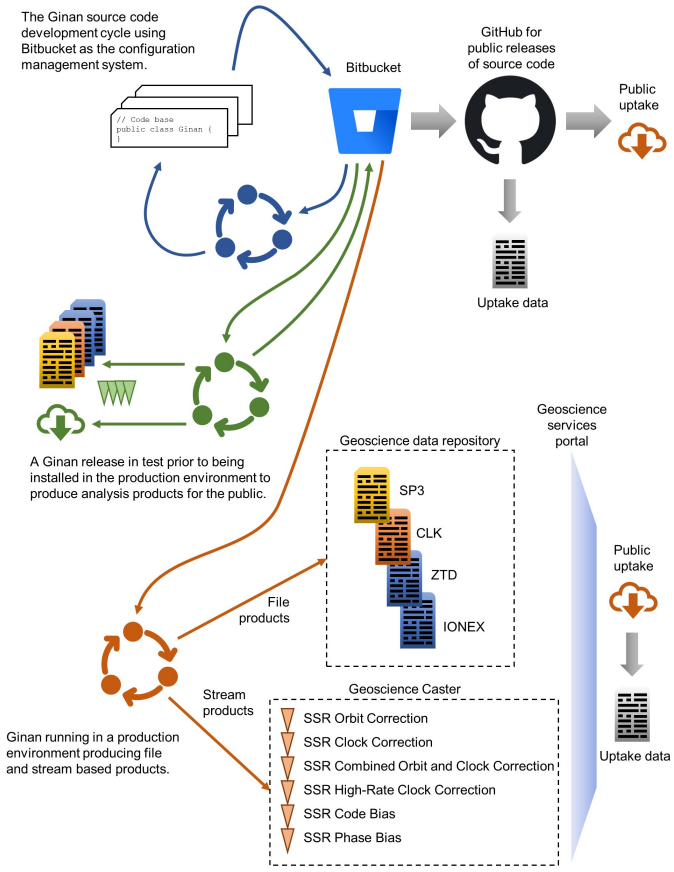

Figure 1.2: The Ginan product offering.

1.3 This document

This document forms part of the Ginan documentation suite. Ginan is a software toolkit aimed at meeting the rigorous processing requirements to support high precision geodetic positioning, as well as supporting a larger range of general GNSS positioning applications. This documentation intends to give insight into theoretical aspects of GNSS data processing. It assumes that the reader has a good understanding of the concepts and principles behind GNSS positioning.

If you are new to the science of GNSS positioning there are many great resources available for free on the internet. One great place to start is the Ginan support website: https://geoscienceaustralia.github.io/ginan

2 Equation Conventions

In this manual we will be adhering to the following conventions:

2.1 List of Symbols

i or r	Receiver identification
j or s	Satellite identification
k or t	Epoch number
q	GNSS type (GPS,GALILEO,GLONASS,QZSS)
c	Speed of light [m/s]
x	Vector of parameters to be estimated
y	Vector of observations
v	Vector of residuals
H	Design matrix
P	Covariance matrix
σ	Standard deviation of observable
Δ	Increment to a priori values [m]
λ or $\lambda_1, \lambda_2, \lambda_5$	Wavelength
f_1, f_2, f_5	frequency
N	Ambiguity or N Real valued ambiguity and z Integer part of real valued ambiguity
α	level of significance
d	Code Biases
b	Phase Biases
z	(Integer) Carrier phase ambiguities
dt	Clock error [s]
κ	Correction - relativity
ι or I	Ionosphere
τ or T, T_h, T_w	Troposphere
m_W	elevation dependent mapping function for the troposphere hydrostatic delay
m_W	elevation dependent mapping function for the troposphere wet delay
ξ	Phase wind-up error
ϵ	Error in observations and unmodelled effects [m]
ρ_i^j	Geometric distance between satellite and receiver
L_i^j	Carrier phase observable (times c) [m]
P_i^j	Pseudo range observable [m]

Example: for an undifferenced, uncombined float solution, the linearized observation equations for pseudorange and phase observations from satellite s to receiver r can be described as:

$$\Delta P^s_{r,f} = u^s_r.\Delta x + c.(dt^q_r - dt^s) + M^s_r.T_r + \mu_f.I^s_r \qquad +d^q_{r,f} - d^s_f + \epsilon^s_{P,f} \tag{2.1}$$

$$\Delta L^s_{r,f} = u^s_r.\Delta x + c.(\delta t^q_r - \delta t^s) + M^s_r.T_r - \mu_f.I^s_r + \lambda_f.N^s_{r,f} + b^q_{r,f} - b^s_f + \epsilon^s_{L,f} \tag{2.2}$$

where $\Delta P^{q,s}_{r,f}$ and $\Delta \phi^{q,s}_{r,f}$ are the respective pseudorange and phase measurements on the frequency f(f=1,2), from which the computed values are removed; $u^{q,s}_r$ is the receiver-to-satellite unit vector; Δx is the vector of the receiver position corrections to its preliminary

position; dt_r^q and $dt^{q,s}$ are the receiver and satellite clock errors respectively; c is the speed of light in a vacuum $M_r^{q,s}$ is the elevation dependent mapping function for the troposphere wet delay from the corresponding zenith one T_r; $I_{r,1}^{q,s}$ is the ionosphere delay along the line-of-sight from a receiver to a satellite at the first frequency and $mu_f^q = (\lambda_f^q/\lambda_1^q)^2$; λ_f^q is the wavelength for the frequency f of a GNSS q; $z_{r,f}^{q,s}$ is the phase ambiguity $d_{r,f}^q$ and $b_{r,f}^q$ are the receiver hardware delays of code and phase observations respectively; $d_f^{q,s}$ and $b_f^{q,s}$ are the satellite hardware delays of code and phase observations, respectively; $\epsilon_{P,f}$ and $\epsilon_{L,f}$ are the code and phase measurement noises respectively.

3 Observation Modelling

The Ginan toolkit is based on the concept of Precise Point Positioning (PPP). PPP is a high accuracy positioning method that seeks to correct errors in GNSS positioning and thus improve accuracy. Unlike differential GNSS techniques, which seek to measure GNSS errors using a nearby reference station, PPP is based in the robust modelling and estimation of systematic errors in the GNSS signals. In PPP, the GNSS measurements are modelled as normally distributed random variables with mean:

$$E(P^s_{r,f}) = \rho^s_r + c(dt^q_r - dt^s) + \tau^s_r + \mu_f I^s_r + d^q_{r,f} + d^s_f \tag{3.1}$$

$$E(L^s_{r,f}) = \rho^s_r + c(dt^q_r - dt^s) + \tau^s_r - \mu_f I^s_r + b^q_{r,f} - b^s_f + \lambda_f z^s_{r,f} + \phi^s_{r,f} \tag{3.2}$$

and a constant or elevation dependent variance.

$$\sigma(P^s_{r,f}) = \sigma_0 / sin^2(\theta_{el}) \tag{3.3}$$

In equations 3.1 and 3.3,

$P^S_{r,f}$	represents the pseudorange measurements (m) between satellite s and receiver r for carrier fr
$L^s_{r,f}$	represents the carrier phase measurements (m)
$E()$	notates the expectation and $\sigma()$ the variance.
ρ	the geometric distance (m)
c	speed of light (m/s)
dt^q_r	receiver clock offset (s)
dt^s	satellite clock offset (s)
τ^s_r	slant troposphere delay between satellite s and receiver r (m)
μ_f	Ionosphere delay factor for frequency f
I^s_r	slant Ionosphere delay for s and receiver r (m)
$d^q_{r,f}$	receiver hardware bias for pseudoranges (m)
d^s_f	satellite hardware bias for pseudoranges (m)
$b^q_{r,f}$	receiver ionosphere-free phase bias (m)
b^s_f	satellite ionosphere-free phase bias (m)
λ_f	the signal carrier wavelength for frequency f(m)
$z^S_{r,f}$	carrier phase ambiguity (cycle)

High accuracy GNSS position estimation requires precise estimation of the geometric distance ρ^s_r. The effect of other parameter in 3.1 and 3.2 needs to be eliminated either by modelling/estimating or, in the case of Ionosphere delays, by using linear combinations.

