IonoMoni Users Manual

Yuzhi Zhang, Yihong Zhang, Qi Liu

Contents

1. Introduction

The ionosphere has a significant impact on Global Navigation Satellite System (GNSS) signals, introducing refraction and diffraction effects that can severely degrade positioning performance. With the current peak of Solar Cycle 25, increasingly prominent ionospheric space weather events and disturbances have further intensified adverse effects on GNSS applications. On the other hand, dual-frequency GNSS observations enable effective monitoring of ionospheric conditions, serving as a crucial foundation for the mitigation of ionospheric effects on GNSS positioning and the advancement of research into the physical mechanisms of ionospheric variability. The extraction and analysis of ionospheric parameters are essential for both scientific research and practical applications.

In this context, we developed the IonoMoni ionospheric parameter computation software, based on the secondary development of the GREAT-PVT software within the Visual Studio 2022 environment. The IonoMoni program offers multiple functionalities, including:

1. It supports STEC extraction using the dual-frequency Carrier-to-Code Leveling (CCL) method.

2. It supports STEC extraction based on the undifferenced and uncombined precise point positioning (UCPPP) method.

3. It supports VTEC conversion based on ionospheric mapping function and STEC.

4. It supports the calculation of the Rate of TEC Index (ROTI), which is a widely used indicator for detecting ionospheric irregularities of ionospheric diffractive effects.

5. It supports the estimation of the Along Arc TEC Rate (AATR), an effective metric for monitoring ionospheric disturbances, especially during geomagnetic storms or in equatorial and polar regions.

IonoMoni is able to process multi-GNSS observation data, including GPS, BDS, Galileo, and GLONASS constellations, and is compatible with both Rinex 2.x and Rinex 3.x file formats.

Furthermore, IonoMoni features batch processing and plotting capabilities, enabling efficient extraction and visualization of ionospheric parameters across multiple stations and multiple days.

2. Mathematical models

2.1 Basic Observation Equations

The original GNSS pseudorange ($\mathrm{P}^{s}_{r,n}$) and phase ($\mathrm{L}^{s}_{r,n}$) observations can be expressed as:

$$P^{s}_{r,n} = \rho^{s}_{r} + t_{r} - t^{s} + I^{s}_{r,n} + m^{s}_{r,w} Z_{r,w} + b_{r,n} - b^{s}_{n} + e^{s}_{r,n} \tag{1}$$

$$L^{s}_{r,n} = \rho^{s}_{r} + t_{r} - t^{s} - I^{s}_{r,n} + m^{s}_{r,w} Z_{r,w} + \lambda_{n}\left(N^{s}_{r,n} + B_{r,n} - B^{s}_{n}\right) + \phi^{s}_{r,n} + \varepsilon^{s}_{r,n} \tag{2}$$

P^{s}_{r} : Geometric distance from the satellite antenna phase center (s) to the receiver antenna phase center (r), in meters;

n: frequency of observation (such as 1, 2);

t_{r} and t^{s}: Receiver and satellite clock biases, in meters.

$\mathrm{Z}_{r,w}$: Zenith wet delay of the troposphere, in meters.

$\mathrm{m}^{s}_{r,w}$: Mapping function for the zenith wet delay.

$\mathrm{I}^{s}_{r,n}$: Ionospheric delay at the frequency n, in meters.

$\mathrm{N}^{s}_{r,n}$: Integer carrier phase ambiguity, in cycles.

$\mathrm{b}_{r,n}$ and b^{s}_{n}: Receiver and satellite pseudorange hardware delays, in meters, typically stable over a day or a continuous arc.

$\mathrm{B}_{r,n}$ and B^{s}_{n}: Receiver and satellite phase bias, in cycles, which can be divided into constant and time-varying parts. The constant part can be fully absorbed by the ambiguity parameter.

$\phi^s_{r,n}$: wind-up effects of carrier phase.

$e^s_{r,n}$ and $\varepsilon^s_{r,n}$: Sum of observation noise and multipath effects for pseudorange and phase observations, respectively.

It is worth noting that the tropospheric delay, whether wet or dry, can be expressed as the product of the zenith tropospheric delay and the corresponding mapping function. The dry component is usually corrected using a priori models (e.g., the Saastamoinen model), while the wet component, due to its high uncertainty, is typically estimated as a parameter. Other errors, such as satellite and receiver antenna phase center offsets (PCO) and variations (PCV), relativistic effects, solid and ocean tides, and phase wind-up, can be corrected using existing error models.

