locuslock Documentation

Release 1.0.0

October 03, 2023

Chapter 1

Introduction

This documentation serves to help you navigate how to set-up and operate Locus Lock's *software-defined* GNSS receiver. Our GNSS receiver enables secured-centimeter resolution location data access for military and enterprise products, incorporating cutting-edge signal processing with robust estimation techniques, to provide advanced signal situational awareness and high-fidelity positioning. Our mission is to deliver centimeter-accurate real-time positioning to ensure a globally available, all-weather solution at a fraction of the cost.

If you have any questions regarding the hardware or software manuals, please email info@locuslock.com to learn more.

Chapter 2

Hardware User Guide

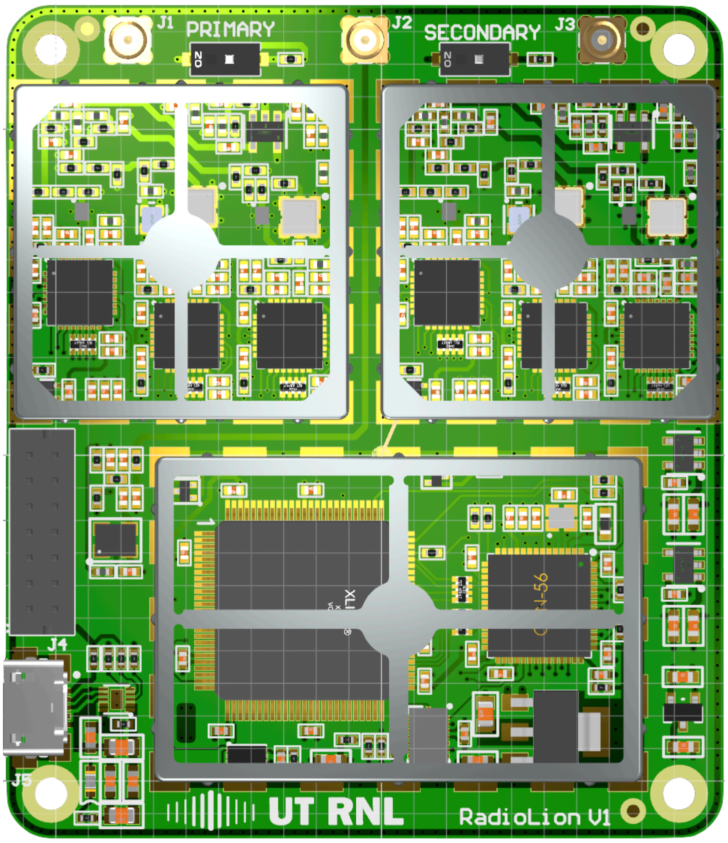
This section will describe how to setup the RadioLion hardware prior to using the GNSS software suite.

2.1 Antennas

The RadioLion board is a triple-frequency, dual-antenna front-end board, which means it simultaneously digitizes signals at three frequencies (GPS L1, L2, and L5) and from two antennas. Using two antennas is especially advantageous for rover receivers, as this (1) allows rover heading to be determined directly, (2) affords spatial diversity for more reliable precise solutions, and (3) offers a powerful defense against GNSS spoofing. A reference receiver can also benefit from dual-antenna input, but it's not so important as for a rover receiver. Here is a list of triple-frequency antennas suitable for use with the RadioLion front end (not inclusive of all options).

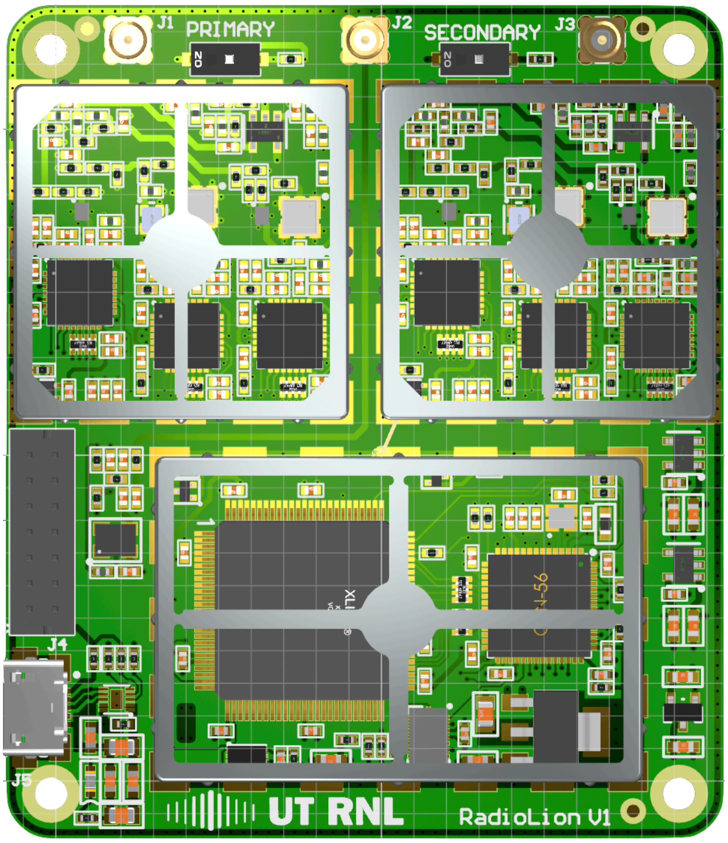
- Harxon GPS1000: These run about $210 each in low quantities.
- Tallysman TW3972: These cost about $370 each in low quantities.

The RF input connector on the RadioLion board is an MMCX jack. You'll need to ensure that the cable coming from the antenna terminates in an MMCX plug, or can be made to do so with the help of an adapter (e.g., an SMA-to-MMCX adapter). Ensure the primary antenna is connected to the MMCX jack (left) with the label "primary" and the secondary antenna is connected to the MMCX jack (right) with the label "secondary". See picture below:



2.2 RadioLion

The RadioLion dual-antenna, triple-frequency RF Front-End receives, filters, amplifies, down-mixes, and digitizes the raw analog GNSS signals from the antenna(s) to produce a binary stream of intermediate-frequency samples that the receiver computer can process. The board also houses a Bosch BMI088 inertial measurement sensor (IMU).



2.2.1 Key Features

- Integrated digital radio & inertial measurement front-end designed for use with software-defined GNSS and GNSS-inertial fusion systems
- Multi-constellation and multi-frequency heading and positioning solution
- Streams RF data samples and IMU measurements to processor
- Supports simultaneous capture from two antenna ''channels'' on three frequencies each via a total of six radio integrated circuits (ICs) and accompanying filter networks
- Real time operation
- USB connectivity with computer
- Generates a configurable 1-pulse-per-second (PPS) output synchronized to GPS time: a valuable feature for sensor synchronization, vehicle coordination, and system integration
- Small form factor for seamless product integration
- High accuracy clock oscillator (20 MHz)
- Preconfigured for external or internal clock source (switching requires minimal rework)
- Software-reconfigurable, high-rate data streaming at up to 196 Mbits per second
- Flexible sampling rates

2.3 Computer

GRID is a mature C++ GNSS software suite (TRL 7) that is portable (desktop & embedded) and modular, featuring SDR techniques, with easy configuration and extensibility, scintillation-robust tracking loops, real-time and replay capabilities, and data bit wipe-off. Software can be deployed on multiple form factors. It can be tailored to static, desktop testing using a high performance Intel NUC or can be tailored to a smaller embedded form factor, such as an Odroid C4. Here is a list of recommended processors suitable for GNSS-SDR (not inclusive of all options).

- Odroid C4: These run about $55-85 each in low quantities.
- Intel NUC: These cost about ~$500-900 each in low quantities, depending on the RAM and storage needed.

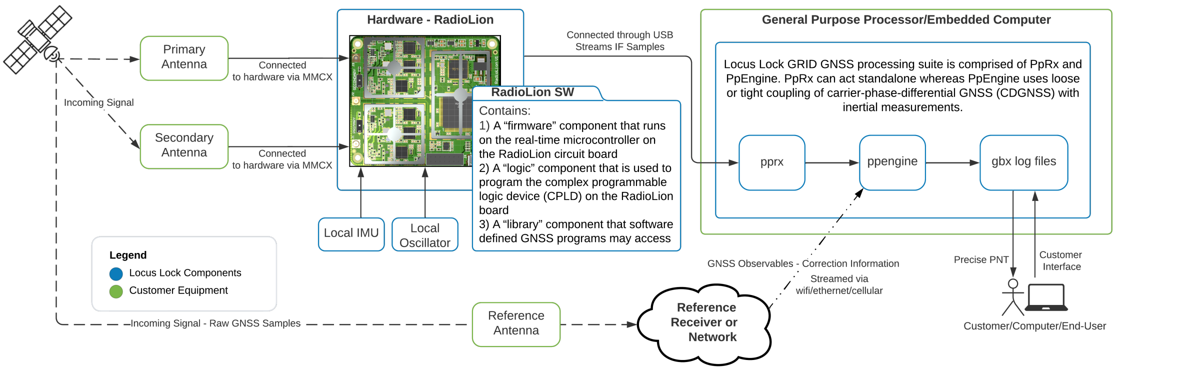
Chapter 3

Software User Guide

This section will describe how to use the GRID Software Suite.

3.1 Software Overview

See the high-level diagram below for how the RadioLion RF Front-End interacts with GRID.



3.2 Precise Positioning Receiver (pprx)

This page describes how to configure, run, and interpret output from pprx, which is an implementation of GRID, the General Radionavigation Interfusion Device.

3.2.1 Editing Options Files

pprx is highly configurable via both command-line options and configuration parameters. Type `pprx --help` to see a list of command-line options. These options are specified in a `.opt` file as seen in the *example* below:

```
-i /dev/radiolynx0
--imu-file /dev/radiolynx1
-e
--realtime
--ref-host=refnet1.ae.utexas.edu
-c ./pprx.config
--bitpack lynx
--depth 8
-d
-v
```

```
-t -1
--log-interval 10
--ref-interval 20
--acq-interval 1000
--binary-only
# For output to a gbx file
# -o ./pprx.gbx
# For output to a named pipe
-o pprx_write
-T 4
```

3.2.2 Editing Configuration Parameters

pprx configuration files are broken into configuration blocks. The start of each block is indicated by a block header, e.g., ESTIMATOR. A pprx `.opt` file (i.e. pprx.opt) has a corresponding configuration file, `.config` (i.e. pprx.config). See this `example configuration file` for pprx.

Each block in the `pprx.config` file contains various configuration parameters. See each block below to view all of its configuration parameter choices and if applicable the choices for each parameter:

BASETIME

Below is an *example* of the BASETIME block:

```
[BASETIME]
GPS_WEEK_REFERENCE = 2210
```

The BASETIME block contains all of the following configuration parameters:
GPS_WEEK_REFERENCE

Default: N/A

Definition: GPS Week #. See here

FRONT-END

Below is an *example* of the FRONT-END block:

```
[FRONT_END]
NUM_FRONT_ENDS = 1
FE01 = LYNX

[LYNX]
# Lynx clock frequency is slightly slower than nominal 19.2 MHz; see note E1 in
ut16.config.
SAMPLE_FREQ_NUMERATOR = 19199984
SAMPLE_FREQ_DENOMINATOR = 2
QUANTIZATION = 2
NUM_SUPPORTED_SIGNAL_TYPES = 8
SUPPORTED_SIGNAL_TYPES = GPS_L1_CA_PRIMARY GPS_L1_CA_ALT1 GPS_L2_CLM_PRIMARY
GPS_L2_CLM_ALT1 SBAS_L1_I_PRIMARY SBAS_L1_I_ALT1 GALILEO_E1_BC_PRIMARY
GALILEO_E1_BC_ALT1
# Frequencies corrected for slower Lynx clock; see note E1 in ut16.config for
details.
FREQ_IF_HZ = 2.392739428572e6 2.392739428572e6 2.412449561905206e+06
2.412449561905206e6 2.392739428572e6 2.392739428572e6 2.392739428572e6
2.392739428572e6
PLL_SIGN_FPLL = 1 1 1 1 1 1 1 1
CODE_PHASE_BIAS_METERS = 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

The FRONT-END block contains all of the following configuration parameters:

Note There should be three banks labeled `[FRONT_END]`, `[LION]`, and `[LION_L5]`. The configuration options for `[FRONT_END]` are below.

NUM_FRONT_ENDS

Default: N/A. Required input. For RadioLion this should be set to 2.

Definition: The number of specified front-ends.
FE01

Default: N/A. Required input. For RadioLion this should be set to LION.

Definition: Which bank configurations apply to Front-End 01.
FE02

Default: N/A. Required input. For RadioLion this should be set to LION_L5.

Definition: Which bank configurations apply to Front-End 02.

Note The configuration options for `[LION]` and `[LION_L5]` are below.

SAMPLE_FREQ_NUMERATOR

Default: N/A. Required input.

Definition: It is assumed that the sampling frequency of any RF front end supported by this code can be expressed in Hz as a ratio of two integers. For example, the sampling frequency for the Zarlink/Plessey front end is 40e6/7 Hz, which would correspond to SAMPLE_FREQ_NUMERATOR = 40e6 and SAMPLE_FREQ_DENOMINATOR = 7. It is further assumed that double precision is sufficient to represent the time within a single interval of SAMPLE_FREQ_DENOMINATOR seconds.
SAMPLE_FREQ_DENOMINATOR

Default: N/A. Required input.

Definition: It is assumed that the sampling frequency of any RF front end supported by this code can be expressed in Hz as a ratio of two integers. For example, the sampling frequency for the Zarlink/Plessey front end is 40e6/7 Hz, which would correspond to SAMPLE_FREQ_NUMERATOR = 40e6 and SAMPLE_FREQ_DENOMINATOR = 7. It is further assumed that double precision is sufficient to represent the time within a single interval of SAMPLE_FREQ_DENOMINATOR seconds.
QUANTIZATION

Default: N/A. Required input.

Definition: The number of bits used in the front end's quantization scheme.
NUM_SUPPORTED_SIGNAL_TYPES

Default: N/A. Required input.

Definition: The number of signals desired to track.
SUPPORTED_SIGNAL_TYPES

Default: N/A. Required input.

Definition: An array listing the supported signal types. In the arrays that follow, data at the iith index correspond to the signal type at the iith index.
FREQ_IF_HZ

Default: N/A. Required input.

Definition: Intermediate frequency of the signal, in Hz. See `this python script` for how to calculate these numbers.

```
def calc_fIF():
    """
    This function calculates the intermediate frequencies necessary for
the options file.