For PPP processing, the geometric distance ρ^s_r is linearized around a-priori values of satellite and receiver positions.

$$\rho^s_r = \sqrt{X^{s-} - X^-_r} + \Delta X^s - \Delta X_r \tag{3.4}$$

where X^{s-} is the a-priory satellite position with respect to the earth centre and X^-_r the receiver/station position.

The satellite position can be assumed known, and read from external sources, or estimated from the GNSS measurements. When estimated by Ginan, the a-priori satellite position is defined as a function of initial satellite position, initial satellite velocity and up to 15 solar radiation pressure parameters as explained in chapter 6. Then the each satellite position component is linearized with respect to the orbit parameters.

$$\Delta X^s = -\dot{e_{rec}} \sum \frac{\partial X^s}{\partial OP_i} \Delta OP_i \tag{3.5}$$

where e_{rec} is the satellite-to-receiver vector, $\frac{\partial X^s}{\partial OP_i}$ is the partial derivative of the satellite position with respect to the orbital parameter (initial condition or SRP) OP_i, and ΔOP_i the difference between the estimated and the a-priori value of OP_i

The station/receiver position can be assumed known (and read from a SINEX file), estimated as a constant (in case of a static receiver) or estimated as random walk (in case of a moving receiver) variable. When estimated, a standard precision position (SPP) is used the a-priori receiver position.

$$\Delta X_r = \dot{e_{rec}} \Delta X_r \tag{3.6}$$

ΔX_r is the difference between the estimated and the a-priori value of X_r. The SPP is calculated each epoch by applying iterative least squares on pseudorange measurements (SPP is initialised at the centre of earth and updated for each of up to a 10 iterations).

Satellite clock offsets can be assumed known, and obtained from external sources, or estimated as a random walk variable. the receiver clock is modelled as a random walk variable.

3.1 Ionosphere free combination

The current version of the Ginan software, the effect of Ionosphere delay is eliminated using the Ionosphere free combination of two measurements. Thus the Ionosphere-free combination of pseudorange

$$P^s_{r,IF} = \frac{\mu_2}{\mu_2 - \mu_1} P^s_{r,1} - \frac{\mu_1}{\mu_2 - \mu_1} P^s_{r,2} = \frac{\lambda_2^2}{\lambda_2^2 - \lambda_1^2} P^s_{r,1} - \frac{\lambda_1^2}{\lambda_2^2 - \lambda_1^2} P^s_{r,2} \tag{3.7}$$

and carrier phase

$$L^s_{r,IF} = \frac{\mu_2}{\mu_2 - \mu_1} L^s_{r,1} - \frac{\mu_1}{\mu_2 - \mu_1} L^s_{r,2} = \frac{\lambda_2^2}{\lambda_2^2 - \lambda_1^2} L^s_{r,1} - \frac{\lambda_1^2}{\lambda_2^2 - \lambda_1^2} L^s_{r,2} \tag{3.8}$$

are used instead of their uncombined versions. The combinations in 3.7 and 3.8 will remove the first order (with respect to frequency) component of the Ionosphere delays, leaving the higher order components known to be smaller then a few centimetres. It also preserves the scale of geometric (positions, clocks and tropospheric) biases.

$$E(P^s_{r,IF}) = \rho^s_r + c(dt^q_r - dt^s) + \tau^s_r + d^q_{r,IF} + d^s_{IF} \tag{3.9}$$

$$E(L^s_{r,IF}) = \rho^s_r + c(dt^q_r - dt^s) + \tau^s_r + b^q_{r,IF} - b^s_{IF} + A^s_{r,IF} + \phi^s_{r,IF} \tag{3.10}$$

However the carrier phase combination will also remove the integer nature of the ambiguity

$$A^s_{r,IF} = \frac{\lambda_2^2\lambda_1}{\lambda_2^2-\lambda_1^2}z^s_{r,1} - \frac{\lambda_1^2\lambda_2}{\lambda_2^2-\lambda_1^2}z^s_{r,2} = \frac{\lambda_1\lambda_2}{\lambda_1+\lambda_2}z^s_{r,1} + \frac{\lambda_1^2\lambda_2}{\lambda_2^2-\lambda_1^2}(z^s_{r,1}-z^s_{r,2}) \tag{3.11}$$

thus another ionosphere-free combination, the Melbourne-Wubenna combination, is used to isolate and estimate the $(z^s_{r,1} - z^s_{r,2})$ ambiguity

$$A^s_{r,IF} = E(\frac{L^s_{r,1}}{\lambda_1} - \frac{L^s_{r,2}}{\lambda_2} + \frac{\lambda_2-\lambda_1}{\lambda_2+\lambda_1}(\frac{P^s_{r,1}}{\lambda_1} + \frac{P^s_{r,2}}{\lambda_2})) = z^s_{r,1} - z^s_{r,2} + d^s_{r,MW} \tag{3.12}$$

3.2 Troposphere modelling

The tropospheric delays are separated into two components: the hydrostatic delay (dependent on temperature and pressure), and the wet delay (dependent also on humidity). Each component can be expressed as the product of a zenith delay and an elevation based mapping.

$$\tau^s_r = m_H(\theta^s_{el,r})\tau_{ZHD,r} + m_W(\theta^s_{el,r})\tau_{ZWD,r} \tag{3.13}$$

In the case of Ginan, the hydrostatic components are assumed to be deterministic, while the zenith wet delay can be set as deterministic or estimated as part of the PPP solution process. The nominal values of Zenith hydrostatic delays $\tau_{ZHD,r}$ and hydrostatic mapping function $m_H(\theta^s_{el,r})$ and the wet mapping function $m_W(\theta^s_{el,r})$ are estimated using either the VMF3 (daily measured) or GPT2/GMF (empirical) models.

When estimated, the tropospheric wet delay is modelled by one variable $\tau_{0,r}$ or three variables

$$\tau_{ZWD,r} = \tau_{0,r} + cot(\theta^s_{el,r})(cos(\theta^s_{az,r})grad_{ns,r} + sin(\theta^s_{az,r})grad_{ew,r}) \tag{3.14}$$

each variable is modelled as a random walk of a First order Gauss-Markov process.