2.2 Dual-frequency carrier-to-code leveling method

The Geometry-Free (GF) combination of dual-frequency pseudorange and carrier phase measurements can be expressed as:

$$P^s_{r,4} = P^s_{r,1} - P^s_{r,2} = I^s_{r,1} - I^s_{r,2} + \mathrm{DCB}_r - \mathrm{DCB}^s + e^s_{r,1} - e^s_{r,2} \tag{3}$$

$$L^s_{r,4} = L^s_{r,1} - L^s_{r,2} = - I^s_{r,1} + I^s_{r,2} + \lambda_1\left(N^s_{r,1} + B_{r,1} - B^s_1\right) - \lambda_2\left(N^s_{r,2} + B_{r,2} - B^s_2\right) + \phi^s_{r,1} - \phi^s_{r,2} + \varepsilon^s_{r,1} - \varepsilon^s_{r,2} \tag{4}$$

where $\mathrm{DCB}_r = \mathrm{b}_{r,1} - \mathrm{b}_{r,2}$, and $\mathrm{DCB}^s = \mathrm{b}^s_1 - \mathrm{b}^s_2$.

And the Smoothed $\widetilde{\mathrm{P}^s_{r,4}}$ at epoch t can be expressed as follows:

$$\widetilde{\mathrm{P}^s_{r,4}}(t) = w_t P^s_4(t) + (1 - w_t)(\widetilde{\mathrm{P}^s_4}(t-1) + [L^s_4(t) - L^s_4(t-1)]) \quad (t > 1) \tag{5}$$

where t is the epoch, $\mathrm{w_t}$ is the weight factor. Cycle slips and gross errors in the carrier phase observations should be removed before using the carrier phase observations to smooth the pseudorange observations. Here, both dual-frequency pseudorange code observations and ionospheric residual observations are used to detect cycle slips and gross errors.

And the slant total electron content (STEC) can be extracted from GNSS dual-frequency observations as follow:

$$\mathrm{STEC(t)} = \frac{f_1^2 f_2^2}{40.3(f_2^2 - f_1^2)}\left(\widetilde{\mathrm{P}^s_{r,4}}(t) + \mathrm{DCB}^s(\mathrm{t}) - \mathrm{DCB}_r(\mathrm{t})\right) \tag{6}$$

2.3 Uncombined Precise Point Positioning method

The basic functional model for pseudorange and phase in the undifferenced and uncombined model is as follows:

$$\begin{cases} P^s_{r,n} = \mu^s_r \cdot x + t_r - \bar{t}^s_{IF} + \gamma_n \cdot I^s_{r,1} + m^s_{r,w} Z_{r,w} + b_{r,n} - b^s_n \\ + \left(\alpha_{ij} b^s_i + \beta_{ij} b^s_j\right) + e^s_{r,n} \\ L^s_{r,n} = \mu^s_r \cdot x + t_r - \bar{t}^s_{IF} - \gamma_n \cdot l^s_{r,1} + m^s_{r,w} Z_{r,w} + \lambda_n\left(B_{r,n} - B^s_n + N^s_{r,n}\right) \\ + \left(\alpha_{ij} b^s_i + \beta_{ij} b^s_j\right) + \varepsilon^s_{r,n} \end{cases} \tag{7}$$

It is important to note that the hardware delay terms depend on the frequencies used. Specifically, when using the frequencies corresponding to the ionosphere-free clock products (n = i or n = j):

$$\begin{cases} P^s_{r,i} = \mu^s_r \cdot x + \hat{t}_r - \bar{t}^s_{IF} + \hat{t}^s_{r,i} + m^s_{r,w} Z_{r,w} + e^s_{r,i} \\ P^s_{r,j} = \mu^s_r \cdot x + \hat{t}_r - \bar{t}^s_{IF} + \gamma_{ij} \hat{t}^s_{r,i} + m^s_{r,w} Z_{r,w} + e^s_{r,j} \\ L^s_{r,i} = \mu^s_r \cdot x + \hat{t}_r - \bar{t}^s_{IF} - \hat{t}^s_{r,i} + m^s_{r,w} Z_{r,w} + \widehat{N}^s_{r,i} + \varepsilon^s_{r,i} \\ L^s_{r,j} = \mu^s_r \cdot x + \hat{t}_r - \bar{t}^s_{IF} - \gamma_{ij} \cdot \hat{I}^s_{r,i} + m^s_{r,w} Z_{r,w} + \widehat{N}^s_{r,j} + \varepsilon^s_{r,j} \end{cases} \tag{8}$$