    Please paste in the output of the syslog/dmesg log when plugging in
the RadioLion. You can paste the
    output directly into the appropriate configuration file for pprx.
```

```
    """

    iS1 = input("Enter first dmesg log: ")
    iS2 = input("Enter second dmesg log: ")
    iS3 = input("Enter third dmesg log: ")
    iS4 = input("Enter fourth dmesg log: ")
    iS5 = input("Enter fifth dmesg log: ")
       fCOMP = float(fTCXO)/float(RDIV)
       fLO_actual = fCOMP * (float(NDIV) + float(FDIV) / 1048576)
       fIF_actual = float(fC) - fLO_actual
       fIF_actual_list.append(fIF_actual)

    print("------------ L1 and L1 alt ------------")
    print("FREQ_IF_HZ = %.6f %.6f" %(fIF_actual_list[0], fIF_actual_list[1]))
    print("------------ L2 and L2 alt ------------")
    print("FREQ_IF_HZ = %.6f %.6f" %(fIF_actual_list[2], fIF_actual_list[3]))
    print("------------ L5 and L5 alt ------------")
    print("FREQ_IF_HZ = %.6f %.6f" %(fIF_actual_list[4], fIF_actual_list[5]))


calc_fIF()
```

PLL_SIGN_FPLL

Default: N/A. Required input.

Definition: Indicates high (-1) or low (1) side mixing

CODE_PHASE_BIAS_METERS

Default: N/A. Required input.

Definition: The bias below is the amount by which a biased pseudorange exceeds its bias-free representation. Typically, the bias for SignalType GPS_L1_CA is set to zero and all other SignalTypes are referenced to this.

BUFFER_LOADER

Below is an *example* of the BUFFER_LOADER block:

```
[BUFFER_LOADER]
DEVICE = LYNX
# Set TYPE = FILE to post-process capture files.
TYPE = USB
```

The BUFFER_LOADER block contains all of the following configuration parameters:

NUM_BUFFER_LOADERS

Default: N/A. Required input. For RadioLion this should be set to 1.

Definition: How many buffer loaders you have. One RadioLion = 1 Buffer Loader.

BL01

Default: N/A. Required input.

Definition: `BL_LION` corresponds to buffer loader 1, with device LION.

DEVICE

Default: N/A. Required input.

Definition: `LION` corresponds to the device name.

TYPE

Default: N/A. Required input.

Definition: Operational modes. Select from the following options:

- `FILE`: Post-process capture files

- `USB`: Support live capture operations

FRONT_ENDS

Default: N/A. Required input.

Definition: `LION LION_L5` corresponds to the RadioLion.

BANK

Below is an *example* of the BANK block:

```
[BANK]
NUM_BANKS = 8
BK01 = GPS_L1_CA_PRIMARY
BK02 = SBAS_L1_I_PRIMARY
BK03 = GALILEO_E1_BC_PRIMARY
BK04 = GPS_L2_CLM_PRIMARY
BK05 = GPS_L1_CA_ALT1
BK06 = SBAS_L1_I_ALT1
BK07 = GALILEO_E1_BC_ALT1
BK08 = GPS_L2_CLM_ALT1

[GPS_L1_CA_PRIMARY]
FRONT_END = LYNX
MAXCHANNELS = 13
NOM_MIN_DOPPLER_FREQ_HZ = -5000
NOM_MAX_DOPPLER_FREQ_HZ =  5000
NUM_SUBACCUM_PER_ACCUM = 10
MAX_ACQ_SEARCH_DEPTH = 2
CODEGEN_TYPE = LOOKUP
CH_PRUNE_INTERVAL_SUBACCUMS = 1000
CH_PRUNE_THRESHOLD = 10
TRACKING_STRATEGY = HYBRID
PLL_DEFAULT_BANDWIDTH_HZ = 25
PLL_HYBRID_BANDWIDTH_HZ = 25
PLL_ENABLE_LOOP_BANDWIDTH_ADAPTATION = TRUE
PLL_DEFAULT_LOOP_ORDER = ORDER2
PLL_HYBRID_LOOP_ORDER = ORDER2
PLL_PHASE_FLAG_THRESHOLD = 0.4
EML_CHIP_SPACING = 0.2
DLL_DEFAULT_BANDWIDTH_HZ = 0.5
CH_IQSQ_FILTER_TAU_SEC = 0.5
ALLOW_DATA_SYMBOL_PREDICTION = TRUE
DATA_SYMBOL_PULL_SNR_THRESHOLD = 23
DATA_SYMBOL_PUSH_SNR_THRESHOLD = 100
DIRECTED_ACQ_ONLY = TRUE
ELEVATION_MASK_ANGLE_ACQ_DEG = 10.0
NOISE_FLOOR_CORRECTION_FACTOR = 1.0
CIRCBUFF_STREAM_IDX = 0
```

The BANK block contains all of the following configuration parameters:

Warning There will be one bank called `[BANK]`. Each signal type you specify you want to track in that bank corresponds to an additional bank you must make specifically for that signal. For example, if in the `[BANK]` block, you specify you want to track 2 signals, GPS_L1_CA_PRIMARY and GPS_L1_CA_ALT1, then you will have three total banks: `[BANK]`, `[GPS_L1_CA_PRIMARY]`, and `[GPS_L1_CA_ALT1]`. I.e. there must be banks must be specified for all reported signal types in the BK01, BK02, … parameters.

Note The configuration options for `[BANK]` are below.

NUM_BANKS

Default: N/A. Required input.

Definition: Specifies the number of signals to track.

BK01, BK02, …

Default: N/A. Required input.

Definition: Specifies which signal type to track in each bank.

Note The configuration options for each signal type are below, for example `[GPS_L1_CA_PRIMARY]`.

NOM_MIN_DOPPLER_FREQ_HZ

Default: -7000

Definition: Nominal minimum Doppler frequency used in the carrier replica array

NOM_MAX_DOPPLER_FREQ_HZ

Default: 7000

Definition: Nominal maximum Doppler frequency used in the carrier replica array

DOPPLER_FREQ_STEP_HZ

Default: 1000

Definition: Doppler frequency step used in the carrier replica array

NOM_CHIPRATE_CPS

Default: Dependent on the Signal Type selected. Nominal rates for each Signal Type are stored and pulled at runtime if you do not specify anything in the configuration file.

Definition: Nominal chipping rate of PRN code, in chips per second

FREQ_CARRIER_HZ

Default: Dependent on the Signal Type selected. Nominal rates for each Signal Type are stored and pulled at runtime if you do not specify anything in the configuration file.

Definition: Nominal carrier frequency, in Hz

EML_CHIP_SPACING

Default: 0.3

Definition: Spacing between early and late correlators, in chips

NUM_CHIPS_PER_SUBACCUM

Default: Dependent on the Signal Type selected. Nominal rates for each Signal Type are stored and pulled at runtime if you do not specify anything in the configuration file.

Definition: Number of chips per subaccumulation. NUM_CHIPS_PER_SUBACCUM must be an integer divisor of the number of chips in a PRN code cycle (NUM_CHIPS_PER_CODE).

NUM_CHIPS_PER_CODE

Default: Dependent on the Signal Type selected. Nominal rates for each Signal Type are stored and pulled at runtime if you do not specify anything in the configuration file.

Definition: Number of chips in PRN code

NUM_SUBACCUM_PER_SYMBOL

Default: Dependent on the Signal Type selected. Nominal rates for each Signal Type are stored and pulled at runtime if you do not specify anything in the configuration file.

Definition: Number of subaccumulations per symbol

MAXCHANNELS

Default: 8

Definition: Maximum number of channels that the Bank object is allowed to have.

NUM_SUBACCUM_PER_ACCUM

Default: 10

Definition: Integer number of subaccumulation intervals per accumulation interval. If the signal to be tracked is data modulated, then NUM_SUBACCUM_PER_ACCUM must be an integer divisor of the number of subaccumulations per data symbol (NUM_SUBACCUM_PER_SYMBOL).

CH_IQSQ_FILTER_TAU_SEC

Default: 0.5

Definition: Time constant for the low-pass filter that smoothes the one-C/A-code-period I^2 + Q^2 quantities to get E[I^2 + Q^2]

CH_DISTORTION_FILTER_TAU_SEC

Default: 0.5

Definition: As with the I^2 + Q^2 filter, the symmetric difference distortion statistic filter is a digitized low-pass filter.

CH_PRUNE_INTERVAL_SEC

Default: 1.0

Definition: Nominal pruning interval, in seconds

CH_PRUNE_THRESHOLD

Default: 10

Definition: Number of consecutive times a signal must fail the pruning criteria, which is called every CH_PRUNE_INTERVAL_SEC, before it gets pruned.

CH_PRUNE_CN0_THRESHOLD_MIN

Default: 23.0

Definition: Minimum C/N0 thresholds for pruning signals, in dB-Hz.

CH_PRUNE_CN0_THRESHOLD_MAX

Default: 65.0

Definition: Maximum C/N0 thresholds for pruning signals, in dB-Hz.

MAX_ACQ_SEARCH_DEPTH

Default: 5

Definition: Maximum acquisition search depth

MAX_DIRECTED_STANDARD_ACQ_ATTEMPTS_PER_CYCLE

Default: 2

Definition: Maximum number of directed standard acquisition attempts allowed per acquisition cycle.

FLOORED_RADIX2_FFT_ACQ_SIZE

Default: False

Definition: With radix-2-only FFT libraries, samples are interpolated to next radix-2 size. If FLOORED_RADIX2_FFT_ACQ_SIZE is set, the interpolation size is the previous radix-2 size, improving computational performance but also reducing bandwidth and causing aliasing.

DIRECT_TO_TRACK_ACQ_INITIAL_CN0_DB_HZ

Default: 40

Definition: C/N0 estimate used as an initial guess when initializing a channel via direct-to-track acquisition.

DIRECTED_ACQ_ONLY

Default: False

Definition: Allow only directed acquisition (direct-to-track acquisition or directed standard acquisition) after initial acquisition. Note that this flag gets overridden when too few root bank signals have been acquired. The override feature ensures that processing power gets devoted to standard acquisition upon startup or after a complete loss of signals when directed acquisition isn't yet operable.

DIRECT_TO_TRACK_ACQ_ONLY

Default: False

Definition: Allow only direct-to-track acquisition after initial acquisition. Note that this is more restrictive than DIRECTED_ACQ_ONLY, which allows direct-to-track acquisition or directed standard acquisition. A runtime error will be thrown if this option is asserted simultaneously with DIRECTED_STANDARD_ACQ_ONLY; they are mutually exclusive.

DIRECTED_STANDARD_ACQ_ONLY

Default: False

Definition: Allow only directed standard acquisition after initial acquisition. Note that this is more restrictive than DIRECTED_ACQ_ONLY, which allows direct-to-track acquisition or directed standard acquisition. A runtime error will be thrown if this option is asserted simultaneously with DIRECT_TO_TRACK_ACQ_ONLY; they are mutually exclusive.

BACKGROUND_ACQ_ONLY

Default: False

Definition: Allow only background acquisition; prevent initial acquisition

DISABLE_STANDARD_ACQ

Default: False

Definition: Disable standard acquisition during background acquisition. This directive will not be overriden under any circumstance. Note that standard acquisition may be performed during initial (exhaustive) acquisition if BACKGROUND_ACQ_ONLY is false.

FORCE_HEALTHY_WHEN_TRACQUIRED

Default: False

Definition: Denote signal as healthy when tracquired so there is no need to wait for the signal's embedded health indicator to arrive (e.g., in subframe 1 for GPS L1 C/A) before the signal can participate in a navigation solution. (Note that this option typically only applies when ephemeris are imported, since tracquisition is not attempted for signals marked unhealthy. Imported ephemeris do not indicate health status.) The temporary healthy indication forced by this option will be overridden by the signal-borne health indicator as soon as it arrives.

FORCE_HEALTHY

Default: False

Definition: Signals marked unhealthy may still be useful. When this flag is asserted, all signals in the corresponding bank will be marked healthy.

ELEVATION_MASK_ANGLE_ACQ_DEG

Default: 0

Definition: Elevation mask angle for acquisition, in radians. Signals arriving at the receiver from transmitters below the elevation mask angle will be excluded from direct-to-track acquisition. Set to -PI/2 to prevent elevation masking. In the receiver config file, the elevation mask angle is given in degrees as ELEVATION_MASK_ANGLE_ACQ_DEG.

BORESIGHT_ELEVATION_MASK_ANGLE_ACQ_RAD

Default: 0

Definition: Boresight-relative elevation mask angle for acquisition, in radians. See ELEVATION_-MASK_ANGLE_ACQ_RAD.

PLL_DEFAULT_BANDWIDTH_HZ

Default: 25.0

Definition: Default PLL bandwidth, in Hz.

PLL_DEFAULT_LOOP_ORDER

Default: ORDER2

Definition: Possible closed-loop loop orders for phase tracking loops. Select from the following options:

- `ORDER1`
- `ORDER2`
- `ORDER3`

PLL_HYBRID_BANDWIDTH_HZ

Default: 25.0

Definition: Hybrid PLL bandwidth, in Hz.
PLL_HYBRID_LOOP_ORDER

Default: If not specified, defaults to whatever is set for PLL_DEFAULT_LOOP_ORDER. **Definition:** Possible closed-loop loop orders for phase tracking loops. Select from the following options:

- `ORDER1`
- `ORDER2`
- `ORDER3`

PLL_DEFAULT_DISCRIMINATOR_TYPE

Default: AT4_DISC

Definition: Types of phase tracking loop discriminators. Select from the following options:

- `AT_DISC`: Two-quadrant arctangent: atan(Q/I)
- `AT4_DISC`: Four-quadrant arctangent: atan2(Q,I)

TRACKING_STRATEGY

Default: TRADITIONAL

Definition: Signal tracking strategy. Select from the following options:

- `TRADITIONAL`: Traditional fll/pll/dll tracking loops
- `HYBRID`: Vector-aided fll/pll/dll tracking loops
- `VECTOR`: Fully vectorized tracking with batch superaccumulation fitting
- `DEEP`: Fully vectorized tracking with batch superaccumulation fitting and IMU aiding at the lowest level

NOISE_FLOOR_CORRECTION_FACTOR

Default: 1.0

Definition: Set greater than unity to correct for thermal noise floor underestimation that occurs when the incoming data samples are time correlated. Values less than unity are considered invalid.
PLL_PHASE_LOCK_THRESHOLD

Default: 0.9

Definition: The PLL's phase lock threshold is compared against the phase lock statistic in Equation 118 on page 393 of the Blue Book, volume 1.
PLL_PHASE_FLAG_THRESHOLD

Default: 0.4

Definition: Determines when a phase lock flag is raised to indicate possible cycle slippage

PLL_NUM_SUB_PER_PHASELOCK

Default: 20

Definition: Number of subaccumulations per phase lock detection interval.

PLL_ENABLE_LOOP_BANDWIDTH_ADAPTATION

Default: False

Definition: Indicates whether dynamic loop bandwidth adaptation (based in signal power) is enabled for the PLL.

PLL_FREQ_UPDATE_ON_SILENCE

Default: False

Definition: Set to true to allow the PLL to update the frequency estimate during intervals of known transmitter silence (applicable to TDMA signals).

DLL_DEFAULT_BANDWIDTH_HZ

Default: 0.03

Definition: Default DLL bandwidth, in Hz.

DLL_CARRIER_AIDING

Default: True

Definition: Specifies whether the DLL is aided by the carrier tracking loop.

ALLOW_DATA_SYMBOL_PREDICTION

Default: False

Definition: If true, data symbol estimates for purposes of data symbol wipeoff and data parsing and interpretation may be based on predicted values from DatabitManager. If false, data symbol estimates are based only on immediately measured symbol values.

DATA_SYMBOL_PULL_SNR_THRESHOLD

Default: 20

Definition: Thresholds governing behavior of data symbols pushed and pulled from DataBitManager. Thresholds are expressed as a signal-to-noise ratio in dB. SNR is defined for the symbol interval so that SNR = Tsym*C/N0, where Tsym is the symbol interval. For example, if Tsym = 0.02, then SNR = 20 dB corresponds to C/N0 = 37 dB-Hz.

DATA_SYMBOL_PUSH_SNR_THRESHOLD

Default: 22

Definition: Thresholds governing behavior of data symbols pushed and pulled from DataBitManager. Thresholds are expressed as a signal-to-noise ratio in dB. SNR is defined for the symbol interval so that SNR = Tsym*C/N0, where Tsym is the symbol interval. For example, if Tsym = 0.02, then SNR = 20 dB corresponds to C/N0 = 37 dB-Hz.

EXPORT_DATA_BIT_BLOCKS

Default: False

Definition: When asserted, each block of received and error-checked data bits will be exported to the internal GBX stream. The definition of a block differs by SignalType:GenericType. For GPS_L1_CA, it is an LNAV frame. For SBAS_L1_I, it is a 1-second block.

CIRCBUFF_STREAM_IDX

Default: 0

Definition: Index of circular buffer stream this bank should use.

FLL_NOM_BANDWIDTH_HZ

Default: 5.0

Definition: Nominal FLL bandwidth, in Hz.

FLL_WEAK_BANDWIDTH_HZ

Default: 1.0

Definition: Weak-signal FLL bandwidth, in Hz.

FLL_DEFAULT_LOOP_ORDER

Default: ORDER1

Definition: Possible closed-loop loop orders. Select from the following options:

- `ORDER1`
- `ORDER2`
- `ORDER3`

FLL_NBS1_NOM

Default: 12

Definition: Upper thresholds for the histogram-based data bit synchronization process, nominal-strength signals. See details in Blue Book volume 1 p. 395.

FLL_NBS1_WEAK

Default: 80

Definition: Upper thresholds for the histogram-based data bit synchronization process, weak signals. See details in Blue Book volume 1 p. 395.

FLL_NBS2_NOM

Default: 7

Definition: Lower thresholds for the histogram-based data bit synchronization process, nominal-strength signals. See details in Blue Book volume 1 p. 395.

FLL_NBS2_WEAK

Default: 70

Definition: Lower thresholds for the histogram-based data bit synchronization process, weak signals. See details in Blue Book volume 1 p. 395.

FLL_FREQ_UPDATE_ON_SILENCE

Default: False

Definition: Set to true to allow the FLL to update the frequency estimate during intervals of known transmitter silence (applicable to TDMA signals).

TXID_LIST

Default: []

Definition: List of TxIds valid for bank. If this list is empty, then all TxIds for the bank's system are assumed valid.

CODEGEN_TYPE

Default: PSIAKI

Definition: Type of oversampled code generators. Select from the following options:

- `NONE`
- `LOOKUP`
- `PSIAKI`
- `FULL_PRECISION`
- `MULTI_TAP`

MSAMPFRAC

Default: 14

Definition: Parameter governing the adjustment resolution of oversampled code replicas: an oversampled code will have MSAMPFRAC levels of adjustment per sample. Thus, the location of the first sample within a code can be specified to within the sampling interval divided by MSAMPFRAC of the desired location.

NUM_TAPS, PROMPT_TAP, EARLY_TAP, LATE_TAP

Default: 0

Definition: Parameters for the multi-tap generator

ESTIMATOR

Below is an *example* of the ESTIMATOR block:

```
[ESTIMATOR]
USE_IONO_CORR  = TRUE
USE_TROPO_CORR = TRUE
ELEVATION_MASK_ANGLE_DEG = 15
ENFORCE_STRICT_SELECTION = TRUE
ALLOW_ALT_CHANNEL_DATA = FALSE
DEWEIGHT_NON_GPS_L1_CA_PRIMARY = FALSE
ALLOW_DELTR_FIXUPS = TRUE
MAX_ABS_DELTR_SEC = 0.0001
DYNAMICS_MODEL = NEARLY_CONSTANT_VELOCITY
SQRT_Q_TILDE = 0.3
CLOCK_TYPE = OCXO_LOW_QUALITY
ZENITH_PSEUDORANGE_STD = 0.8
ZENITH_DOPPLER_STD = 0.8
ELEVATION_DEPENDENT_WEIGHTING = TRUE
INNOVATIONS_TEST_THRESHOLD_FACTOR = 2
#ZERO_VELOCITY_UPDATE_THRESHOLD_MPS = 0.15
#CONSTRAIN_ECEF_POSITION = TRUE
#KNOWN_ECEF_POSITION = -741204.663  -5462368.681 3197945.603
```

The ESTIMATOR block contains all of the following configuration parameters:

PLATFORM_ID

Default: 1

Definition: Identifier of the receiver platform.

ENFORCE_STRICT_SELECTION

Default: True

Definition: Allow only phase-error-free signals with a steady-state DLL to participate in the estimation solution.

USE_IONO_CORR

Default: True

Definition: Set true to correct for ionospheric delay when estimating the state

USE_TROPO_CORR

Default: True

Definition: Set true to correct for tropospheric (neutral atmospheric) delay when estimating the state

ALLOW_DELTR_FIXUPS

Default: True

Definition: Indicates whether to allow fixups to set Observables:tOffset so that receiver time remains close to true time.

ELEVATION_MASK_ANGLE_RAD

Default: 0.25

Definition: Elevation mask angle, in radians. Signals arriving at the receiver from transmitters below the elevation mask angle will be excluded from the estimation solution. Set to -PI/2 to prevent elevationmasking. In the receiver config file, the elevation mask angle is given in degrees as ELEVATION_-MASK_ANGLE_DEG.

BORESIGHT_ELEVATION_MASK_ANGLE_RAD

Default: 0.25

Definition: Boresight-relative elevation mask angle, in radians. See ELEVATION_MASK_ANGLE_RAD.

MAX_ABS_DELTR_SEC

Default: 0.032

Definition: Maximum permitted absolute value of receiver clock bias relative to true GPS time, expressed in seconds. If, due to clock drift or initialization error, the absolute value of receiver clock bias exceeds this value, a receiver clock fixup will be triggered. As part of this clock fixup, concomitant changes to pseudorange and carrier phase observables will be made to keep the observables consistent, as required by the RINEX standard (See RINEX discussion under "DEFINITION OF THE OBSERVABLES"). Larger values of MAX_ABS_DELTR_METERS will lead to less frequent clock fixups. Due to constraints in how the fixup is performed and to avoid overflow in the carrier phase container, MAX_ABS_DELTR_SEC must satisfy the following inequality: DELTR_FIXUP_RESOLUTION_SEC() <= MAX_ABS_DELTR_SEC <= DELTR_FIXUP_MAX_SEC()

INNOVATIONS_TEST_THRESHOLD_FACTOR

Default: 2

Definition: Innovations testing within Estimator is based on the inner product of the normalized innovations vector, dzr = dot(zr, zr), which under a consistent estimator is chi-square distributed with zr.n_elem degrees of freedom. The threshold for excluding a set of measurements is taken as the factor below multipled by zr.n_elem.

CLOCK_TYPE

Default: TCXO_LOW_QUALITY

Definition: Assumed receiver clock model. Select from the following options:

- `TCXO_LOW_QUALITY`
- `TCXO`
- `OCXO_LOW_QUALITY`
- `OCXO`

DYNAMICS_MODEL

Default: NEARLY_CONSTANT_VELOCITY

Definition: Assumed antenna motion model. Select from the following options:

- `STATIC`
- `NEARLY_CONSTANT_VELOCITY`
- `NEARLY_CONSTANT_ACCELERATION`
- `LOW_EARTH_ORBIT`
- `IMU_BASED`

SQRT_Q_TILDE

Default: 5

Definition: The process noise for all dynamics models is expressed in terms of SQRT_Q_TILDE, the square root of the noise intensity. See Bar Shalom "Estimation with Applications to Tracking and Navigation" sections 6.2.1 to 6.2.3 for details. The units of ``SQRT_Q_TILDE``are as follows for each dynamics model:

- STATIC meters/sqrt(sec)
- NEARLY_CONSTANT_VELOCITY meters/sqrt(sec^3)

- NEARLY_CONSTANT_ACCELERATION meters/sqrt(sec^5)

SQRT_Q_TILDE represents the standard deviation of error induced on position, velocity, or acceleration state elements by the process noise over a 1-second propagation step. The standard deviation corresponding to a T-second step is then approximated as sigmaX = sqrt(T)*SQRT_Q_TILDE (see, e.g., Eq. 6.22-13 in Bar Shalom). This approximation is valid for short T; for long T, one needs to take multiple short propagation steps.

INIT_VELOCITY_STD

Default: 1000.0

Definition: At initialization, velocity and acceleration will be constrained because they are not immediately observable. Those constraints will assume a zero-mean Gaussian distribution with the provided standard-deviation for each scalar element. Standard deviation of initialization velocity, in meters/sec.

INIT_ACCELERATION_STD

Default: 1000.0

Definition: Standard deviation of initialization acceleration, in meters/sec^2.

INTERCHANNEL_BIAS_STD

Default: 0.0

Definition: Applicable only to the multi-antenna Estimator, INTERCHANNEL_BIAS_STD represents the standard deviation of error induced on the interchannel biases of the ALT channels relative to the primary channel by the process noise over a 1-second propagation step. In the multi-antenna Estimator, each of the primary and ALT channels have their own receiver-clock offsets that are mutually correlated through a common receiver-clock offset rate. When set to 0, a single receiver-clock offset (and rate) will be used for all channels. The units of INTERCHANNEL_BIAS_STD are seconds/sqrt(sec).

ZENITH_PSEUDORANGE_STD

Default: 3

Definition: Standard deviation of pseudorange measurements assuming a transmitter at zenith, in meters.

ZENITH_DOPPLER_STD

Default: 1

Definition: Standard deviation of Doppler measurements assuming a transmitter at zenith, in Hz.

ELEVATION_DEPENDENT_WEIGHTING

Default: True

Definition: Assert to weight observables by 1/sin(el), where el is the elevation angle. This has the effect of de-weighting multipath-corrupted low-elevation signals. If not asserted, all observables are weighted equally.

ASSUME_COINCIDENT_GROUPS

Default: False

Definition: Set true to treat all antennas specified in GROUPS as coincident and estimate only a single, common position (and velocity, acceleration, etc.). Set false to estimate independent positions for each antenna.

DEWEIGHT_NON_GPS_L1_CA_PRIMARY

Default: False

Definition: Assert to de-weight all non-GPS_L1_CA_PRIMARY SignalTypes in the navigation solution. The de-weighted signals types may still be incorporated into the solution, but they are de-weighted to the extent that their effect is negligible. This de-weighting is useful for (1) resorting to a trusted GPS_L1_CA_PRIMARY-only navigation solution, and (2) examining the innovations of non-GPS_L1_CA_PRIMARY signals against a trusted GPS_L1_CA_PRIMARY-based solution.

GROUPS

Default: All Signal Types supported by Primary Antenna

Definition: Specify the antenna groups involved in estimation.
DEMAND_ALL

Default: True

Definition: Set true to register demand for all SignalTypes and TxIds. This is the default value and provides the legacy behavior of PpRx. Set false to register demand only for those Groups involved in estimation.
CONSTRAIN_ECEF_POSITION

Default: False

Definition: Assert to constrain ECEF position to KNOWN_ECEF_POSITION. Only applicable when dynamics model is STATIC.
KNOWN_ECEF_POSITION

Default: [0 0 0]

Definition: Known ECEF position of L1 antenna phase center, in meters.
ZERO_VELOCITY_UPDATE_THRESHOLD_MPS

Default: 0.0

Definition: If the norm of the estimated velocity vector is below this value, expressed in meters per second, then the estimator will assume that thevelocity is actually zero and update the state accordingly.
DELTR_FIXUP_RESOLUTION_SEC

Default: 0.0001

Definition: This quantity is the time fixup resolution, in seconds. A time fixup DELTR is always chosen to be an integer multiple of this quantity. This implies that DELTR_FIXUP_RESOLUTION_SEC must be chosen such that for any carrier frequency fc considered within the code, fc*DELTR_FIXUP_RESOLUTION_SEC is an integer number of cycles.

DISPLAY

Below is an *example* of the DISPLAY block:

```
[DISPLAY]
DISPLAY_EXTRA_FIELDS = TRUE
```

The DISPLAY block contains all of the following configuration parameters:
USE_COLOR

Default: True

Definition: Color the display to stdout
MESSAGE_PRINTING_PERSISTENCE

Default: 5

Definition: Number of consecutive displays to print a given status message
DISPLAY_EXTRA_FIELDS

Default: False

Definition: Print extra information like phase lock statistics
MAXIMUM_BANK_COLUMNS

Default: 2

Definition: Maximum number of columns for Bank displays

WRAP_BANK_ROWS

Default: False

Definition: When true, Banks that would exceed MAXIMUM_BANK_COLUMNS are wrapped to the next row. When false, those Banks are not displayed.

MIX_BANK_ROWS

Default: False

Definition: MIX_BANK_ROWS determines whether different GenericTypes can be displayed in the same row. This setting is only meaningful when WRAP_COLUMNS is true.

REDRAW_PERIOD

Default: 1

Definition: Display data are printed every REDRAW_PERIOD log intervals

3.2.3 How to Run

Examining `--help` **Documentation** Running the following command