3.3 Hardware bias and ambiguities

Both pseudorange and carrier phase GNSS measurements are known to include the effect of biases product of delays in satellite and receiver hardware. Ideally the PPP algorithm will seek to separate and estimate these biases. The system of equations defined by 3.1 and 3.1 are rank deficient (2 x freq biases + other parameters to be solved using 2 x freq. number of equations) however, and thus only a combination of biases can be solved. In the case of Ginan processing of GNSS measurements, the clock offsets being estimated correspond to the ionosphere-free combination of two pseudorange measurements.

$$\tilde{cdt}_r = cdt_r + d_{r,IF} \tag{3.15}$$

$$\tilde{cdt}^s = cdt^s + d^s_{IF} \tag{3.16}$$

Also, when estimating GNSS error parameters without the help of an underlining Ionosphere model, the geometry free combination of pseudorange measurements is assimilated into the Ionosphere delay estimate

$$\tilde{I^s_r} = I^s_r - \frac{\lambda_1^2}{\lambda_2^2-\lambda_1^2}(d_{r,1} - d_{r,2} - d^s_1 + d^s_2) \tag{3.17}$$

Reparameterising 3.1 and 3.2 to include cdt' and I'^s_r results in

$$E(P^s_{r,f}) = \rho^s_r + c(\tilde{dt^q_r} - \tilde{dt^s}) + \tau^s_r + \mu_f\tilde{I^s_r} \tag{3.18}$$

$$E(L^s_{r,f}) = \rho^s_r + c(\tilde{dt^q_r} - \tilde{dt^s}) + \tau^s_r - \mu_f I'^s_r + \tilde{b^q_{r,f}} - \tilde{b^s_f} + \lambda_f z^s_{r,f} + \phi^s_{r,f} \tag{3.19}$$

where

$$\tilde{b^q_{r,f}} = b_{r,f} + d^q_{r,f} - d_{r,IF} \tag{3.20}$$

$$\tilde{b^s_f} = b^s_f + d^s_f - d^s_{IF} \tag{3.21}$$

The current version of Ginan performs Ionosphere modelling separately, thus the phase biases estimated alongside geometric parameters like clocks and troposphere correspond to 3.20 and 3.20.

When performing Ionospheric delay modelling, as detailed in chapter 7, the Ginan software generates the an Ionosphere delay map from which to estimate I^s_r and differential pseudorange biases $d_{r,1} - d_{r,2}$ and $d^s_1 - d^s_2$. If the end user intends to use both Ionosphere corrections and ambiguity resolution, the following set of biases needs to be used alongside the Ionosphere maps for I^s_r:

$$\hat{d^s_f} = -\frac{\lambda^2_f}{\lambda^2_2 - \lambda^2_1}(d^s_1 - d^s_2) \tag{3.22}$$

$$\hat{b^s_f} = \tilde{b^s_f} - \hat{d^s_f} \tag{3.23}$$

3.4 Deterministic biases

Aside from the biases/errors described above, the Ginan software accounts for a number of biases by calculating then using deterministic models. Solid tide corrections can be applied when estimating the position of static stations. These corrections can include the effect of polar motion and ocean loading. Relativitistic effects on satellite clock are estimated according to the GPS interface control document. Receiver side effects are not accounted for. Relative phase windup effects are estimated from standard satellite attitude models. Antenna phase centre and antenna phase variance characteristics of both satellite and receiver side antennas are obtained from ANTEX files.

4 Kalman Filtering

The Kalman filter is an algorithm that processes observation data over time to produce precise estimates of unknown parameters that may not be directly observable. Named after Rudolf E.Kálmán, the great success of the Kalman filter is due to its small computational requirement, elegant recursive properties, and its status as the optimal estimator for one-dimensional linear systems with Gaussian error statistics. Kalman filtering is used in a wide range of applications include global positioning system receivers, control systems, smoothing the output from laptop trackpads, and many more. The PEA and POD software utilities are fundamentally an application-specific Kalman filter that takes observations (namely, GNSS signals) and fuses them together to provide robust estimates of parameters of interest.

4.1 Overview of Kalman Filtering

Kalman filters are typically used to estimate parameters which change with time. A Kalman filter has measurements y_t, with unknown error ϵ_t, and a state vector x_t (or a parameter list) which have specified statistical properties.
The observation equation at time t is given by:

$$y_t = H_t x_t + \epsilon_t \tag{4.1}$$

The state transition equation is given by:

$$x_{t+} = F_t x_t + w_t \tag{4.2}$$

Kalman filter processing is broken up into two main steps: *Prediction* and *Update*.

4.1.1 Prediction Step

The *Prediction* step uses a model to 'predict' the parameters at the next data epoch. The state transition matrix F projects the state vector (parameters) forward to the next epoch, as follows:

$$\hat{x}_t^{t-1} = F_t \hat{x}_{t-1}^{t-1} \tag{4.3}$$

where $\hat{x}_t$ is the state vector and F_t is the state transition matrix. Subscripts denote the time that a quantity refers to, and superscripts denote the time of the most recent observation used to estimate the quantity. E.g. $\hat{x}_t^{t-1}$ is the state vector estimate for time t, using observations up to and including time $t-1$.
Similarly, the state transition matrix is used to project the state covariance matrix (the uncertainty of the state vector) forward in time:

$$P_t^{t-1} = F_t P_{t-1}^{t-1} F_t^{\intercal} + Q_t \tag{4.4}$$

where P is the state covariance matrix and Q_t is the process noise covariance matrix. Process noise (Q matrix) is added to reflect the increase in uncertainty caused by projecting states forward in time. Deterministic parameters have Q elements of 0, and stochastic parameters have positive Q elements.
The state transition matrix F can take several forms depending on the nature of the state being estimated - e.g.:

- For random walk states: F = 1
- For states with rate terms: F is the matrix $\begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix}$
- For first-order Gauss-Markov (FOGM) states: $F = e^{-\delta t \beta}$
- For white noise: F = 0

The Kalman gain is the matrix that allocates the differences between the observation at time t+1 and their predicted value at this time based on the current values of the state vector according to the noise in the measurements and the state vector noise.

4.1.2 Update Step

The *Update* step 'updates' the predicted states with observation data from the current epoch.
Firstly, the prefit residual is calculated, which measures the error between the actual observations and the expected observations given the predicted state vector:

$$\hat{y}_t = z_t - H_t \hat{x}_t^{t-1} \tag{4.5}$$

where $\hat{y}_t$ is the prefit residual, z_k is the observation vector (list of observations made this epoch) and H_t is the design matrix. The design matrix transforms from state-space into observation-space - i.e. $H_t\hat{x}$ is the observation you would get if you perfectly observed the state $\hat{x}$.
The corresponding prefit residual covariance matrix is given by:

$$S_t = H_t P_t^{t-1} H_t^T + R_t \tag{4.6}$$

where R_t is the measurement noise matrix - the covariance matrix corresponding to the observation vector.
The optimal Kalman gain is then calculated:

$$K_t = P_t^{t-1} H_t^T + S_t^{-1} \tag{4.7}$$

which is then used to calculate the optimal adjustment to the predicted states to get the updated states:

$$\hat{x}_t^t = \hat{x}_t^{t-1} + K_t \hat{y}_t \tag{4.8}$$

and corresponding updated state covariance matrix:

$$P_t^t = (I - K_t H_t) P_t^{t-1} \tag{4.9}$$

4.2 Comparison between Weighted Least Squares and Kalman Filtering

- In Kalman filtering, apriori constraints must be given for all parameters. This is not needed in weighted least squares, but can also be done.

- Kalman filters can allow for 0 variance parameters, this cannot be done in WLS, as this requires the inversion of the constraint matrix.
- Kalman filters can allow for a method of applying absolute constraints, WLS can only tightly constrain parameters.
- Kalman filters are more prone to numerical stability problems, and take longer to run (they have more parameters).
- Process noise models can be implemented in WLS, but they are computationally slow.