where $\hat{\mathrm{t}}_\mathrm{r}$ is the estimated receiver clock bias parameter, $\hat{\mathrm{I}}^s_{r,i}$, $\widehat{\mathrm{N}}^s_{r,i}$ and $\widehat{\mathrm{N}}^s_{r,j}$ are the estimated ionospheric parameter and ambiguity parameters for the two frequencies, respectively, expressed as:

$$\begin{cases}\hat{I}_{r,i}^{s} = I_{r,i}^{s} + \beta_{ij}\left(b_{r,i} - b_{r,j}\right) - \beta_{ij}\left(b_{i}^{s} - b_{j}^{s}\right) \\ \widehat{N}_{r,i}^{s} = N_{r,i}^{s} + \left[\left(B_{r,i} - B_{i}^{s}\right) - \left(b_{r,i} - b_{i}^{s}\right)\right] + 2\beta_{ij}\left[\left(b_{r,i} - b_{i}^{s}\right) - \left(b_{r,j} - b_{j}^{s}\right)\right] \\ \widehat{N}_{r,j}^{s} = N_{r,j}^{s} + \left[\left(B_{r,j} - B_{j}^{s}\right) - \left(b_{r,j} - b_{j}^{s}\right)\right] + 2\beta_{ij}\left[\left(b_{r,j} - b_{j}^{s}\right) - \left(b_{r,i} - b_{i}^{s}\right)\right]\end{cases} \tag{9}$$

It is important to note that when using the same frequencies as the ionosphere-free clock products, the satellite pseudorange hardware delay $\left(b_i^s - b_j^s\right)$ can be absorbed by the ionospheric parameter.

If different frequencies are used, the satellite pseudorange hardware delay must be corrected using corresponding pseudorange bias products (e.g., differential code bias). The parameters to be estimated in the dual-frequency undifferenced and uncombined model include:

$$X = \left(x \quad \hat{t}_r \quad Z_{r,w} \quad \hat{I}_{r,i}^{s} \quad \widehat{N}_{r,i}^{s} \quad \widehat{N}_{r,j}^{s}\right)^T \tag{10}$$

The corrected ionospheric delay can be expressed as:

$$I_{r,1}^{s} = \hat{I}_{r,1}^{s} - \beta_{ij}\left(b_{r,i} - b_{r,j}\right) + \beta_{ij}\left(b_i^s - b_j^s\right) \tag{11}$$

Since the estimated quantity is the slant ionospheric delay on the first frequency, the corresponding STEC can be extracted as:

$$\mathrm{STEC} = \frac{f_1^2 \cdot I_{r,1}^{s}}{40.3} \tag{12}$$

2.4 VTEC conversion from STEC and mapping function

The function for converting STEC to VTEC can be expressed as:

$$\mathrm{VTEC} = \frac{\mathrm{STEC}}{M(\varepsilon)} \tag{13}$$

where, $M(\varepsilon)$ is the mapping function dependent on the satellite elevation angle ε.

The Single-Layer Model mapping function can be expressed as:

$$M(\varepsilon) = \left(1 - \left(\frac{R_E \cos\varepsilon}{R_E + h}\right)^2\right)^{-1/2} \tag{14}$$

where R_E is the mean radius of the Earth, h is the assumed ionospheric shell height.

The Modified Single-Layer Model mapping function can be expressed as:

$$M_{\mathrm{MSLM}}(\varepsilon) = \left[1 - \left(\frac{R_E \sin(\alpha z)}{R_E + h}\right)^2\right]^{-1/2} \tag{15}$$

where z is the zenith angle, and α is an empirical correction factor.