```
pprx --help
```

will show all the command-line options available for pprx, including a brief description of each.

Running pprx Suppose `pprx.opt` is a properly-formatted pprx options file as described Editing Options Files. Navigate to the directory where this file is located and type the following command into the terminal window.

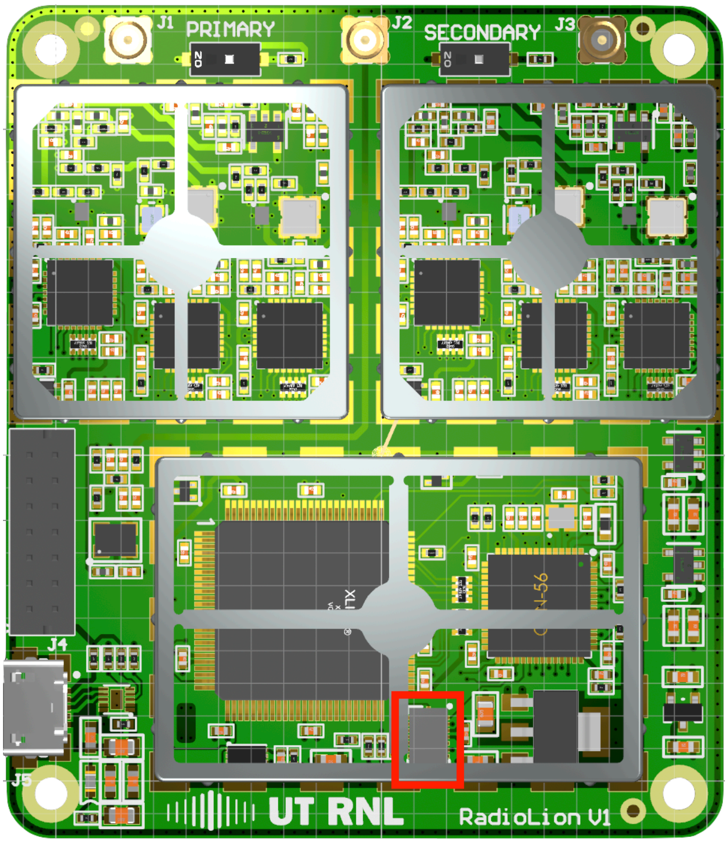
```
pprx -f pprx.opt
```

3.3 Precise Positioning Engine (ppengine)

This page describes how to configure, run, and interpret output from ppengine, which is an implementation of GRID, the General Radionavigation Interfusion Device.

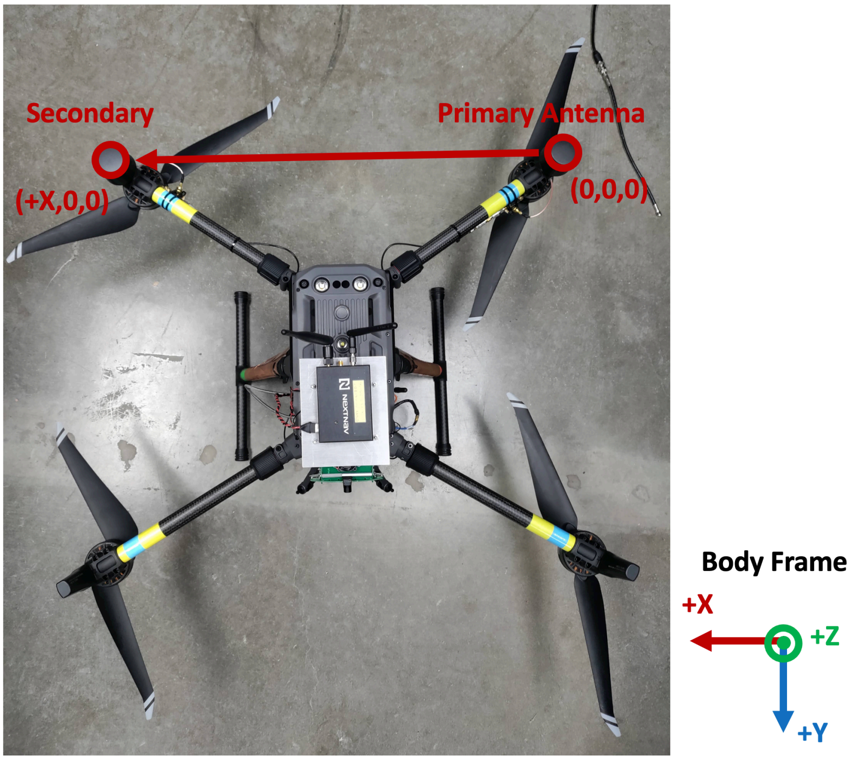
3.3.1 Steps to Enable PPengine

1. Determine if your system has (1) or (2) antennas capability. (2) is preferred if possible. Designate one as the primary antenna.
2. Fix the location of the RadioLion with respect to the antennas.
3. The following steps will describe how to measure the location of the RadioLion IMU with respect to the primary antenna. The RadioLion IMU is located in the red square in the following picture:



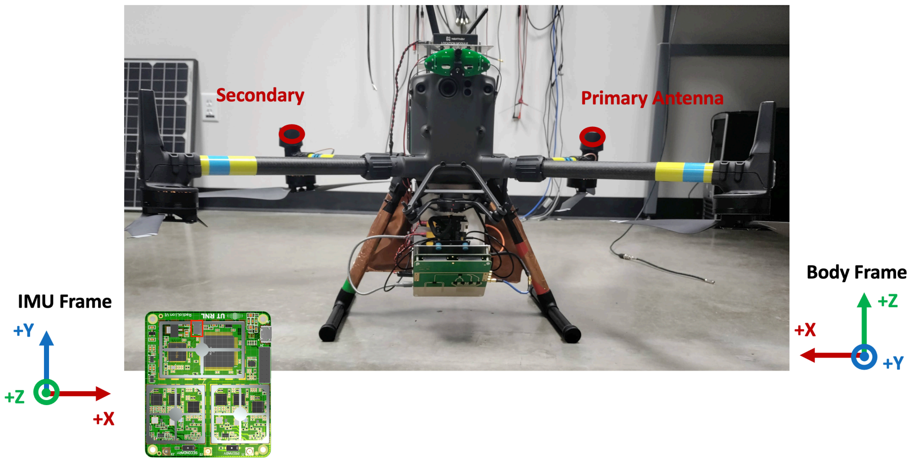
The measurements needed to enable **ppengine** are as follows:

1. In the `a2d.config` file, change the `BASELINE_LENGTH_CONSTRAINT` to the distance between the primary and secondary antenna. This measurement should be the positive X direction between antennas, measured in meters.
2. In the `ppose.config` file, change the `ESTIMATOR_TYPE` to `POSE_AND_TWIST_27`. Change the `POS_ALT1_ANTENNA_B` to the location of the secondary antenna with respect to the first antenna in the body frame. This is a three coordinate vector (x, y, x), measured in meters. Please be as precise as possible. See the picture below for an example of this calculation on a quad-copter.



3. In the `ppose.config` file, in the `IMU` Bank, there are numerous parameters that need to be filled out:

   - For `IMU_TYPE`, set it to `IMU_TYPE = CUSTOM2`.
   - `POS_IMU_B` is the position of the IMU in the body frame. This is a three coordinate vector (x, y, x), measured in meters. Please be as precise as possible. See the picture below for an example of this calculation on a quadcopter.



   - `ORIENTATION_IMU_B` is a quaternion (4 elements) that represents the rotational position of the IMU with respect to the body frame. See the picture below for an example of the IMU frame vs. the body frame.
   - To start `ACCELEROMETER_BIAS_U = 0 0 0`
   - To start `ACCELEROMETER_SCALE_FACTORS_U = 1 1 1`
   - To start `GYRO_BIAS_U = 0 0 0`
   - To start `GYRO_SCALE_FACTORS_U = 1 1 1`

4. Run the start-up script `stack.py` (see the page Operational Modes for more details) for a few minutes to obtain a good pprx and ppengine solution. Make sure ppengine is run with the `-s mat` option. Collect the output scripts `diagnostics.log` and `poseandtwist.mat`. Save them in a local folder.
5. Use the following two scripts: `IMU calibration` and `pose and twist analysis`.
6. In the `onlineImuCalibration.m` script, change the parameters `ORIENTATION_IMU_B`, `ACCELEROMETER_SCALE_FACTORS_U`, `GYRO_SCALE_FACTORS_U`, and `POS_IMU_B` to what you have set in the previous few steps. Download `this parser` and place it in the same folder that you have placed the `diagnostics.log` file. Make sure the datadir variable in the matlab script points to where your files are located.
   The script should output new parameters for `ORIENTATION_IMU_B`, `ACCELEROMETER_SCALE_FACTORS_U`, `GYRO_SCALE_FACTORS_U`. Take these new values and update the `ppose.config` file.
7. In the `poseAndTwistAnal.m` script, set the current_accel_bias to the value of `ACCELEROMETER_BIAS_U` (which for the first time will be 0 0 0) and set the current_gyro_bias to the value of `GYRO_BIAS_U` (which for the first time will be 0 0 0). Run the script. The script will output a few different variables, but you will want `bgU_mean` and `baU_mean`. These are the new values for the biases.
   Take these values and **ADD** them to the previous biases. The first time this is run, the current values are (0 0 0) so the new means will just become the new biases, but this will not be the case on subsequent runs. These new values for `ACCELEROMETER_BIAS_U` and `GYRO_BIAS_U` should be placed in the `ppose.config` file.
8. With the new values from this process, run ppengine again (you can use the same pprx file you've already collected) and save the `diagnostics.log` and `poseandtwist.mat` files once again. Repeat these steps a few times or until the numbers converge and stop changing (about 3-4 times).
9. Once this process is done, go back to `ppose.config` file, change the `ESTIMATOR_TYPE` to `POSE_AND_TWIST_18`. The IMU is calibrated and precise positioning mode is now tuned to your set-up.

3.3.2 Editing Options Files

ppengine is highly configurable via both command-line options and configuration parameters. Type `ppengine --help` to see a list of command-line options. These options are specified in a `.opt` file as seen in the *example* below:

```
--input-file ./ppengine_read
--verbose
--realtime
--config-file ./sbrtk.config
--config-file ./a2d.config
--ppose-config-file ./ppose.config
--skip 30
-t -1
-r 10
# For output to a gbx file
# -o ./ppengine.gbx
# For output to a named pipe
-o ./ppengine_write
--binary-only
```

3.3.3 Editing Configuration Parameters

ppengine configuration files are broken into configuration blocks. The start of each block is indicated by a block header, e.g., ESTIMATOR. A ppengine `.opt` file (i.e. ppengine.opt) has three corresponding configurations files:

- `ppose.config`: See this `example ppose configuration file`.
- `a2d.config`: See this `example attitude2d configuration file`.
- `sbrtk.config`: See this `example single baseline rtk configuration file`.

Each of the three configuration files uses different combinations of blocks. Each block in the `.config` file contains various configuration parameters. See each block below to view all of its configuration parameter choices and if applicable the choices for each parameter:

CDGNSS

Below is an *example* of the CDGNSS block:

```
[CDGNSS]
USE_IONO_CORR = TRUE
USE_TROPO_CORR = TRUE
IA_TEST_TYPE = DIFFERENCE
IA_OVERRIDE_TEST_RESULT = FALSE
IA_OVERRIDE_VALUE = FALSE
IA_USE_FIXED_FAILURE_RATE = TRUE
IA_DEFAULT_TEST_PARAM = 5
IA_ALLOWABLE_FAILURE_PROBABILITY = 1e-3
NUM_BASELINES = 1
ESTIMATION_STRATEGY = SINGLE_EPOCH
ADMISSIBLE_GENERIC_TYPES = GPS_L1_CA GALILEO_E1_BC SBAS_L1_I
TXID_EXCLUDE_LIST =
ELEVATION_MASK_ANGLE_DEG = 30
ENFORCE_STRICT_SELECTION = TRUE
STRICT_SELECTION_CN0_THRESHOLD = 39
STRICT_SELECTION_PHASE_LOCK_STAT_THRESHOLD = 0.7
INNOVATIONS_TEST_THRESHOLD_FACTOR = 10
SQRT_Q_TILDE_ATT = 0.01
UNDIFFERENCED_ZENITH_PSEUDORANGE_STD = 0.7
UNDIFFERENCED_ZENITH_PHASE_STD = 0.0035
ELEVATION_DEPENDENT_WEIGHTING = TRUE
MAXIMUM_AGE_OF_VALID_REFERENCE_DATA_SEC = 1.0
MAX_ELEVATION_FOR_INTEGER_INITIAL_GUESS_DEG = 15
```

The CDGNSS block contains all of the following configuration parameters:

USE_IONO_CORR

Default: True

Definition: Set true to correct for ionospheric delay when estimating the state

USE_TROPO_CORR

Default: True

Definition: Set true to correct for tropospheric (neutral atmospheric) delay when estimating the state

ILS_NUM_THREADS

Default: @HHH

Definition: Sets the number of threads ILS is allowed to use for modes capable of parallelization (e.g., grid search).

IA_TEST_TYPE

Default: @HHH

Definition: Specifies the type of Integer Aperture test to be performed. Select from the following options:

- `IALS`
- `RATIO`

- DIFFERENCE

IA_OVERRIDE_TEST_RESULT

Default: False

Definition: Additional configuration for selected integer aperture test in configuration parameter IA_TEST_TYPE. Allows for the Integer Aperture test to be effectively disabled by overriding the output with overrideValue so that all fixed or all float results can be seen. The test will still be performed so that all statistics from the test can be viewed, but the returned result is fixed regardless of the test result.

IA_OVERRIDE_VALUE

Default: False

Definition: Additional configuration for selected integer aperture test in configuration parameter IA_TEST_TYPE.

IA_USE_FIXED_FAILURE_RATE

Default: True

Definition: Additional configuration for selected integer aperture test in configuration parameter IA_TEST_TYPE. Allows fixed-failure-rate testing to be turned off with the test parameter set to a fixed default value

IA_DEFAULT_TEST_PARAM

Default: 0.8

Definition: Additional configuration for selected integer aperture test in configuration parameter IA_TEST_TYPE. Sets the default parameter that is used when not using the fixed-failure rate test

IA_ALLOWABLE_FAILURE_PROBABILITY

Default: 1e-3

Definition: Additional configuration for selected integer aperture test in configuration parameter IA_TEST_TYPE. Sets the maximum allowable failure probability for the fixed-failure rate test. Note that some of the Integer Aperture tests have fixed-failure rate algorithms that look up model coefficients from a table based (at least partially) on this failure rate. In these cases, the failure rate is rounded down to the nearest entry in the table. If the specified failure rate is lower than all values in the table, then the Integer Aperture test will throw an ASSERT on configuration.

IA_MINIMUM_TEST_PARAM

Default: -INFINITY

Definition: Additional configuration for selected integer aperture test in configuration parameter IA_TEST_TYPE. Sets the minimum test statistic that is allowed when using the fixed-failure rate test. This helps to protect against cases when the fixed-failure rate test sets a low or zero threshold due to an overly optimistic high model strength, often caused by unmodeled biases such as multipath.

USE_POINT_REPELLING_MODEL

Default: False

Definition: Production of the spherical grid used in constrained baseline ILS can employ a point repelling model to refine the grid points by making them more uniformly spaced. However, this model can take a while to run for baselines longer than 1 meter. Only relevant to the ATTITUDE_2D estimator.

SPHERE_GRID_SPACING_MULT

Default: 0.95

Definition: The spherical grid used in constrained baseline ILS requires grid points to be spaced by no more than half the smallest wavelength (L1 wavelength). However, it is usually a good idea to try for a little tighter than this because it is theoretically provable that uniform spacing on a sphere is impossible, past a certain number of points, and our algorithms for producing the grid are just an approximation. This value is a multiplication factor of LAMBDA_L1_M/2 to determine the goal grid spacing. Value cannot be larger than 1 or less than or equal to 0. Only relevant to the ATTI-

TUDE_2Destimator.
ELEVATION_MASK_ANGLE_RAD

Default: 0.25

Definition: Elevation mask angle, in radians. Signals arriving at the receiver from transmitters below the elevation mask angle will be excluded from the estimation solution. Set to -PI/2 to prevent elevation masking. In the receiver config file, the elevation mask angle is given in degrees as ELEVATION_-MASK_ANGLE_DEG.
BORESIGHT_ELEVATION_MASK_ANGLE_RAD

Default: -INFINITY

Definition: Boresight-relative elevation mask angle, in radians. See ELEVATION_MASK_ANGLE_RAD. This value is only used when vehicle attitude information is provided in the incoming GBX stream.
ENFORCE_STRICT_SELECTION

Default: True

Definition: When true, carrier phase and pseudorange measurements associated with a phase lock statistic below STRICT_SELECTION_PHASE_LOCK_STAT_THRESHOLD or C/N0 values below STRICT_SELECTION_CN0_THRESHOLD will be excluded from the precise navigation solution. When false, such measurements will be permitted.
STRICT_SELECTION_CN0_THRESHOLD

Default: 37.5

Definition: Carrier-to-noise ratio threshold for strict selection. See comments for ENFORCE_STRICT_SELECTION.
STRICT_SELECTION_PHASE_LOCK_STAT_THRESHOLD

Default: 0.55

Definition: Phase lock statistic threshold for strict selection. See comments for ENFORCE_STRICT_SELECTION.
INNOVATIONS_TEST_PF

Default: 1e-5

Definition: Threshold for double-difference pseudorange innovations test. (also known as the "float" innovations test). The innovations test is a chi-squared test using the normalized innovations squared (NIS) statistic, with a threshold corresponding to a constant false alarm rate of INNOVATION-S_TEST_PF. If the test is failed for a batch of DD GNSS observables, the POSITION_RTK and ATTI-TUDE_2D estimators reinitialize themselves. The POSE_AND_TWIST estimators reject that batch of observables but do not reinitialize.
FIX_INNOVATIONS_TEST_PF

Default: 1e-5

Definition: Threshold for double-difference carrier phase innovations test. (also known as the "fix" innovations test). The innovations test is a chi-squared test using the normalized innovations squared (NIS) statistic, with a threshold corresponding to a constant false alarm rate of FIX_INNOVATION-S_TEST_PF. Carrier phase NIS is approximated as chi-square distributed, but this not exactly the case because large residuals "tick over" to the next integer (therefore, large carrier phase residuals are not possible). If the test is failed for a batch of DD GNSS observables, the estimator falls back to a float solution (equivalent to a pseudorange-only solution when performing single-epoch integer ambiguity resolution)
DD_PSEUDORANGE_SCALAR_OUTLIER_THRESH_STD

Default: +INFINITY

Definition: Threshold, in standard deviations, for scalar DD pseudorange outlier rejection. If the scalar normalized innovations value for a DD pseudorange measurement exceeds this threshold, the corresponding GNSS satellite is excluded on all frequencies for all baselines.

See the section titled "Outlier Rejection using Pseudorange Innovations" in this paper.

SOFT_RESET_NIS_HISTORY_WINDOW

Default: 0

Definition: Window size, in epochs, of the false fix detection and recovery mechanism. If the double-difference carrier phase normalized innovations squared (NIS) over this window exceeds a chi-squared test threshold, a "soft reset" is performed, falling back to the "float-only" estimator. Only used by the POSE_AND_TWIST_15_MULTI_MODEL estimator.

See the section titled "False fix detection and recovery" in this paper.

SOFT_RESET_TEST_PF

Default: 1e-5

Definition: False-alarm probability used to set the false-fix detection chi-squared threshold. Only used by the POSE_AND_TWIST_15_MULTI_MODEL estimator.

SOFT_RESET_MIN_NUM_FIXES_IN_WINDOW

Default: 5

Definition: The false fix detector will only be able to fire if there have been at least SOFT_RESET_MIN_NUM_FIXES_IN_WINDOW integer fixes in the rolling window. Only used by the POSE_AND_TWIST_15_MULTI_MODEL estimator.

ALLOW_RESEED

Default: True

Definition: Enable the "re-seed" mechanism of the POSE_AND_TWIST_15_MULTI_MODEL estimator. If the re-seed criteria are met, the "float-only" estimator state will be "re-seeded" with the latest integer fixed solution. Only used by the POSE_AND_TWIST_15_MULTI_MODEL estimator.

See the section titled "Float-only estimator re-seeding" in this paper.

RESEED_LAST_FIX_NIS_THRESHOLD

Default: 1.0

Definition: Maximum (Carrier phase NIS / # carrier phase measurements) for the most recent fixed-integer solution to permit a re-seed. Only used by the POSE_AND_TWIST_15_MULTI_MODEL estimator.

RESEED_LAST_WINDOW_NIS_THRESHOLD

Default: 0.5

Definition: Maximum (window carrier phase NIS / # carrier phase measurements in window) for a re-seed. Only used by the POSE_AND_TWIST_15_MULTI_MODEL estimator.

RESEED_LAST_FIX_MIN_NDD

Default: 10

Definition: A re-seed is only performed if there were at least RESEED_LAST_FIX_MIN_NDD double-difference measurements used in the most recent fixed solution. Only used by the POSE_AND_TWIST_15_MULTI_MODEL estimator.

RESEED_MIN_FIXES_IN_WINDOW

Default: 10

Definition: A re-seed is only performed if there were at least RESEED_MIN_FIXES_IN_WINDOW fixed solutions in the rolling window used by the false fix detector. Only used by the POSE_AND_TWIST_15_MULTI_MODEL estimator.

SQRT_Q_TILDE_POS

Default: 0.2

Definition: The position process noise for some dynamics models is expressed in terms of SQRT_Q_TILDE_POS, the square root of the noise intensity. See Bar Shalom "Estimation with Applications to Tracking and Navigation" sections 6.2.1 to 6.2.3 for details. The units of SQRT_Q_TILDE_POS are as follows for each dynamics model:

STATIC meters/sqrt(sec) NEARLY_CONSTANT_VELOCITY meters/sqrt(sec^3) INERTIAL_MEASUREMENT_UNIT meters/sqrt(sec^3) NEARLY_CONSTANT_ACCELERATION meters/sqrt(sec^5)

SQRT_Q_TILDE_POS represents the standard deviation of error induced on position, velocity, or acceleration state elements by the process noise over a 1-second propagation step. The standard deviation corresponding to a T-second step is then approximated as sigmaX = sqrt(T)*SQRT_Q_TILDE_POS (see, e.g., Eq. 6.22-13 in Bar Shalom). This approximation is valid for short T; for long T, one needs to take multiple short propagation steps. Note that SQRT_Q_TILDE_POS is only used to generate a Q matrix for the INERTIAL_MEASUREMENT_UNIT dynamics model to cover any propagation step that may be required between thelatest IMU measurement before the measurement update and the measurement update itself.

SQRT_Q_TILDE_BODY_VEC

Default: [0 0 0]

Definition: SQRT_Q_TILDE_BODY_VEC represents the standard deviation of error induced on position, velocity, or acceleration state elements by the process noise over a 1-second propagation step, in the body X (nominally aligned with forward motion), Y (lateral), and Z (up) directions. This configuration option is used in place of SQRT_Q_TILDE when the BODY_NEARLY_CONSTANT_VELOCITY dynamics model is employed. It allows a simple way to constrain vehicle motion in the lateral and up directions.

SQRT_Q_TILDE_ATT

Default: 0.05

Definition: The attitude corollary to SQRT_Q_TILDE_POS is SQRT_Q_TILDE_ATT. The units of this quantity depend on if the state includes angular velocity or not and are as follows

no angular velocity rad/sqrt(sec) with angular velocity rad/sqrt(sec^3)

SQRT_Q_TILDE_ATT_VEC

Default: [0.05 0.05 0.05]

Definition: See SQRT_Q_TILDE_ATT.

UNDIFFERENCED_ZENITH_PSEUDORANGE_STD

Default: 1.0

Definition: Standard deviation of undifferenced pseudorange measurements assuming a transmitter at zenith, in meters. This value applies for all frequencies.

UNDIFFERENCED_ZENITH_PHASE_STD

Default: 0.004

Definition: Standard deviation of undifferenced carrier phase measurements assuming a transmitter at zenith, in meters. This value applies for all frequencies.

UNDIFFERENCED_ZENITH_STATIONARY_PSEUDORANGE_STD

Default: 5

Definition: Standard deviation of the undifferenced pseudorange measurements while rover is stationary, assuming a transmitter at zenith, in meters. This value can be set larger than UNDIFFERENCED_ZENITH_PSEUDORANGE_STD to account for the increased multipath that stationary rover antennas suffer. If unspecified by the user, this parameter defaults to the same value as UNDIFFERENCED_ZENITH_PSEUDORANGE_STD. Stationarity is determined using the velocity component of the rover standard navigation solution.

UNDIFFERENCED_ZENITH_TRANSIENT_PSEUDORANGE_STD

Default: 1

Definition: Alternate undifferenced pseudorange standard deviation at zenith, in meters, that is used during the FIRST_TRANSIENT period of the DLL.

ELEVATION_DEPENDENT_WEIGHTING

Default: True

Definition: Assert to weight undifferenced observables by 1/sin(el), where el is the elevation angle. This has the effect of de-weighting multipath-corrupted low-elevation signals. If not asserted, all observables are weighted equally.

OUTLIER_DETECTION

Default: False

Definition: Perform outlier detection on pseudorange and carrier phase measurements assuming (A) that the ratio test is passed and (B) that the minimum number of DD carrier phase measurements exceeds MINIMUM_NUMBER_DD_SIGNALS_FOR_OUTLIER_DETECTION. Multiple outliers can be detected during each epoch, if necessary.

MINIMUM_NUMBER_DD_SIGNALS_FOR_OUTLIER_DETECTION

Default: 8

Definition: If OUTLIER_DETECTION = true, continue to perform outlier detection on signals until the number of carrier phase measurements drops below this threshold.

MINIMUM_NUMBER_DD_SIGNALS

Default: 1

Definition: Minimum number of double-differenced (DD) signal pairs required to promote a float solution to a fixed solution. If the most recent solution was a fixed solution, then this minimum is ignored, unless FORCE_NDD_REQUIREMENT is TRUE.

FORCE_NDD_REQUIREMENT

Default: False

Definition: If FALSE, a fixed solution with NDD less than MINIMUM_NUMBER_NDD_SIGNALS will still be accepted if the previous solution was fixed and certain other criteria are met. If TRUE, MINIMUM_NUMBER_NDD_SIGNALS is strictly enforced.

MAXIMUM_AGE_OF_VALID_REFERENCE_DATA_SEC

Default: 0.5

Definition: Maximum allowed age of reference data relative to current rover epoch for the reference data to be considered valid. Note that if the reference data are more recent than the rover data, then the age will be negative and will always be less than the value below, which is assumed to be positive. Note also that if one sets this value smaller than the reference data's inter-epoch interval Te then, for each rover epoch processed, the age of data aod will be on the range -Te < aod <= 0. Thus, one should choose Te to be an acceptably small age of data.

REFERENCE_ECEF_POSITION

Default: @HHH

Definition: Precise XYZ ECEF position of the mean L1 phase center of the static reference antenna used in single-baseline RTK, expressed in meters in the same coordinates as the ephemeris records. The Boolean component indicates validity.

ROVER_ECEF_POSITION

Default: @HHH

Definition: Precise XYZ ECEF position of the mean L1 phase center of a static rover receiver antenna used in single-baseline RTK, expressed in meters in the same coordinates as the ephemeris records. The Boolean component indicates validity.

CONSTRAIN_ROVER_ECEF_POSITION

Default: False

Definition: Indicates whether the rover antenna position should be constrained to the value ROVER_ECEF_POSITION.

CONSTRAINT_SIGMA_METERS

Default: 1e-4

Definition: The rover antenna constraint, in meters. If CONSTRAIN_ROVER_ECEF_POSITION is asserted, then an artificial constraint with a standard deviation of CONSTRAINT_SIGMA_METERS will tie the rover position to ROVER_ECEF_POSITION.

CONSTRAIN_ROVER_TO_TRUTH_POSITION

Default: False

Definition: If true, constrains the rover to positions supplied via incoming "truth" PoseAndTwist GBX messages. This option requires the STATIC dynamics model.

TRUTH_POSITION_OFFSET_ENU

Default: @HHH

Definition: ENU frame offset added to incoming "truth" PoseAndTwist messages

ADMISSIBLE_GENERIC_TYPES

Default: @HHH

Definition: To avoid mismatched double differences, only a single SignalType:GenericType is admissible for each System at each center frequency. For example, at L2 only one of GPS_L2_CL, GPS_L2_CM, or GPS_L2_CLM is allowed. The following vector lists all admissible GenericTypes. .. .. code-block:: c .. std::vector<SignalType::GenericType> ADMISSIBLE_GENERIC_TYPES{ .. SignalType::GPS_L1_CA, SignalType::SBAS_L1_I, SignalType::GALILEO_E1_BC, .. SignalType::GPS_L2_CLM .. };

TXID_EXCLUDE_LIST

Default: []

Definition: List of TxIds and frequencies to exclude from participating in PpEstimator solution. Each element in the list is entered by the user according to the format

[Alphabetic System Designator][Number][Frequency Code]

The alphabetic system designator is the same as the RINEX convention: G (GPS), E (Galileo), S (SBAS), etc. The frequency code is either L1 or L2. For example, to exclude GPS PRN 23 only on L2, Galileo PRN 13 on all frequencies, and SBAS PRN 138 on L1:

TXID_EXCLUDE_LIST G23L2 E13L1 E13L2 S138L1

FORCE_PIVOT_LIST

Default: []

Definition: List of TxIds and frequencies that are commanded to be used as pivots in double differencing. The input format is the same as for TXID_EXCLUDE_LIST. If this list is empty, then pivots will be chosen according to the default internal algorithm.

GALILEO_E1_BC_TO_GPS_L1_CA_DD_DCB

Default: 0

Definition: Differential code bias in the double difference pseudorange observation for a pivot satellite with GenericType GPS_L1_CA and a non-pivot satellite with the GenericType indicated. If the pivot and non-pivot roles are reversed, then the negative of this DCB value is applied. This type of DCB arises when the rover receiver and the reference receiver have dissimilar front ends or code replica generator configuration.

SBAS_L1_I_TO_GPS_L1_CA_DD_DCB

Default: 0

Definition: Differential code bias in the double difference pseudorange observation for a pivot satellite with GenericType GPS_L1_CA and a non-pivot satellite with the GenericType indicated. If the pivot and non-pivot roles are reversed, then the negative of this DCB value is applied. This type of DCB arises when the rover receiver and the reference receiver have dissimilar front ends or code replica generator configuration.

FORCE_VALID_REFERENCE_OBSERVABLES

Default: False

Definition: Observables from GRID/pprx are marked invalid if the transmitter is indicated to be unhealthy, or if the health status is not known. When the following flag is asserted, all processed observables from the reference stream are considered valid. Rover stream observables are never forced valid; their validity can only be overriden by appropriate pprx configuration.

OUTLIER_EXCLUSION_DEPTH

Default: 0

Definition: Depth of outlier exclusion search. Signals are ordered from most to least likely to cause integer fixing failure and single-signal (N-choose-1) exclusion is attempted on each of the ordered signals up to and including the nth one, with n = OUTLIER_EXCLUSION_DEPTH. A value of 0 prevents outlier exclusion from being performed.

BACKWARD

Default: False

Definition: When true, the estimator is configured to run backward in time. Setting this parameter to true merely configures the estimator to expect and operate on a time-reversed data stream. It does not cause a normal data stream to be reversed.

A2D_MAX_ELEVATION_FOR_INTEGER_INITIAL_GUESS_RAD

Default: -1

Definition: Initial relative position guesses will be confined to +/- this elevation value, which should be positive, in radians. A value of -1 indicates that no elevation constraint should be applied.

A2D_SOLUTION_ELEVATION_MASK_RAD

Default: PI/2

Definition: If the solution lies outside the region +/- this threshold the integrityCheckPassed flag will be lowered. This parameter should be positive and in radians. A value of -1 indicates that no elevation threshold should be applied to the attitude solution.

USE_UNSCENTED_UPDATE

Default: True

Definition: @HHH

DISABLE_AFTER_PPOSE_INIT

Default: False

Definition: If true, this estimator will stop consuming incoming GBX reports whenever the attached PPose reports it is initialized. (this is a measure to save CPU usage when using PosRTK/A2D only for PPose initialization)

POS_PRI_ANTENNA_B

Default: [0 0 0]

Definition: Position of the PRIMARY GNSS antenna in the body frame. Used for antenna combining and inertial aiding of the POSITION_RTK solution. Only used by POSITION_RTK.

POS_ALT1_ANTENNA_B

Default: [0 0 0]

Definition: Position of the ALT_1 GNSS antenna in the body frame. Used for antenna combining and inertial aiding of the POSITION_RTK solution. Only used by POSITION_RTK.

POS_IMU_B

Default: [0 0 0]

Definition: Position of the IMU in the body frame. Used for inertial aiding of the POSITION_RTK solution. Only used by POSITION_RTK.

DO_ANTENNA_COMBINING

Default: False

Definition: If true, attitude-aided antenna combining is performed by POSITION_RTK using a fixed a priori vehicle attitude given by incoming MEASUREMENTS GBX reports. Only used by POSITION_RTK.

ANTENNA_COMBINING_BODY_FRAME_BIAS_STATE

Default: False

Definition: If true, an additional additive bias to POS_ALT1_ANTENNA_B is estimated online when performing antenna combining in POSITION_RTK. This bias is modeled as an Ornstein-Uhlenbeck process with the given steady-state uncertainty and time constant. Only used by POSITION_RTK.
ANTENNA_COMBINING_BODY_FRAME_BIAS_UNCERTAINTY_M

Default: NaN

Definition: @HHH
ANTENNA_COMBINING_BIAS_TIME_CONSTANT_SEC

Default: NaN

Definition: @HHH
CONDITION_ON_FIXED_SOLUTION

Default: True

Definition: If false, a fixed solution (if available) will be reported in the GBX output, but only the float solution will be committed to the filter state. Only used by POSITION_RTK.
MAX_VERTICAL_SEPARATION_FOR_FIX

Default: +INFINITY

Definition: If the vertical separation between the reference and rover (in meters) (determined by the rover's standard navigation solution) is greater than this threshold, an integer fix is not attempted. Only used by POSITION_RTK.
MAX_HORIZONTAL_SEPARATION_FOR_FIX

Default: +INFINITY

Definition: If the horizontal separation between the reference and rover (in meters) (determined by the rover's standard navigation solution) is greater than this threshold, an integer fix is not attempted. Only used by POSITION_RTK.
VERTICAL_SEPARATION_FOR_RESET

Default: +INFINITY

Definition: If this configuration option is set, an estimator reset is performed if the vertical separation between the reference and rover (in meters) as determined by the rover's standard navigation solution is between VERTICAL_SEPARATION_FOR_RESET and VERTICAL_SEPARATION_FOR_RESET+200.

ESTIMATOR

Below is an *example* of the ESTIMATOR block:

```
[ESTIMATOR]
ESTIMATOR_TYPE = ATTITUDE_2D
DYNAMICS_MODEL = STATIC
SPF_NUM_THREADS = 1
```

The ESTIMATOR block contains all of the following configuration parameters:
ALPHA

Default: 1e-3

Definition: Standard sigma point filter parameter alpha that governs the reach of the sigma points around the current best estimate
BETA

Default: 2

Definition: Standard sigma point filter parameter beta that governs the reach of the sigma points around the current best estimate

KAPPA

Default: 0

Definition: Standard sigma point filter parameter kappa that governs the reach of the sigma points around the current best estimate

DYNAMICS_MODEL

Default: STATIC

Definition: Assumed relative vector dynamics model, used by PositionRtk and Attitude2D estimators. (PoseAndTwist estimators require an IMU). Select from the following options:

- `STATIC`
- `NEARLY_CONSTANT_VELOCITY`
- `NEARLY_CONSTANT_ACCELERATION`
- `BODY_NEARLY_CONSTANT_VELOCITY` Applies a nearly constant velocity model in the body reference frame.
- `INERTIAL_MEASUREMENT_UNIT`

TAU_P

Default: 1000

Definition: The bias in the standard navigation solution position has various different intensities and time constants associated with multipath, satellite clock modeling errors, satellite ephemeris errors, and ionospheric modeling errors. We eliminate some of these with a turn-on-bias removal. The remainder we try to capture with a single Gauss-Markov bias model. We make the time constant long so that the bias estimate reverts only slowly to zero after a loss of the precise positioning solution. Standard navigation solution position bias time constant, in seconds. Only relevant to POSE_AND_TWIST_18.

SIGMA_P_STANDARD

Default: 1

Definition: Standard navigation solution position bias steady-state standard deviation, in meters Only relevant to POSE_AND_TWIST_18.

POS_ALT1_ANTENNA_B, POS_ALT2_ANTENNA_B

Default: @HHH

Definition: Position of the ALT1 and ALT2 antennas' L1 phase centers in meters in body coordinates (note that the position of the primary antenna's L1 phase center in body coordinates is (0;0;0)). Only relevant to POSE_AND_TWIST estimators. Initialize to nan to force user to input these before use

SIGMA_CONSTRAINED_BASELINE_ERROR_RAD

Default: 0.1

Definition: The standard deviation of the angular error in the constrained baseline vector, in radians. This is a parameter in the QUEST formulation of the error covariance model for the unit vector correponding to the constrained baseline. This default is appropriate for short-baseline setups such as quads or VR array. Only relevant to POSE_AND_TWIST estimators.

AZIMUTH_ONLY_FROM_CONSTRAINED_BASELINE

Default: True

Definition: Only exploit the azimuth angle from the constrained baseline vector. The standard deviation of the azimuth angle will continue to be SIGMA_CONSTRAINED_BASELINE_ERROR_RAD. Only relevant to POSE_AND_TWIST estimators.

ESTIMATOR_TYPE

Default: POSE_AND_TWIST_15

Definition: Indicates the type of sigma-point estimator that will be applied. Select from the following options:

PPose Estimators

- `POSE_AND_TWIST_15`
- `POSE_AND_TWIST_18`
- `POSE_AND_TWIST_27`
- `POSE_AND_TWIST_15_VISION`
- `POSE_AND_TWIST_15_MULTI_MODEL`

Position-only estimator

- `POSITION_RTK`

Constrained-baseline estimator

- `ATTITUDE_2D`

INCLUDE_STANDARD_NAVIGATION_SOLUTION_VELOCITY_MEASUREMENT

Default: False

Definition: Indicates whether the standard navigation solution velocity measurement should be included in the estimator

PRECISE_POS_MEASUREMENT_SIGMA_INFLATION_FACTOR

Default: 3

Definition: Inflation factor by which the precise position measurement error standard deviation is inflated to compensate for its being optimistic due to neglect of multipath errors. Only relevant to POSE_AND_TWIST estimators when consuming SBRTK and A2D reports.

OUTPUT_EVENT

Default: MEASUREMENT_UPDATE

Definition: The event that triggers output of the estimator's solution. Only relevant to POSE_AND_TWIST estimators. POSITION_RTK and ATTITUDE_2D output on every rover epoch. Select from the following options:

- `TIME_UPDATE`
- `MEASUREMENT_UPDATE`

Note that we don't allow output at both time and measurement updates to avoid updates with different solutions but marked at the same time.

INTEGRATOR_TYPE

Default: EULER_METHOD

Definition: Which type of dynamics integration to use. Only relevant to POSE_AND_TWIST estimators. Select from the following options:

- `EULER_METHOD`
- `PIECEWISE_CONSTANT_AW_OMEGAB`

SPF_NUM_THREADS

Default: @HHH

Definition: Sets the number of threads SPF is allowed to use for sigma point evaluation.

APPLY_VEHICLE_VELOCITY_CONSTRAINTS

Default: False

Definition: Indicates whether to apply vehicle near-zero-sideslip and near-zero-vertical velocity constraints. See documentation in `estimation of vehicle frame extrinsics`.. Also see `this`

`matlab script`. Only relevant to POSE_AND_TWIST estimators.
ORIENTATION_B2V

Default: @HHH

Definition: Orientation of the body (B) frame relative to the vehicle (V) frame, expressed as a quaternion. The quaternion should be formed such that RVB = navtbx:quat2dc(ORIENTATION_B2V) is the direction cosine matrix that translates a vector expressed in the B frame to one expressed in the V frame: vV = RVB*vB. Only relevant to POSE_AND_TWIST estimators.
POS_V0_B

Default: @HHH

Definition: Position of the vehicle center of rotation V0 (which is also the vehicle frame origin) in meters in body coordinates. Only relevant to POSE_AND_TWIST estimators.
SIGMA_VEHICLE_VELOCITY_CONSTRAINT_MPS

Default: 0.2 0.3

Definition: The standard deviations of the near-zero vehicle velocity constraints in the vehicle Y and Z directions, in meters per second. Only relevant to POSE_AND_TWIST estimators.
POLYNOMIAL_COEFFICIENTS_OMEGABZ_TO_V0VY

Default: 0 0

Definition: Polynomial coefficients relating the angular rate in the body Z direction and vVy, the y-component of the vehicle velocity with respect to W and expressed in V: vVy = P(0) + P(1)*omegaBz + P(2)*omegaBz^2 + …, where [P(0) P(1) … P(N)] is the ordering from the config file. Beware that this coefficient order convention is opposite Matlab's. Only relevant to POSE_AND_TWIST estimators.
APPLY_ZERO_VELOCITY_CONSTRAINT

Default: False

Definition: Indicates whether to apply a zero-velocity constraint triggered by the inertial sensor. Only relevant to POSE_AND_TWIST estimators.
SIGMA_ZERO_TRANSLATIONAL_VELOCITY_CONSTRAINT_MPS

Default: 0.02

Definition: The standard deviation of the zero translational velocity constraint triggered by the inertial sensor, in meters per second. This sigma applies to the vehicle Y and Z directions; the sigma in the vehicle X (forward) direction is scaled up internally to account for the greater uncertainty in the X direction (e.g., due to a slow vehicle roll). Only relevant to POSE_AND_TWIST estimators.
SIGMA_ZERO_ROTATIONAL_VELOCITY_CONSTRAINT_RPS

Default: 0.002

Definition: The standard deviation of the zero rotational velocity constraint triggered by the inertial sensor, in radians per second. This sigma applies equivalently to the IMU (U) roll, pitch, and yaw directions. Only relevant to POSE_AND_TWIST estimators.
INNOVATIONS_TEST_PF

Default: 1e-6

Definition: Innovations testing within SigmaPointFilter is based on the normalized innovations squared (NIS) statistic, NIS = dot(dzn, dzn), which under a consistent estimator is chi-square distributed with zr.n_elem degrees of freedom. A constant false-alarm rate test is performed using an NIS with a false-alarm probability of INNOVATIONS_TEST_PF (see chisquaredtest.h/cpp in gss). Only relevant to POSE_AND_TWIST estimators. GNSS-related innovations testing is configured in Cdgnss-Config.
PERFORM_INNOVATIONS_TESTING

Default: True

Definition: When false, innovations testing using INNOVATIONS_TEST_PF is not performed.

BACKWARD

Default: False

Definition: When true, the estimator is configured to run backward in time. Setting this parameter to true merely configures the estimator to expect and operate on a time-reversed data stream. It does not cause a normal data stream to be reversed.

CONSUME_EXTERNAL_CDGNSS_REPORTS

Default: True

Definition: When true, measurement updates are performed with incoming SingleBaselineRtk and Attitude2D GBX reports. Otherwise, SingleBaselineRtk and Attitude2D GBX reports are only used for filter initialization

ZERO_VELOCITY_UPDATE_DF_MAGNITUDE_THRESHOLD

Default: 0.8

Definition: Accelerometer and gyro thresholds used to detect vehicle stationarity for zero-velocity updates. The vehicle is considered stationary when the vector norms of deltas between the two most recent accelerometer (DF) and gyroscope (DOMEGATILDE) measurements are both below these thresholds for at least ZERO_VELOCITY_UPDATE_CONSECUTIVE_COUNT_THRESHOLD IMU measurements. Only relevant to POSE_AND_TWIST estimators.

ZERO_VELOCITY_UPDATE_DOMEGATILDE_MAGNITUDE_THRESHOLD

Default: 0.006

Definition: Accelerometer and gyro thresholds used to detect vehicle stationarity for zero-velocity updates. The vehicle is considered stationary when the vector norms of deltas between the two most recent accelerometer (DF) and gyroscope (DOMEGATILDE) measurements are both below these thresholds for at least ZERO_VELOCITY_UPDATE_CONSECUTIVE_COUNT_THRESHOLD IMU measurements. Only relevant to POSE_AND_TWIST estimators.

ZERO_VELOCITY_UPDATE_CONSECUTIVE_COUNT_THRESHOLD

Default: 10

Definition: Accelerometer and gyro thresholds used to detect vehicle stationarity for zero-velocity updates. The vehicle is considered stationary when the vector norms of deltas between the two most recent accelerometer (DF) and gyroscope (DOMEGATILDE) measurements are both below these thresholds for at least ZERO_VELOCITY_UPDATE_CONSECUTIVE_COUNT_THRESHOLD IMU measurements. Only relevant to POSE_AND_TWIST estimators.

PPOSE_ESTIMATOR

Below is an *example* of the PPOSE_ESTIMATOR block:

```
[PPOSE_ESTIMATOR]
ESTIMATOR_TYPE = POSE_AND_TWIST_18
# ESTIMATOR_TYPE = POSE_AND_TWIST_27 # used for IMU calibration
DYNAMICS_MODEL = INERTIAL_MEASUREMENT_UNIT
SIGMA_CONSTRAINED_BASELINE_ERROR_RAD = 0.1
AZIMUTH_ONLY_FROM_CONSTRAINED_BASELINE = TRUE
PRECISE_POS_MEASUREMENT_SIGMA_INFLATION_FACTOR = 1.0
INTEGRATOR_TYPE = EULER_METHOD
TAU_P = 100
SIGMA_P_STANDARD = 1.5
OUTPUT_EVENT = MEASUREMENT_UPDATE
POS_ALT1_ANTENNA_B = 0.195 0.0 0.0
PERFORM_INNOVATIONS_TESTING = FALSE
CONSUME_EXTERNAL_CDGNSS_REPORTS = TRUE # FALSE Tight Coupling
INNOVATIONS_TEST_THRESHOLD_FACTOR = 4
SPF_NUM_THREADS = 1
APPLY_VEHICLE_VELOCITY_CONSTRAINTS = FALSE
APPLY_ZERO_VELOCITY_CONSTRAINT = FALSE
```

The PPOSE_ESTIMATOR block contains all of the following configuration parameters:

ALPHA

Default: 1e-3

Definition: Standard sigma point filter parameter alpha that governs the reach of the sigma points around the current best estimate

BETA

Default: 2

Definition: Standard sigma point filter parameter beta that governs the reach of the sigma points around the current best estimate

KAPPA

Default: 0

Definition: Standard sigma point filter parameter kappa that governs the reach of the sigma points around the current best estimate

DYNAMICS_MODEL

Default: STATIC

Definition: Assumed relative vector dynamics model, used by PositionRtk and Attitude2D estimators. (PoseAndTwist estimators require an IMU). Select from the following options:

- `STATIC`
- `NEARLY_CONSTANT_VELOCITY`
- `NEARLY_CONSTANT_ACCELERATION`
- `BODY_NEARLY_CONSTANT_VELOCITY` Applies a nearly constant velocity model in the body reference frame.
- `INERTIAL_MEASUREMENT_UNIT`

TAU_P

Default: 1000

Definition: The bias in the standard navigation solution position has various different intensities and time constants associated with multipath, satellite clock modeling errors, satellite ephemeris errors, and ionospheric modeling errors. We eliminate some of these with a turn-on-bias removal. The remainder we try to capture with a single Gauss-Markov bias model. We make the time constant long so that the bias estimate reverts only slowly to zero after a loss of the precise positioning solution. Standard navigation solution position bias time constant, in seconds. Only relevant to POSE_AND_TWIST_18.

SIGMA_P_STANDARD

Default: 1

Definition: Standard navigation solution position bias steady-state standard deviation, in meters Only relevant to POSE_AND_TWIST_18.

POS_ALT1_ANTENNA_B, POS_ALT2_ANTENNA_B

Default: @HHH

Definition: Position of the ALT1 and ALT2 antennas' L1 phase centers in meters in body coordinates (note that the position of the primary antenna's L1 phase center in body coordinates is (0;0;0)). Only relevant to POSE_AND_TWIST estimators. Initialize to nan to force user to input these before use

SIGMA_CONSTRAINED_BASELINE_ERROR_RAD

Default: 0.1

Definition: The standard deviation of the angular error in the constrained baseline vector, in radians. This is a parameter in the QUEST formulation of the error covariance model for the unit vector correponding to the constrained baseline. This default is appropriate for short-baseline setups such as quads or VR array. Only relevant to POSE_AND_TWIST estimators.

AZIMUTH_ONLY_FROM_CONSTRAINED_BASELINE

Default: True

Definition: Only exploit the azimuth angle from the constrained baseline vector. The standard deviation of the azimuth angle will continue to be SIGMA_CONSTRAINED_BASELINE_ERROR_RAD. Only relevant to POSE_AND_TWIST estimators.

ESTIMATOR_TYPE

Default: POSE_AND_TWIST_15

Definition: Indicates the type of sigma-point estimator that will be applied. Select from the following options:

PPose Estimators

- `POSE_AND_TWIST_15`
- `POSE_AND_TWIST_18`
- `POSE_AND_TWIST_27`
- `POSE_AND_TWIST_15_VISION`
- `POSE_AND_TWIST_15_MULTI_MODEL`

Position-only estimator

- `POSITION_RTK`

Constrained-baseline estimator

- `ATTITUDE_2D`

INCLUDE_STANDARD_NAVIGATION_SOLUTION_VELOCITY_MEASUREMENT

Default: False

Definition: Indicates whether the standard navigation solution velocity measurement should be included in the estimator

PRECISE_POS_MEASUREMENT_SIGMA_INFLATION_FACTOR

Default: 3

Definition: Inflation factor by which the precise position measurement error standard deviation is inflated to compensate for its being optimistic due to neglect of multipath errors. Only relevant to POSE_AND_TWIST estimators when consuming SBRTK and A2D reports.

OUTPUT_EVENT

Default: MEASUREMENT_UPDATE

Definition: The event that triggers output of the estimator's solution. Only relevant to POSE_AND_TWIST estimators. POSITION_RTK and ATTITUDE_2D output on every rover epoch. Select from the following options:

- `TIME_UPDATE`
- `MEASUREMENT_UPDATE`

Note that we don't allow output at both time and measurement updates to avoid updates with different solutions but marked at the same time.

INTEGRATOR_TYPE

Default: EULER_METHOD

Definition: Which type of dynamics integration to use. Only relevant to POSE_AND_TWIST estimators. Select from the following options:

- `EULER_METHOD`

- `PIECEWISE_CONSTANT_AW_OMEGAB`

SPF_NUM_THREADS

Default: @HHH

Definition: Sets the number of threads SPF is allowed to use for sigma point evaluation.

APPLY_VEHICLE_VELOCITY_CONSTRAINTS

Default: False

Definition: Indicates whether to apply vehicle near-zero-sideslip and near-zero-vertical velocity constraints. See documentation in `estimation of vehicle frame extrinsics`. Also see `this matlab script`. Only relevant to POSE_AND_TWIST estimators.

ORIENTATION_B2V

Default: @HHH

Definition: Orientation of the body (B) frame relative to the vehicle (V) frame, expressed as a quaternion. The quaternion should be formed such that RVB = navtbx:quat2dc(ORIENTATION_B2V) is the direction cosine matrix that translates a vector expressed in the B frame to one expressed in the V frame: vV = RVB*vB. Only relevant to POSE_AND_TWIST estimators.

POS_V0_B

Default: @HHH

Definition: Position of the vehicle center of rotation V0 (which is also the vehicle frame origin) in meters in body coordinates. Only relevant to POSE_AND_TWIST estimators.

SIGMA_VEHICLE_VELOCITY_CONSTRAINT_MPS

Default: 0.2 0.3

Definition: The standard deviations of the near-zero vehicle velocity constraints in the vehicle Y and Z directions, in meters per second. Only relevant to POSE_AND_TWIST estimators.

POLYNOMIAL_COEFFICIENTS_OMEGABZ_TO_V0VY

Default: 0 0

Definition: Polynomial coefficients relating the angular rate in the body Z direction and vVy, the y-component of the vehicle velocity with respect to W and expressed in V: vVy = P(0) + P(1)*omegaBz + P(2)*omegaBz^2 + …, where [P(0) P(1) … P(N)] is the ordering from the config file. Beware that this coefficient order convention is opposite Matlab's. Only relevant to POSE_AND_TWIST estimators.

APPLY_ZERO_VELOCITY_CONSTRAINT

Default: False

Definition: Indicates whether to apply a zero-velocity constraint triggered by the inertial sensor. Only relevant to POSE_AND_TWIST estimators.

SIGMA_ZERO_TRANSLATIONAL_VELOCITY_CONSTRAINT_MPS

Default: 0.02

Definition: The standard deviation of the zero translational velocity constraint triggered by the inertial sensor, in meters per second. This sigma applies to the vehicle Y and Z directions; the sigma in the vehicle X (forward) direction is scaled up internally to account for the greater uncertainty in the X direction (e.g., due to a slow vehicle roll). Only relevant to POSE_AND_TWIST estimators.

SIGMA_ZERO_ROTATIONAL_VELOCITY_CONSTRAINT_RPS

Default: 0.002

Definition: The standard deviation of the zero rotational velocity constraint triggered by the inertial sensor, in radians per second. This sigma applies equivalently to the IMU (U) roll, pitch, and yaw directions. Only relevant to POSE_AND_TWIST estimators.

INNOVATIONS_TEST_PF

Default: 1e-6

Definition: Innovations testing within SigmaPointFilter is based on the normalized innovations squared (NIS) statistic, NIS = dot(dzn, dzn), which under a consistent estimator is chi-square distributed with zr.n_elem degrees of freedom. A constant false-alarm rate test is performed using an NIS with a false-alarm probability of INNOVATIONS_TEST_PF (see chisquaredtest.h/cpp in gss). Only relevant to POSE_AND_TWIST estimators. GNSS-related innovations testing is configured in CdgnssConfig.

PERFORM_INNOVATIONS_TESTING

Default: True

Definition: When false, innovations testing using INNOVATIONS_TEST_PF is not performed.

BACKWARD

Default: False

Definition: When true, the estimator is configured to run backward in time. Setting this parameter to true merely configures the estimator to expect and operate on a time-reversed data stream. It does not cause a normal data stream to be reversed.

CONSUME_EXTERNAL_CDGNSS_REPORTS

Default: True

Definition: When true, measurement updates are performed with incoming SingleBaselineRtk and Attitude2D GBX reports. Otherwise, SingleBaselineRtk and Attitude2D GBX reports are only used for filter initialization

ZERO_VELOCITY_UPDATE_DF_MAGNITUDE_THRESHOLD

Default: 0.8

Definition: Accelerometer and gyro thresholds used to detect vehicle stationarity for zero-velocity updates. The vehicle is considered stationary when the vector norms of deltas between the two most recent accelerometer (DF) and gyroscope (DOMEGATILDE) measurements are both below these thresholds for at least ZERO_VELOCITY_UPDATE_CONSECUTIVE_COUNT_THRESHOLD IMU measurements. Only relevant to POSE_AND_TWIST estimators.

ZERO_VELOCITY_UPDATE_DOMEGATILDE_MAGNITUDE_THRESHOLD

Default: 0.006

Definition: Accelerometer and gyro thresholds used to detect vehicle stationarity for zero-velocity updates. The vehicle is considered stationary when the vector norms of deltas between the two most recent accelerometer (DF) and gyroscope (DOMEGATILDE) measurements are both below these thresholds for at least ZERO_VELOCITY_UPDATE_CONSECUTIVE_COUNT_THRESHOLD IMU measurements. Only relevant to POSE_AND_TWIST estimators.

ZERO_VELOCITY_UPDATE_CONSECUTIVE_COUNT_THRESHOLD

Default: 10

Definition: Accelerometer and gyro thresholds used to detect vehicle stationarity for zero-velocity updates. The vehicle is considered stationary when the vector norms of deltas between the two most recent accelerometer (DF) and gyroscope (DOMEGATILDE) measurements are both below these thresholds for at least ZERO_VELOCITY_UPDATE_CONSECUTIVE_COUNT_THRESHOLD IMU measurements. Only relevant to POSE_AND_TWIST estimators.

BASELINE

Below is an *example* of the BASELINE block:

```
[BASELINE1]
REFERENCE_ANT_STREAM = REFERENCE
REFERENCE_ANT_GROUP = PRIMARY
ROVER_ANT_STREAM = ROVER
ROVER_ANT_GROUP = PRIMARY
```

The BASELINE block contains all of the following configuration parameters:

REFERENCE_ANT_STREAM

Default: REFERENCE

Definition: Indicates the reference antenna by specifying the stream (i.e., ROVER or REFERENCE).

REFERENCE_ANT_GROUP

Default: PRIMARY

Definition: Indicates the reference antenna by specifying the Group (i.e., PRIMARY, ALT1, etc.).

ROVER_ANT_STREAM

Default: ROVER

Definition: Indicates the rover antenna by specifying the stream (i.e., ROVER or REFERENCE).

ROVER_ANT_GROUP

Default: PRIMARY

Definition: Indicates the rover antenna by specifying the Group (i.e., PRIMARY, ALT1, etc.).

BASELINE_CONSTRAINED

Default: False

Definition: Boolean indicating if the baseline is length constrained

BASELINE_LENGTH_CONSTRAINT

Default: -1

Definition: Baseline length constraint, in meters. Enforced if BASELINE_CONSTRAINED = true.

BASELINE_VECTOR

Default: [1 0 0]

Definition: Antenna baseline unit vector in the body frame for a pair of baseline constrained antennas. Only used if BASELINE_CONSTRAINED = true.

REFERENCE_ANT_BORESIGHT_VECTOR

Default: [0 0 0]

Definition: Unit vectors pointing out the reference antenna boresight, expressed in the body frame.

ROVER_ANT_BORESIGHT_VECTOR

Default: [0 0 0]

Definition: Unit vectors pointing out the rover antenna boresight, expressed in the body frame.

IMU

Below is an *example* of the IMU block:

```
[IMU]
IMU_TYPE = CUSTOM2
POS_IMU_B = 0.073  0.026  -0.004
ORIENTATION_IMU_B = 0.00083628  0.010923  0.010181  0.99989
ACCELEROMETER_BIAS_U = -0.3  -0.3729   -0.1721
ACCELEROMETER_SCALE_FACTORS_U = 0.97582  0.97874  0.98361
GYRO_BIAS_U = 0  0  0
GYRO_SCALE_FACTORS_U = 1  1  1
```

The IMU block contains all of the following configuration parameters:

IMU_TYPE

Default: AUTOMOTIVE

Definition: Assumed relative vector dynamics model, used by PositionRtk and Attitude2D estimators. (PoseAndTwist estimators require an IMU). Select from the following options:

- `AUTOMOTIVE`
- `INDUSTRIAL`

- TACTICAL_LOW_QUALITY
- TACTICAL_HIGH_QUALITY
- NAVIGATION
- CUSTOM1
- CUSTOM2
- CUSTOM3
- CUSTOM
- LORD
- TASNAL

POS_IMU_B

Default: [0 0 0]

Definition: Position of the IMU's accelerometer triad in meters in body coordinates

ORIENTATION_IMU_B

Default: [0 0 0 1]

Definition: Orientation of the IMU frame relative to the body frame, expressed as a quaternion. The quaternion should be formed such that `RBU = navtbx:quat2dc(ORIENTATION_IMU_B_)` is the direction cosine matrix that translates a vector written in the IMU frame U to one written in B: vB = RBU*vU.

ACCELEROMETER_BIAS_U

Default: [0 0 0]

Definition: Static bias of the accelerometer, in m/sec^2 in the U frame.

ACCELEROMETER_SCALE_FACTORS_U

Default: [1 1 1]

Definition: Accelerometer scale factors along the X, Y, and Z axes in the U frame

GYRO_BIAS_U

Default: [0 0 0]

Definition: Static bias of rate gyros, in rad/s in the U frame

GYRO_SCALE_FACTORS_U

Default: [1 1 1]

Definition: Gyro scale factors along the X, Y, and Z axes in the U frame

SENSOR_ID

Default: 0

Definition: Identifies which IMU out of multiple possible

DIFF_TROPO

The DIFF_TROPO block contains all of the following configuration parameters:

MODEL_TYPE

Default: SAASTAMOINEN_NMF

Definition: Differential tropospheric delay model

- `SAASTAMOINEN_NMF`: Applies Saastamoinen zenith tropospheric delay models and Niell mapping functions, evaluated separately at the reference and rover positions. Configuration-provided, GBX-provided, or default surface weather conditions (temperature, pressure, humidity) apply for the reference station and are extrapolated to the rover station following page 3 of http://gauss2.gge.unb.ca/papers.pdf/ion52am.collins.pdf.
- `GBAS_MCGRAW`: An implementation of the GBAS/LAAS differential tropopsheric delay

model described in https://www.ion.org/publications/abstract.cfm?articleID=5959. This model was derived by applying simplifying assumptions (1/sin mapping function, equal SV elevation angle at reference and rover, etc.) to what is essentially the Saastamoinen model. This model probably does not offer any benefit beyond `SAASTAMOINEN_NMF`.

- `ONE_STATE_RESIDUAL_REL_ZTD_SAASTAMOINEN_NMF`: SAASTAMOINEN_NMF plus a single additional "residual relative zenith tropospheric delay" state added to the filter state vector. This additional state is modeled as a zero-mean OU process and is an additional bias added to the delta ZTD predicted by the Saastamoinen model. The online residual delay estimation only seems to be helpful when vertical separation between the reference and rover is on the order of 1 km or greater. Only PositionRtk can use this model.
- `ONE_STATE_RESIDUAL_REL_ZTD_NONE` NONE plus the same additional filter state as described above. In this case, the filter state is the total delta zenith tropospheric delay between the reference and rover.
- `ONE_STATE_RESIDUAL_REL_ZTD_GBAS_MCGRAW`: GBAS_MCGRAW plus the same additional filter state as described above.
- `NONE`

STEADY_STATE_DZTD_UNCERTAINTY

Default: 0.05

Definition: Steady-state uncertainty (in meters) for the Ornstein-Uhlenbeck (OU) process describing the differential zenith tropo delay in the "stateful" differential tropospheric delay mdoels (currently ``ONE_STATE_RESIDUAL_REL_ZTD_``*).

DZTD_TIME_CONSTANT

Default: 60

Definition: Time constant (in seconds) of the aforementioned differential ZTD OU process.

DEFAULT_AMBIENT_PRESSURE

Default: 101325

Definition: Default surface weather measurements used by models in the absence of configuration file parameters or incoming AtmosphericParameters GBX messages. These surface weather parameters are assumed to apply at the reference station, and in some models are extrapolated to calculate approximate atmospheric conditions at the rover. Ambient pressure, in Pascals

DEFAULT_AMBIENT_TEMPERATURE

Default: 294.261111

Definition: Ambient temperature, in Kelvin

DEFAULT_RELATIVE_HUMIDITY_PCT

Default: 60

Definition: Relative humidtiy, expressed as a percentage (0-100)

For `ppose.config` the following blocks are used:

- `PPOSE_ESTIMATOR`
- `BASELINE`
- `IMU`

For `a2d.config` the following blocks are used:

- `CDGNSS`
- `ESTIMATOR`
- `BASELINE`

For `sbrtk.config` the following blocks are used:

- `CDGNSS`
- `ESTIMATOR`
- `BASELINE`
- `DIFF_TROPO`

3.3.4 How to Run

Examining `--help` **Documentation** Running the following command

```
ppengine --help
```

will show all the command-line options available for pprx, including a brief description of each.

Running pprx Suppose `ppengine.opt` is a properly-formatted ppengine options file as described Section 3.2.1. Navigate to the directory where this file is located and type the following command into the terminal window.

```
ppengine -f ppengine.opt
```

3.4 Binflate

GRID/pprx natively outputs data in the GRID Binary Exchange (GBX) format. Use the binflate utility to interactively view the data (`-x` or `-p`), or convert them to ASCII log files (`-s log`), Matlab .mat files (`-s mat`), a Rinex 2.11 observables file (`-s rin`), or a Google Earth KML file (`-s kml`).

Have a look at the various binflate options by typing

```
binflate --help
```

To view GBX data in interactive mode, use the `-x` option:

```
binflate -i some_gbx_file.gbx -x
```

While in interactive mode, you can type `h` at the prompt to see a list of options.

To view select GBX reports graphically while in interactive mode, binflate requires gnuplot, which you can install as follows:

```
sudo apt install gnuplot
```

Then use the `-p` option:

```
binflate -i some_gbx_file.gbx -p
```

To create several ASCII log files described column-by-column in the documents below:

- `attitude2d.txt`
- `channel.txt`
- `display.txt`
- `iono.txt`
- `iq.txt`
- `iqtaps.txt`
- `navsol.txt`
- `poseandtwist.txt`

- sbrtk.txt
- scint.txt
- txinfo.txt

Call binflate like this:

```
binflate -i some_gbx_file.gbx -s log
```

For example, the format of channel.log is documented as below:

```
This file defines the columns of data in the channel.log file produced by the
GRID software receiver. Two types of measurement time stamps are given in
channel.log, both corresponding to the same event, namely, the observables
measurement event:

    (1) Raw Receiver Time (RRT): This time stamp is linked directly to the
        receiver's sampling clock.  It starts at zero when the receiver is
        initialized and is never interrupted or adjusted by the code.  RRT is
        guaranteed to be non-decreasing.  The increment between subsequent RRT
        values is only approximately uniform; it may vary by up to a few
        milliseconds as GRID adjusts its internal updates to keep all channels
        at approximately the same receiver time.

    (2) Offset Receiver Time (ORT): This time stamp is equal to RRT plus an
        offset that brings the result close (within a few ms) to true GPS
        time:

        ORT = RRT + tOffset

       GRID automatically adjusts tOffset every so often to bring ORT within a
       few ms of true GPS time.  When tOffset is adjusted, a small jump in ORT
       is introduced and concomitant shifts occur in the pseudorange and
       carrier phase data.  For maximum resolution, ORT is given in separate
       columns for week, whole second, and fractional seconds.

=============================================================================


Column         Quantity

1 ----------- RRT week number.

2 ----------- RRT seconds of week.

3 ----------- ORT week number.  A value of 9999 indicates that the ORT is not
              yet valid.

4 ----------- ORT whole seconds of week.

5 ----------- ORT fractional second.

6 ----------- Apparent Doppler frequency in Hz (positive for approaching TXs).

7 ----------- Beat carrier phase in cycles (changes in the same sense as
              pseudorange).

8 ----------- Pseudorange in meters.

9 ----------- Carrier-to-noise ratio (C/N0) in dB-Hz.

10 ---------- Flag indicating whether (1) or not (0) the pseudorange and
              carrier phase measurements are valid.
```

```
11 ---------- Error indicator: An encoded 8-bit value indicating various error
              states.  Let <i> indicate the ith bit value, with <0> being the
              LSB.  An error state is indicated by assertion of the
              corresponding bit:
              <0>     Phase tracking anomaly (large phase error)
              <1>     Spoofing detected
              <2>     Possible half cycle phase offset
              <3:7>   Reserved

12 ---------- Channel status indicator:
              0       STATUS_NULL
              1       STATUS_ALLOCATED
              2       STATUS_ACQUIRING
              3       STATUS_ACQUIRED
              4       STATUS_SYMBOL_LOCK
              5       STATUS_FREQ_LOCK
              6       STATUS_PHASE_LOCK
              7       STATUS_DATA_LOCK

13 ---------- Signal GenericType. See signaltype.h for GenericType values.

14 ---------- Transmitter identification number (TXID).

15 ---------- Signal Group.

16 ---------- PLL's phase lock statistic.

17 ---------- Distortion statistic.

=============================================================================
```

If you'd rather inflate the GBX files to binary Matlab .mat files for easy loading into Matlab, call binflate like this:

```
binflate -i some_gbx_file.gbx -s mat
```

You can draw the .mat files into Matlab as follows:

```
load channel.mat;
     channel = channel'; % Transpose to put in columnar format
```

The format of the matrices that get loaded into Matlab is the same as that of the ASCII files.

Likewise, the `-s rin` and `-s kml` options inflate to Rinex 2.11 format and Google Earth KML format. By default, binflate inflates GBX files into Matlab .mat files.

3.5 Operational Modes

3.5.1 Receiver Autonomous Operation

Receiver Autonomous Operation is defined as when the GNSS receiver **is not** connected to a reference station. Refer to the Sofware User Guide on pprx to set up the receiver for autonomous operation.

3.5.2 Carrier-Phase Differential (RTK) Operation

Carrier-Phase Differential (RTK) Operation is defined as when the GNSS receiver **is** connected to a reference station via wifi. Refer to the Sofware User Guide on pprx and ppengine to set up the rover receiver.

The default configuration files described above are for a rover receiver connected to a reference network. To be able to use these as provided, set up a local reference station running pprx and refnet_server. Ensure that the reference station's IP address is accessible to the rover computer. Put the reference station's IP address in pprx's options file; e.g., `~git/gss_top/run/pprx.opt`:

```
--ref-host=10.0.0.2    # <---- Put the reference station's IP address here
```

To ensure the reference station is connected to the rover receiver:

- `refnet_monitor` is a simple utility that allows one to connect to and monitor GBX traffic received from a reference server. `refnet_monitor --help` offers rather straight-forward command-line options. When done, kill it using Ctrl-C. Example usage:

  ```
  refnet_monitor --hostname 10.0.0.2
  ```

Instructions on how run the rover receiver with reference station:

1. Run `refnet_monitor` to check that the reference receiver is working and the network connection is good:

   ```
   refnet_monitor --hostname 10.0.0.2
   ```

2. Create pipes for realtime operation:

   ```
   cd ~/git/gss_top/scripts
   ./make_pipes
   ```

3. Run the Python startup script:

   ```
   cd ~/git/gss_top/scripts
   ./stack.py
   ```

4. tmux window 2 will show the progress of PpEngine as it processes signals tracked by pprx.
5. After 30 seconds, pprx will be stably tracking several satellites. At this point, PpEngine will attemp to obtain a precise Attitude 2D fix, a Single Baseline RTK fix, or both, depending on how ppengine.opt is configured. If there are enough satellites with clean enough measurements, the indicator 'fixed' will appear. If PpEngine fails to obtain a fix, see below for troubleshooting.
6. If PpEngine has been configured in ppengine.opt to produce a Pose solution, it will display this solution at the bottom of the text display. This solution gives the East-North-Up (ENU) coordinates of the primary snapon-antenna-assembly antenna relative to the reference antenna, and the pitch, roll, and yaw (3-1-2 Euler rotation) angles that define the antenna assembly's attitude relative to the local ENU frame.
7. To quit, go to the tmux window where PpEngine is running (tmux window 2), hit ''Ctrl-C'', and then type ''tkill'' at the prompt.