4.3 Implementation in the PEA

4.3.1 Robust Kalman Filter Philosophy

It is well known that the Kalman filter is the optimal technique for estimating parameters of interest from sets of noisy data - provided the model is appropriate.
In addition, statistical techniques may be used to detect defects in models or the parameters used to characterise the data, providing opportunities to intervene and make corrections to the model according to the nature of the anomaly.
By incorporating these features into a single generic module, the robustness that was previously available only under certain circumstances may now be automatically applied to all systems to which it is applied. These benefits extend automatically to all related modules (such as RTS), and often perform better than modules designed specifically to address isolated issues.

4.3.2 Initialisation

When parameters' initial values are not known a-priori, it is often possible to determine them using a least-squares approach.
To minimise processing times, the minimal subset of existing states, measurements, and covariances are used in least-squares estimation whenever the initial value and variance of a parameter is unspecified.
For rate parameters, multiple epoch's worth of data are required for an ab-initio initialisation. This logic is incorporated into the filter and is applied automatically as required.

4.3.3 Outlier detection, Iteration, and Hypothesis Testing

As a statistical machine, the Kalman filter is capable of detecting measurements that do not fit within the system as modelled.
In these cases, the model may be adjusted on-the-fly, to allow all measurements to be continued to be used without contaminating the results in the filter.
A typical example of a modelling error in GNSS processing is a cycle-slip, in which the ambiguity term (which usually modelled with no change over time) has a discontinuity. Other examples may include clock-jumps or satellite burns.
Hypotheses are to be generated for any measurements that are statistical outliers, and the model iterated as required.

4.3.4 Performance Optimisation

The inversion of large matrices as required by the Kalman filter easily dominates the processing time required during operation. Techniques are available to reduce, and distribute this processing burden across multiple processors.

The Eigen library is used for algebraic manipulation which allows for automatic parallelisation of vector algebra, and improves code robustness by checking matrix dimensions while in use.

4.3.4.1 Chunking

By dividing measurements into multiple smaller sub-matrices, the long inversion times may be reduced, as the inversion order is of $O(n^3)$

4.3.4.2 Blocking

By separating the filter covariance matrix into a block-diagonal form, individual blocks of the filter may be processed individually, without degradation in accuracy. This may improve performance, and may also enable blocks that are relatively independent to be processed separately, albeit with some degradation in accuracy.

4.4 Configuration

All elements within the Kalman filter are configured using the yaml configuration file, and use a consistent format.

default_filter_parameters

```
1
2  default_filter_parameters:
3
4      stations:
5
6          error_model:         elevation_dependent          #uniform
       elevation_dependent
7          code_sigmas:         [0.15]
8          phase_sigmas:        [0.0015]
9
10         pos:
11             estimated:           true
12             sigma:               [0.1]
13             proc_noise:          [0] #0.57 mm/sqrt(s), Gipsy default value
       from slow-moving
14             proc_noise_dt:       second
15             #apriori:                                  # taken from other
       source, rinex file etc.
16             #frame:               xyz #ned
17             #proc_noise_model:    Gaussian
18
19         clk:
20             estimated:           true
21             sigma:               [0]
22             proc_noise:          [10]
23             proc_noise_dt:       second
24             #proc_noise_model:    Gaussian
25
26         clk_rate:
27             estimated:           false
28             sigma:               [500]
29             proc_noise:          [1e-4]
30             proc_noise_dt:       second
31             clamp_max:           [+500]
32             clamp_min:           [-500]
33
34     satellites:
```

```
35
36          clk:
37              estimated:          true
38              sigma:              [1000]
39              proc_noise:         [1]
40              #proc_noise_dt:      min
41              #proc_noise_model:   RandomWalk
42
43          # clk_rate:
44          #     estimated:          true
45          #     sigma:              [10]
46          #     proc_noise:         [1e-5]
47          #     # clamp_max:          [+500]
48          #     # clamp_min:          [-500]
49
50          orb:
51              estimated:          false
52
53      eop:
54          estimated:  true
55          sigma:      [40]
56
57
58  override_filter_parameters:
59
60      stations:
61          #ALIC:
62              pos:
63                  sigma:              [0.001]
64                  proc_noise:         [0]
```

Listing 4.1: Filter Parameters:

The majority of estimated states are configured in this section. These configurations are applied to all estimates unless another configuration overrides these parameters in the override_filter_parameter section.
The parameters that are available for estimation include:

- stations: - per station parameters
  - **pos** - position of station
  - **pos_rate** - rate of change of position of station (velocity)
  - **clk** - clock offset of station‘
  - **clk_rate** - rate of change of clock offset of station (clock skew)
  - **amb** - carrier phase ambiguity
  - **trop** - vertical troposphere delay at station
  - **trop_grads** - gradients of troposphere in North and East components
- satellites: - per satellite parameters
  - **pos** - position of satellite (coming soon)
  - **pos_rate** - rate of change of position (coming soon) of satellite (velocity)
  - **clk** - clock offset of satellite

  - **clk_rate** - rate of change of clock offset of satellite (clock skew)
  - **orb** - orbital corrections to be used with POD module.

- **eop** - earth orientation parameters (polar motion, delta length of day)

estimated:
Boolean to add the state(s) to the Kalman filter for estimation.

sigma:
List of a-priori sigma values for each of the components of the state.
If the sigma value is left as zero (or not initialised), then the initial variance and value of the state will be estimated by using a least-squares approach. In this case, the user must ensure that the solution is likely rank-sufficient, else the least-squares initialisation will fail. For states with multiple elements (eg, X,Y,Z positions), multiple sigma values may be added to the list. However, if insufficient values are added to the list, the initialiser will use the last value in the list for any extra elements. ie. Setting `sigma: [10]` is sufficient to set all x,y,z components of the apriori standard deviation to 10.

proc_noise:
List of process noises to be added to the state during state transitions. These are typically in m/sqrt(s), but different times may be assigned separately. As for the sigma list, the last value will be used for any elements exceeding the list length.

proc_noise_dt:
Unit of measure for process noise. May be left undefined for seconds, or using sqrt_second, sqrt_seconds, sqrt_minutes, sqrt_hours, sqrt_days, sqrt_weeks, sqrt_years.

override_filter_parameters:
In the case that a specific station or satellite requires an alternate configuration, or to exclude estimates entirely, the override_filter_parameters section may be used to overwrite selected components of the configuration.

user_filter_parameters, network_filter_parameters:
The internal operation of the Kalman filter is specified in this section. It has a large impact on the robustness, and associated processing time that the filter will achieve.

```
1
2  user_filter_parameters:
3
4      max_filter_iterations:      5 #5
5      max_prefit_removals:        3 #5
6
7      rts_lag:                    -1      #-ve for full reverse, +ve for
     limited epochs
8      rts_directory:              ./
9      rts_filename:               PPP-<CONFIG>-<STATION>.rts
10
11     inverter:                   LLT         #LLT LDLT INV
```

Listing 4.2: Filter Operating Parameters:

max_prefit_removals:
Maximum number of pre-fit residuals to reject from the filter.
After the vector of residuals has been generated and before the filter update stage is computed, the residuals are compared with the expected values given the existing states and design matrix. If the values are deemed to be unreasonable - because the variances of the transformed states and measurements do not overlap to with a 4-sigma level of confidence - then these measurements are deweighted by deweight_factor, to prevent the bad values from contaminating the filter.
These measurements are recorded as being rejected, and may have additional consequences according to other configurations such as phase_reject_limit.