The F&K mapping function can be expressed as:

$$M_{\mathrm{F\&K}}(\varepsilon) = \frac{1 + \frac{R_E + h}{R_E}}{\sin\varepsilon + \sqrt{\left(\frac{R_E + h}{R_E}\right)^2 - \cos^2\varepsilon}} \tag{16}$$

The Ou mapping function can be expressed as:

$$\mathrm{mf}_{\mathrm{temp}} = \frac{1}{\cos\left[\arcsin\left(\frac{R_E \cos\varepsilon}{R_E + h}\right)\right]} \tag{17}$$

where $\mathrm{mf}_{\mathrm{temp}}$ is an intermediate variable representing the obliquity factor.

$$M_{\mathrm{Ou}}(\varepsilon) = \begin{cases}\sin(\varepsilon + 50°) \cdot \mathrm{mf}_{\mathrm{temp}}, & \varepsilon < 40° \\ \mathrm{mf}_{\mathrm{temp}}, & \varepsilon \geq 40°\end{cases} \tag{18}$$

The Fanselow mapping function can be expressed as:

$$\mathrm{M}_{\mathrm{Fanselow}}(\varepsilon) = \frac{\sqrt{(\mathrm{R_E}\sin\varepsilon)^2+2\mathrm{R_E}\mathrm{h}_2+\mathrm{h}_2^2}-\sqrt{(\mathrm{R_E}\sin\varepsilon)^2+2\mathrm{R_E}\mathrm{h}_1+\mathrm{h}_1^2}}{\mathrm{h}_2-\mathrm{h}_1} \tag{19}$$

where $\mathrm{h}_1=h-35000, \mathrm{h}_2=h+70000$ are the lower and upper boundaries of the ionospheric layer, respectively.

2.4 ROTI Index

ROTI is defined as the standard deviation of the ROT (Rate Of TEC) over some time interval. It is calculated as follows, where Ln, λ_n, and f_n are the phase measurement, wavelength, and frequency for the nth frequency.

$L_4^S(i)$ is the geometry-free phase combination at time i:

$$L_4^S(i) = L_{r,1}^S(i) - L_{r,2}^S(i) \tag{20}$$

ROT (in TECU/minute) is calculated as：

$$\mathrm{ROT}(i) = \frac{L_4^S(i)-L_4^S(i-1)}{\Delta t\times 10^{16}\times 40.3\times\left(\frac{1}{f_1^2}-\frac{1}{f_2^2}\right)} \tag{21}$$

TECU (TEC Unit) is defined as 10^{16}electrons per m^2. Δt is the time difference between the epochs, in minutes. Finally, ROTI, calculated over N epochs, is

$$\mathrm{ROTI}(i) = \sqrt{\frac{1}{N}\sum_{j=i-N}^{i}\left(\mathrm{ROT}(j)-\overline{\mathrm{ROT}}\right)^2} \tag{22}$$

2.5 AATR Index

The variation of the STEC between two consecutive observations can be computed for a given epoch, i, as:

$$\Delta STEC_r^s(i) = L_{4,r}^s(i) - L_{4,r}^j(i-1) \tag{23}$$

where r denotes the receiver and s the satellite.

For a given transmitter-receiver pair, the variation over time of the STEC (Along Arc STEC Rate, AASR_r^s) can be computed as:

$$AASR_r^s(i) = \frac{\Delta STEC_r^s(i)}{\Delta T} \tag{24}$$

where i indicates the observation epoch, $\Delta\mathrm{STEC}_r^s(i)$ corresponds to the difference of STECs between two consecutive observations in the same satellite-receiver arc and ΔT is the elapsed time between these consecutive observations (typically 30 or 60 seconds). We apply this mapping squared in order to mitigate the effect of low elevation rays (instead of applying a cut-off angle). Then

$$AATR_r^s(i) = \frac{1}{M(\varepsilon)}\cdot\frac{AASR_r^s(i)}{M(\varepsilon)} = \frac{\Delta STEC_r^s(i)}{\left(M(\varepsilon)\right)^2\cdot\Delta T} \tag{25}$$

Finally, based on the previous equation (20), the AATR indicator is defined as the hourly weighted RMS of the instantaneous AATR values, computed over a predefined period for all satellites in view from a particular station, resulting in the AATR index for a given receiver.

$$RMS_{AATR,r}(I) = \sqrt{\frac{1}{N}\sum_{i=I}^{I+\Delta I}\sum_{s=1}^{n_{sat}(i)}\left(AATR_r^s(i)\right)^2} \tag{26}$$

where N is total number of observations during the selected interval $\Delta\mathrm{I}$ (typically one hour), after having summed all satellites in view, $n_{sat}(i)$, at every epoch i.