max_filter_iterations:
Maximum number of times to compute the full update stage due to rejections.
This is similar to the max_filter_rejections parameter, but the 4-sigma check is performed with post-fit residuals, which are much more precise.
Rejections that occur in this stage require the entire filter inversion to be repeated, and has an associated performance hit when used excessively.

inverter:
There are multiple inverters that may be used within the Kalman filter update stage, which may provide different performance outcomes in terms of processing time and accuracy and stability.
The inverter may be selected from:

- llt
- ldlt
- inv

outage_reset_limit:
Maximum number of epochs with missed phase measurements before the ambiguity associated with the measurement is reset.

phase_reject_limit:
Maximum number of phase measurements to reject before the ambiguity associated with the measurement is reset.

rts_X:
For details about rts configuration, see section 5

4.4.1 Process Noise Guidelines
Currently in the PEA we have random walk process noise models implemented.
The units are typically in meters, and they are given as σ = $\sqrt{variance}$
For a random walk process noise, the process noise is incremented at each epoch as $\sigma^2 \times dt$ where dt is the time step between filter updates.
If you want to allow kinematic processing, then you can increase the process noise e.g.
proc_noise [0.003]
proc_noise_dt: second

equates to $0.003\frac{1}{\sqrt{s}}$
Or if you wanted highway speeds 100km/hr = 28 m/s
proc_noise [28]
proc_noise_dt: second
A nice value for using VMF as an apriori value is 0.1mm /sqrt(s)

```
trop:
    estimated:          true
    sigma:              [0.1]
    proc_noise:         [0.01]
    proc_noise_dt:      hour
```

4.5 Recommended Reading

1. https://ocw.mit.edu/courses/earth-atmospheric-and-planetary-sciences/12-540-principles-of-the-global-positioning-system-spring-2012/lecture-notes/MIT12_540S12_lec13.pdf

A Kalman filter has measurements y_t, with unknown error ϵ_t, and a state vector $\hat{x}_t$ (or a parameter list) which have specified statistical properties.

5 Rauch–Tung–Striebel (RTS) Smoothing

While the Kalman filter is an optimal solution to computing state estimates from all previous data, better estimates could be obtained if all future data were also incorporated.
The RTS Smoothing algorithm is an approach to determine estimates of states and uncertainties by considering the state transition between two Kalman filtered estimates, smoothing the transition between prior and 'future' data.

5.1 Example

Consider the system of a single particle moving in one dimension. At time t=0, the particle's position is measured to be x=0. The system then evolves with a random walk, with no more measurements until the time t=100.
At t=99, the particles position has a large uncertainty - it has likely moved from its original position, however, with no more data available, its mean expected value remains as x=0.
At t=100, the particles position is again measured, this time as x=10. If this system were monitored using a Kalman filter, the expected position of the particle would be a constant x=0 for the first 99 seconds, before an abrupt change in location at t=100. During the 100 seconds, the variance would increase steadily, before abruptly returning to a low value when the second measurement is taken.
If the Kalman filter measurements were taken in reverse order, with the first measurement at t=100, the variance of the particle would steadily increase going backward in time until t=0, before the second measurement again reduced the variance, this time at t=0.
Using an RTS Smoother effectively combines both forward and backward filtering. At t=1, the variance is low due to the measurement at t=0. Likewise, at t=99, the variance is low due to the measurement at t=100. In this example, in addition to the measurements at t=0 and t=100, the expected mean value at t=50 can be expected to be the midpoint between the two measurements - a result that can not be obtained using Kalman filtering alone.
In order to accurately calculate the expected position and variance at t=50 however, knowledge of the measurement at t=100 was required (50 seconds later). RTS smoothed estimates necessarily lag behind the primary Kalman filter to allow some time for future data to be obtained. The length of this lag determines the effectiveness of the smoothing.

5.2 Usage

```
user_filter_parameters:

    max_filter_iterations:      5
    max_prefit_removals:        3

    rts_lag:                    -1      #-ve for full reverse, +ve for
   limited epochs
    rts_directory:              ./
    rts_filename:               PPP-<CONFIG>-<STATION>.rts

    inverter:                   LLT         #LLT LDLT INV
```

Listing 5.1: RTS Configuration

All Kalman filters in the toolkit are capable of having RTS Smoothing applied. If configured appropriately with an RTS_lag and output files, intermediate filter results will be stored to file for reverse smoothing.
For real-time processing, a small lag may be applied to improve short-term accuracy. After each filtering stage, the new result is propagated backward through time to correct the previous N epochs. Each epoch worth of lag however requires a comparable processing time to an Kalman filter processing stage - a lag of N epochs may slow processing by up to a multiple of N.
For post-processing, the optimal lag is to use all future and past data. This is achieved by first computing the forward solution, before propagating the final results backward through to the first epoch. The processing time required for a complete backward smoothed filter may be less than 2x a non-smoothed filter - considerably faster than a finite lag in real-time.

rts_lag:

Number of future epochs to use in RTS smoothing. A larger lag will give more optimal smoothing results, at the expense of a longer lag before they are calculated, and requiring more processing time per epoch.
A negative value indicates that the entire solution should be smoothed at the conclusion of processing. This will obtain optimal results, with lowest processing time, but is not suitable for real-time applications.

rts_directory:

Directory to output RTS files.

rts_filename:

Filename for RTS files. Multiple intermediate files are generated by RTS smoothing, as well as an additional output files for Kalman filter states and clocks.
During real-time processing with finite RTS applied, the intermediate values stored in the file will be deleted after last use, to prevent excessively large files from being generated. This results in a rolling history of the filter being stored in the file, unlike other output files used in the software, which are rotated to new files at discrete points in time.

6 Orbit Modelling

6.1 Gravitational Force Models

This section missing.

6.2 Non-Gravitational Force Models

6.2.1 Solar Radiation Force Models

The magnitude of the SRP acting on the satellite depends on a wide range of parameters. The distance to the Sun and the position of the satellite with respect to Earth and Sun (regarding possible eclipses) define the intensity of the incoming radiation. The geometry of the satellite, the optical properties of the external surfaces, and the actual orientation with respect to the Sun largely influence the orientation and magnitude of the evolving SRP. Therefore any SRP model depends on an accurate implementation of the satellite orbit, the attitude, and the geometric/physical properties of the satellite structure.

6.2.2 Cannonball

The most basic approach with regard to its analytic development is referred to as the cannonball model. The cannonball model provides a useful, first-order approximation. However, due to its homogeneous material properties and symmetrical shape approximation, we recommend its use as an apriori model, before estimation.

6.2.3 ECOM I

The ECOM I model has been widely used for a cubic-like satellite and is formed by three unit vectors as defined in the following:

$$e_d = r_{sun} - r_{sat}/|r_{sun} - r_{sat}| \tag{6.1}$$

$$e_y = r_z \times e_d/|r_z \times e_d| \tag{6.2}$$

$$e_b = e_d \times e_y \tag{6.3}$$

Where e_d denotes the satellite-sun vector, e_y denotes the vector along the axis of the solar panel, e_b is given by the right-hand rule of ed and ey. The total SRP acceleration $\ddot{r}_{srp}$ is expressed as

$$\ddot{r}_{srp} = D \cdot e_d + Y \cdot e_y + B \cdot e_b \tag{6.4}$$

Where D denotes the total acceleration in ed, Y denotes the total acceleration in ey, B denotes the total acceleration in eb. The D, Y and B can be expressed as

$$D = D_0 + D_C \cdot cos\Delta u + D_S \cdot sin\Delta u \tag{6.5}$$

$$Y = Y_0 + Y_C \cdot cos\Delta u + Y_S \cdot sin\Delta u \tag{6.6}$$

$$B = B_0 + B_C \cdot cos\Delta u + B_S \cdot sin\Delta u \tag{6.7}$$

Where Δu denotes the argument of latitude of the satellite with respect to the Sun.

6.2.4 ECOM II

The ECOM II model has been widely used for an elongate-like satellite and is also formed by ed, ey and eb. The parameters of the ECOM II are different from the ECOM I:

$$\begin{aligned} D = D_0 &+ D_2C \cdot cos2\Delta u + D_2S \cdot sin2\Delta u \\ &+ D_4C \cdot cos4\Delta u + D_4S \cdot sin4\Delta u \end{aligned} \tag{6.8}$$

$$Y = Y_0 \tag{6.9}$$

$$B = B_0 + B_C \cdot cos\Delta u + B_S \cdot sin\Delta u \tag{6.10}$$

6.2.5 ECOM C

The ECOM C model is resulted from the combination of ECOM I and ECOM II. The idea is to add the even periodic terms from the ECOM II to the ECOM I. This is because some Block types of satellites are sensitive to the even periodic terms in the SRP model and this ECOM C model might be potentially applied to multi-GNSS constellations. The parameters of the ECOM C are expressed as

$$\begin{aligned} D = D_0 &+ D_C \cdot cos\Delta u + D_S \cdot sin\Delta u \\ &+ D_2C \cdot cos2\Delta u + D_2S \cdot sin2\Delta u \\ &+ D_4C \cdot cos4\Delta u + D_4S \cdot sin4\Delta u \end{aligned} \tag{6.11}$$

$$Y = Y_0 + Y_C \cdot cos\Delta u + Y_S \cdot sin\Delta u \tag{6.12}$$

$$B = B_0 + B_C \cdot cos\Delta u + B_S \cdot sin\Delta u \tag{6.13}$$

6.2.6 Box Wing

The SRP effect also can be handled by a so-called box-wing model that takes satellite bus areas, solar panel area, satellite attitude and interactions between photons and optical properties. The box-wing model $\ddot{r}_{boxw}$ used for a flat surface of satellite bus with thermal effect can be expressed as

$$\ddot{r}_{boxw} = -(A \cdot S_0)/(M \cdot C)cos\theta[(\alpha + \delta)(e_d + 2/3 \cdot e_N) + 2\rho cos\theta \cdot e_N] \tag{6.14}$$

Where A denotes the cross-section area, S_0 denotes the solar irradince at 1 AU $(1367W/m^2)$, M denotes the mass of satellite, C denotes the speed of light, α denotes the absorption coefficient, δ denotes the diffusion coefficient, ρ denotes the reflection coefficient, e_N denotes the normal vector of the surface, $cos\theta$ denotes the angle between the e_d and e_N.

6.2.7 Antenna Thrust

The navigation antenna produces a re-bouncing acceleration when the signal is transmitted. This is called as antenna thrust, which generates a constant acceleration in the radial direction of satellite orbit and can be model as

$$\ddot{r}_{ant} = W/(M \cdot C) \tag{6.15}$$

Where W denotes the emitted power in watt.

6.2.8 Albedo

The earth radiation pressure (ERP), called albedo, also creates a small acceleration on navigation satellites. The ERP acceleration can be expressed as For satellite bus and solar panel mast

6.3 Transformation between Celestial and Terrestrial Reference Systems

The variational equations obtained from the POD need to be transformed into the terrestrial reference frame so that the adjustments can be made in the ECEF frame that the PEA operates in.

$$[CRS] = Q(t)R(t)W(t)[TRS] \tag{6.16}$$

Where, CRS is the Celestial Reference System TRS is the Terrestrial Reference Systems $Q(t)$ is the Celestial Pole motion (Precession-Nutation) matrix $R(t)$ is the Earth Rotation matrix $W(t)$ is the Polar motion matrix

$$Q(t) = \begin{bmatrix} 1 - aX^2 & -aXY & X \\ -aXY & 1 - aY^2 & Y \\ -X & -Y & 1 - a(X^2 + Y^2) \end{bmatrix} \tag{6.17}$$

$$R(t) = R_2(-\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\Theta & cos\theta & 0 \\ 0 & 01 & \end{bmatrix} \tag{6.18}$$

7 Ionosphere Mapping/Modelling

Ionospheric delay is the most significant nuisance parameter in GNSS processing. The GNSS processing algorithm needs to account for it by estimating, correcting or cancelling its effects. Single frequency receivers needs to be provided with Ionosphere delay information to perform positioning, and the accuracy of its positioning algorithm will be directly affected by the accuracy of the Ionospheric delay information.
GNSS receivers that track signals from multiple carriers with different frequencies, can estimate and or cancel Ionospheric delays, but the process of jointly estimating the Ionospheric delay and carrier phase ambiguity means it requires up to a few hours for the solutions to converge to centimetre-level precision.

7.1 Ionosphere measurements

Ginan version 1 Alpha, uses Ionosphere-free combinations to estimate most of GNSS error parameters. For this reason Ionospheric delays, signal delays due to propagation through the Ionosphere, are not estimated as part of the main Ginan processing. Ionospheric delay measurement and mapping is thus performed as a separate, complementary process. Where orbit/clock estimation use ionosphere-free combinations, Ginan 1.0 use geometry-free combinations as a proxy to Ionosphere delay measurements.

Two types of Ionosphere measurements can be used in Ginan 1.0 for Ionosphere delay mapping. The *Smoothed pseudorange* geometry free measurements are calculated directly from GNSS measurements, independent from the main GNSS processing.

$$\hat{I}_r^s = \frac{\lambda_1^2}{\lambda_2^2 - \lambda_1^2} \left((L_{1,r}^s - L_{2,r}^s) - \overline{(L_{1,r}^s - L_{2,r}^s + P_{1,r}^s - P_{2,r}^s)} \right) \tag{7.1}$$

where $\overline{x}$ is the average of x. If code and phase biases could be considered constant over one satellites visibility arc, then this measurement will asymptote to the biased Ionosphere delay $\tilde{I}_r^s$ (chapter 3). In reality, the variability of code and phase biases as well as errors in the averaging process.