3. Environmental Requirements

3.1 Environmental Requirements

The executable CUI AP for Windows in the package was built using Microsoft Visual Studio 2022 on Windows 11 (64-bit), with the ReleaseWithXml configuration. All required dynamic link libraries are included in the directory.

3.2 License

IonoMoni is released as open-source software under the GNU General Public License, version 3 (GPLv3). This license permits anyone to use, modify, and redistribute the software, provided that the same license terms are preserved and the source code remains accessible.

For detailed information, please refer to the full license text at: https://www.gnu.org/licenses/gpl-3.0.html.

3.3 Python Environment

IonoMoni includes several auxiliary Python scripts that require a Python environment and a set of dependencies listed in the requirements.txt file. For optimal compatibility, Python version 3.13 is recommended. It is further advised to use a virtual environment manager, such as Conda or venv, to ensure reproducibility and isolate dependencies.

4. Installation

Extract the software package ***IonoMoni.zip*** to a suitable directory <install_dir>. The directory structure of IonoMoni is as follows:

Table 4.1 IonoMoni Directory Structure

IonoMoni	
./src	IonoMoni Source Code
./include	IonoMoni Header Files
./lib	Static Libraries and Export Files
./build_resources	Dynamic Link Libraries and Sample data
./batch_process	Python Batch Processing Script
./plot	Python Plotting Script
./poleut1	Earth Orientation Parameters Generation Program
./doc	User manual and Sample XML
./CMakeLists.txt	CMake Build Configuration File
./CMakePresets.json	CMake Build Preset Settings

The following instructions show how to build IonoMoni on Windows.

(1) Using Visual Studio 2022, open the IonoMoni root folder as a CMake project. In the configuration menu, select "Release with XML", and wait for CMake to configure and generate the build system automatically, as shown in Figures 4.1 and 4.2.

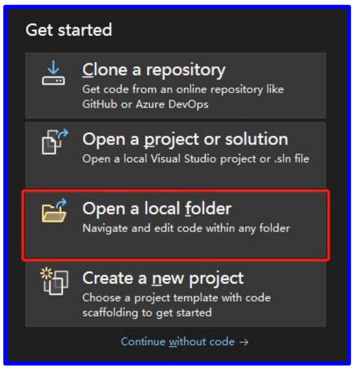

Figure 4.1 open the IonoMoni root folder

Figure 4.2 select "Release with XML"

(2) Copy all contents from the ***./build_resources*** directory to ***./out/build/release-with-xml/***.

(3) In Visual Studio 2022, click Build → Build All (or press Ctrl + Shift + B) to compile and generate the IonoMoni.exe executable in the ./out/build/release-with-xml/ directory, as shown in Figures 4.3.

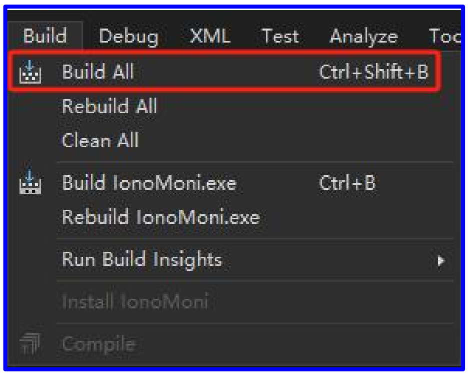

Figure 4.3 generate the IonoMoni.exe executable

5. Execution of the sample data

The input files for the sample case are located in the build_resources folder and should be copied to ***./out/build/release-with-xml/*** during the installation process. (Different modes of IonoMoni require different input files. Details about data preparation can be found in section ***8.Data Acquisition***.) The folder structure is shown as follows：

Table 5.1 Sample Data Folder Structure

Sample data	
./gnss	GNSS data folder
./obs	Observation data folder
./model	System folder
./xml	XML configuration folder