A more precise measurement can be generated if using the results of the main Ginan processing. After ambiguities are resolved and satellite/station biases resolved, the remaining nuisance parameters in the geometry-free combination of phase biases can be eliminated.

$$\tilde{I}_r^s = \frac{\lambda_1^2}{\lambda_2^2 - \lambda_1^2} E \left((L_{1,r}^s - \tilde{b_{1,r}} + \tilde{b_1^s} - \lambda_1 z_{1,r}^s - \phi_{1,r}^s) - (L_{2,r}^s - \tilde{b_{2,r}} + \tilde{b_2^s} - \lambda_2 z_{2,r}^s - \phi_{2,r}^s) \right) \tag{7.2}$$

7.2 Thin layer VTEC maps

Ginan plans to use multiple techniques to map Ionospheric delays in GNSS measurements. So far thin layer, global vertical total electron content maps has been implemented

in Ginan. Single layer thin layer models like Klobuchar and SBAS/IONEX mapping is widely used for GNSS mapping. *Total Electron Content (TEC)* is a measure of the number of electrons/ions in cylinder of $1m^2$ transverse section along the satellite-receiver path. The first order Ionospheric delay seen by a signal while traversing through the Ionosphere is proportional to the TEC and inversely proportional to the square of the frequency.

$$I = \frac{\lambda_f^2 r_e}{2\pi} TEC \tag{7.3}$$

where r_e is the electron radius. The Ionosphere delay used in Ginan I_r^s corresponds to the delay at L1 frequency (1545.75 MHz). Which means the Ionospheric delay will have $I_r^s \approx 0.1687 TEC$ where TEC is in units of TECu, where $TECu = 10^{16} electrons/m^2$. In this mapping/modelling method, the electron content are assumed to be concentrated in thin shells at fixed altitudes. The thin shell model allows the definition of a single piercing point, the point in which the satellite-receiver path intersects the ionosphere shell, for each layer. Assuming that the electron content is constant around the piercing point, the total slant TEC can be approximated as:

$$TEC \approx \sum_{lay} \frac{VTEC_{lay}}{cos\chi_{lay}} \tag{7.4}$$

where χ_{lay} is the angle from zenith and $VTEC_{lay}$ is the vertical total electron content at the piercing point.
The vertical total electron content for each layer is mapped using spherical harmonics.

$$VTEC_{lay}(\varphi_{lay}, \vartheta_{lay}) = \sum_{m,n} Q_{m,n}(\varphi_{lay}) \left(A_{m,n,lay} cos(m\vartheta_{lay}) + B_{m,n,lay} sin(m\vartheta_{lay})\right) \tag{7.5}$$

where the co-latitude φ_{lay} is the angle in latitudinal direction, around an axis orthogonal to both the geographic north and sun direction, between the piercing point and the sun. ϑ_{lay} is the angle in longitudinal direction, rotation axis orthogonal to the axis of co-latitude and the sun. The function $Q_{m,n}$ correspond to the associated Legendre polynomial of degree n and order m, re-parameterized in terms of angles. The ionosphere modelling module of Ginan, use estimates of Ionosphere delays from 7.1 or 7.2 to estimate the Vertical TEC coefficients $A_{m,n,lay}$ and $B_{m,n,lay}$ alongside with the satellite DCB $d_{r,1} - d_{r,2}$ and receiver DCB $d_1^s - d_2^s$

8 Ambiguity Resolution

Estimation and/or resolution of carrier phase ambiguities is central to precise point positioning. Whereas standard precision positioning is performed using the unambiguous pseudorange measurement, these measurements are known to have an accuracy ranging from decimetres to metres. Precise positioning thus rely on the use of carrier phase measurements which have an accuracy of millimetres to a few centimetres, but are ambiguous by an integer number of wavelengths. Thus estimating these ambiguities to centimetre or millimetre level of accuracy is a requirement of precise positioning. Integer ambiguity resolution offers the following advantages, over just real-valued ambiguity estimation.

- Ambiguities resolved to integer ambiguities have no errors, which include the accuracy of the solution, it also make the measurement model more robust to changes in environmental conditions.
- Ambiguity resolution can be attempted when estimates are of around 5cm of accuracy (one 4th of wavelength) reducing their errors to 0. This results in an acceleration of convergence of PPP solutions.
- Ambiguities are constant unless there are cycle slips eliminating the need to estimate then once resolved, this in turn will simplify/accelerate the estimation of other parameters in real time applications.

As with any integer estimation process the ambiguity resolution for GNSS signals in Ginan will follow the steps below

- The ambiguities are estimates as real numbers (with the other parameters).
- Integer values of the ambiguities are resolved using the results of step 1 (the real-value ambiguities and the VCV matrix).
- The validity of integer ambiguities is tested using statistical tests.
- Application of integer ambiguities

The Ginan software can be set to solve ambiguities in GPS and Galileo measurements. Ambiguity resolution for Beidou and QZSS signal can be expected in the future. Due to it use of FDMA in its system, ambiguity resolution on GLONASS is known to be of a particular challenge. For this reason ambiguity resolution for GLONASS is not planned for Ginan until GLONASS's new CDMA signals are fully operational.

8.1 Real-valued ambiguity estimation

Reference or pivot chain is normally done by selecting a station with the most number of observations, however this is not possible to no a-priori for a systems that is designed to run in real-time.

1. Assign a station as anchor, the user is encouraged to select a reliable station.
2. Biases for the anchor station b_r are set to zero

3. Ambiguities for signal in anchor station z_r^s are set to minimise satellite biases
4. Set satellite biases as $b^s = A_r^s - z_r^s - b_r$
5. For each satellite with defined biases:
   (a) Find ambiguity measurements A_r^s for which the receiver b_r side bias is undefined
   (b) Ambiguities for signal z_r^s are set to minimise receiver biases
   (c) Set receiver biases as $b_r = A_r^s - z_r^s - b^s$
6. For each receiver with defined biases:
   (a) Find ambiguity measurements A_r^s for which the satellite side bias b^s is undefined
   (b) Ambiguities for signal z_r^s are set to minimise satellite biases
   (c) Set satellite biases as $b^s = A_r^s - z_r^s - b^s$
7. Repeat steps 5 and 6 until biases are defined for all receivers and satellites

8.2 Integer ambiguity estimation and validation

In the pea we have implemented a number of different ambiguity resolution strategies:

- Integer rounding
- Iterative rounding
- Lambda Integer Least Squares
- Best integer Equivariant (BIE)

8.2.1 Integer rounding

The simplest strategy to apply is to round the real-values estimates to the nearest integers, without using any variance co-variance information.

8.2.2 Iterative rounding

The bootstrapping algorithm takes the first ambiguity and rounds its value to the nearest integer. Having obtained the integer value of this first ambiguity, the real-valued estimates of all remaining ambiguities are then corrected by virtue of their correlation with the first ambiguity. Then the second, but now corrected, real-valued ambiguity estimate is rounded to its nearest integer, and the process is then repeated again with both ambiguities held fixed, and the process is continued until all ambiguities are accommodated. Thus the bootstrapped estimator reduces to 'integer rounding' in case correlations are absent.

8.2.3 Lambda Integer Least Squares

8.2.4 BIE

The previously mentioned algorithms are known as hard decision algorithms.

8.3 Application of integer ambiguities

The Melbourne-Wubbena linear combination (Melbourne 1985),(Wubbena 1985) is a linear combination of the L1 and L2 carrier phase plus the P1 and P2 pseudorange. The geometry, troposphere and ionosphere are eliminated by it. The Melbourne-Wubbena linear combination can be represented as:

$$E(L^S_{r,IF}) - \frac{cf_2 z^s_{r,w}}{f_1^2 - f_2^2} = \rho^s_r + c(dt_{r,IF} - dt^s_{IF}) + \tau^s_r + \lambda_n z^s_{r,1} + (\lambda_{IF}\delta_{r,IF} \quad (8.1)$$

Since „% comprises of both code and phase measurements, it is reasonable to exclude the lower elevation measurements to avoid the multipath impacts from the code observation. Normally, with 30 degree elevation cut-off, an averaging of 5 minutes of (4) is good enough to fixing the wide-lane ambiguities [RD 04]. The rests are the wide-lane phase bias, which can be broadcasted to the user for user side wide-lane ambiguity resolution. Either choosing a pivot receiver bias or a single-differencing between two satellites can avoid the linear dependency. highly correlated need lambda With the fixing of the wide-lane ambiguity, equation (1) and (2) can be further deducted as: The code bias and phase bias in equations **??** and (6) can be lumped into the corresponding receiver and satellite clock errors. Then equations (5) and (6) become: with: By such an reformulation, there are two types of satellite clock: 1) IGS type clock, but estimated only from code measurements; 2) phase clock, estimated using phase measurements and can be used to support PPP ambiguity resolution on the user side. The drawback of this approach is that there is no precise IGS compatible clock after the processing and it has to be derived from the existing PEA processing

- A modified phase clock/bias model to improve PPP ambiguity resolution at Wuhan University Journal of Geodesy - Geng et al. (2019)
- On the interoperability of IGS products for precise point positioning with ambiguity resolution Journal of Geodesy - Simon et al. (2020)
- Resolution of GPS carrier-phase ambiguities in precise point positioning (PPP) with daily observations Journal of Geodesy - Ge et al. (2008)
- Real time zero-difference ambiguities fixing and absolute RTK ION NTM - Laurichesse et al. (2008)
- Undifferenced GPS ambiguity resolution using the decoupled clock model and ambiguity datum fixing Navigation – Collins et al. (2010)
- Improving the estimation of fractional-cycle biases for ambiguity resolution in precise point positioning Journal of Geodesy – Geng (2012).
- Modeling and quality control for reliable precise point positioning integer ambiguity resolution with GNSS modernization GPS Solutions – Li et al. (2014).

9 Minimum Constraints

When running the toolkit in network mode, the positions of receivers and satellites may be estimated simultaneously. This presents a complication in that the absolute positions of all elements may be unconstrained - the model of the system would be completely consistent if every element of the network (receivers and satellites) were on the opposite side of the planet! For the results of such processing to be of value, the system needs to be referenced to a standard reference frame.
One method of ensuring the system is referenced to a suitable reference frame is to constrain receiver positions to their nominal position in the reference frame. Constraining 3 receivers is sufficient to ensure the system is well defined, but this gives select receivers precedence over all others - a movement of one of these receivers will instead show up as a movement of every other receiver on the planet.
An alternative method may be to weakly constrain all receivers to their nominal positions. This removes the priority effect of choosing 3 receivers, but will also apply a small restoring bias to the estimated position of the receivers.
Minimum constrains is a method of referring a system of estimates to a standard reference frame without unduly prioritising any receiver, or biasing the measurements.

9.1 Computation

To compute a minimally constrained system, the deviations of all receivers from their nominal positions are used as pseudo-observations in a filter to estimate a transformation between reference frames.
The filter estimates a rigid transformation comprising of 3d rotation and translation components, which when applied produces a least-squares solution of the errors of all desired receivers.
The estimated transformation is then applied to the network solution, transforming both the estimated position states, as well as the covariances associated with them.

9.2 Usage

```
minimum_constraints:

    estimate:
        translation:    true
        rotation:       true
        #scale:          false   #not yet implemented

    station_default_noise: -1        #constrain none by default (negative numbers are not constrained)
    #station_default_noise: +1       #constrain all by default

    #station_noise:
    #    ALIC: 0.001     #constrain strongly
    #    AGGO: 1
    #    BOAV: 100       #constrain weakly
```

Listing 9.1: Minimum Constraints Configuration

If minimum constraints is enabled, the transformation will be applied after the completion of all forward processing, and configured according to the yaml file.

estimate:

Booleans to enable the modelling of the transformation operations.

station_noise, station_default_noise:

These parameters allow the different stations in the solution to be weighted according to their priority.

The station default noise is applied to all stations unless they are specifically overridden using a station_noise entry.

High noise values indicate that the station should only be weakly constrained to its a-priori position, while low values indicate they should be weighted more strongly. Only the relative weighting between stations is important.

Stations may have zero weighting applied by assigning a negative value to the station_noise override for that station.

10 Flex Events

10.1 Introduction

Certain satellites in the GPS constellation have the ability to change the output power on the signals they transmit toward Earth. Effectively this is achieved by re-distributing the power allocated to each signal component as seen in **?**. This means that individual signal components can therefore transmit above previously stated maximum [**?**, sec. 6.3.1].
This "flexible power" or "flex power" capability is used as an anti-jamming technique. When flex power is activated (or de-activated) by the Control Segment of the GPS system, it is sometimes referred to as a "flex event". When a series of flex events are associated with a given set of GPS satellites, alter the power spectral distribution in similar ways and/or are targeted over the same geographical region, this is categorised as a flex power mode [**?**]. In **?**, three flex power modes are discussed. This includes a global activation for all healthy GPS Block IIR-M and IIF satellite for a period of 4 days (Mode II) to a geographically localised (regional) activation for 10 out of 12 Block IIF satellites on a continuous basis over a point centred in the Middle East (Mode I).
Since the launch of the first GPS block IIR-M satellite in 2005[1], every new GPS satellite has had programmable power output capabilities. This therefore includes the newer block satellites IIF and IIIA as well. Other constellations do not have this programmable variation, i.e. the power output is static.
GLONASS has greater variation in the total power output within a generation, i.e. different satellites in a given generation will vary by 10's of watts. The power output of L1 and L2 frequencies will also differ with gradation of low/medium/high. This is covered in [**?**], but a good summary can be found in **?**.
Galileo also has some variation in signal outputs across the In-Orbit Validation (IOV) satellites although the Full Operational Capability (FOC) satellites are designed to be constant [**?**]. According to the **?** the power range for IOV sats is 95 - 134 W, and FOC sats is 254 - 273 W.

10.2 Ginan Code Example

In order to detect flex events, we require both RINEX3 observation files for a given station and time period, and the associated sp3 orbital files which detail the positions of the satellites. The Ginan code base includes Python scripts to automatically download and process files to find these flex events.
For example, running the following command:

```
python3 find_flex_events.py HOB200AUS 2021-02-15 2021-02-18 S2W /home/user/
    test/test_data/ -c_dir /home/user/test/test_csv/ -p -p_dir /home/user/test
    /test_plts/ -p_span 2400
```

will download RINEX3 and sp3 files for the `HOB200AUS` station between the dates `2021-02-15` and `2021-02-18` into the directory `/home/user/test/test_data/`, find any flex events ("Start" and "End") for all GPS satellites, output a list of these events to a `csv` file in

[1] https://www.e-education.psu.edu/geog862/node/1773

`/home/user/test/testcsv/` and output the results as a series of `.png` plots (one for each event).
Apart from the station, dates and download directory provided in the command, all others are optional;

- `-c_dir` specifies the directory to save the `csv` file;
- `-p` specifies that plots are to be produced;
- `-p_dir` specifies the directory to save the `png` plot files;
- `-p_span` specifies the time span for the plots (in seconds).

11 Attribution

Ginan - Analysis Centre Software
A project funded as Part of the Positioning Australia program.
Geoscience Australia
https://www.ga.gov.au/scientific-topics/positioning-navigation/positioning-australia
clientservices@ga.gov.au
Cnr Jerrabomberra Ave and Hindmarsh Drive
Symonston ACT 2609
Australia