We selected GNSS observation data from stations HKWS and ABPO on November 30, 2023, as the sample dataset. IonoMoni is configured and executed through XML files that define both the processing mode and associated parameters. In the following sections, we provide a detailed explanation of XML configuration usage. By making minor adjustments to the provided example XML templates, users can quickly and effectively begin using IonoMoni. The configuration elements are as follows:

Table 5.2 XML Configuration Elements and Corresponding Descriptions

Item	**Descriptions**	**Element in XML File**
General Settings (First Level Element)		<gen>
Function	Set the operation mode: CCL_STEC, PPP_STEC, ROTI, and AATR.	<function>
Begin Time	Set begin time in form of GPS time. The format is "YYYY-MM-DD hh:mm:ss".	<beg>
End Time	Set end time in form of GPS time. The format is "YYYY-MM-DD hh:mm:ss".	<end>
Station List	Set the list of stations participating in the solution (4-character string).	<rec>
Satellite System	Set the satellite systems to be used in the solution.	<sys>
Sampling Interval	Set the sampling frequency of the observations.（Currently, only 30-second intervals are supported.）	<int>
Estimation Method	Default is filtering.	<est>
Input File Settings (First Level Element)		<inputs>
RINEX OBS File	RINEX observation file for processing. Note that it supports RINEX 2.10, 2.11, 2.12, 3.00, 3.01, 3.02, 3.03, 3.04, 3.05 OBS.	<rinexo>
RINEX NAV File	RINEX navigation file for processing. Note that it supports RINEX 2.10, 2.11, 2.12, 3.00, 3.01, 3.02, 3.03, 3.04, 3.05.	<rinexn>
Precise Clock File	Precise clock file for PPP_STEC processing.	<rinexc>
3Days Precise Clock File	Precise clock file for CCL_STEC ROTI AATR processing.	<THREE_SP3>
Precise Orbit File	Precise orbit file for processing.	<sp3>
DCB File	DCB file for differential code bias correction.	<bias>
Antenna File	Satellite antenna information file for phase center correction.	<atx>
Ocean Tide File	Ocean tide file for tide correction.	<blq>
Planetary Ephemeris File	Planetary ephemeris file for calculating planetary parameters.	<de>
EOP Parameter File	Earth Orientation Parameters file for rotation matrix calculation.	<eop>
Output File Settings (First Level Element)		<outputs>
Log File	Log file for recording output information.	<log>
Results	Output file for PPP calculation results.	<flt>
Processing Settings (First Level Element)		<process>
Sliding window for ROTI	Sliding window size used in ROTI computation (in epochs).	<roti_window>
Unit of ROT	Unit of ROT, used for AATR and ROTI processing: -sec:TECU/sec -min:TECU/min	<rot_unit>
Computation interval for AATR	Generate one AATR every specified number of epochs.	<aatr_interval>
Minimum arc length	Minimum arc length threshold (in epochs).	<arc_min_length>
Ionospheric Height	Height of the ionospheric layer (in meters)	<ion_height>

Item	Descriptions	Element in XML File
Mapping Function	Type of mapping function used for VTEC conversion (0=SLM, 1=MSLM, 2=F_K, 3=Ou, 4=Fanselow)	<mapping_function>
Cutoff Elevation Angle	Minimum elevation angle for satellite observations.	<minimum_elev>
Phase Observations	Whether to use carrier phase observations: - true: yes - false: no	<phase>
Tropospheric Parameters	Whether to estimate tropospheric parameters: - true: yes - false: no	<tropo>
Ionospheric Parameters	Whether to estimate ionospheric parameters: - true: yes - false: no	<iono>
Doppler Observations	Whether to use Doppler observations: - true: yes - false: no	<doppler>
Tropospheric Model	Tropospheric model to be used.	<tropo_model>
A Priori Sigma of Estimated Parameters	A priori sigma for station coordinates.	<sig_init_crd>
	A priori sigma for station velocity.	<sig_init_vel>
	A priori sigma for tropospheric parameters.	<sig_init_ztd>
	A priori sigma for ambiguities.	<sig_init_amb>
	A priori sigma for Galileo inter-system and inter-frequency biases.	<sig_init_gal>
	A priori sigma for GLONASS inter-system and inter-frequency biases.	<sig_init_glo>
	A priori sigma for BDS inter-system and inter-frequency biases.	<sig_init_bds>
	A priori sigma for ionospheric parameters.	<sig_init_iono>
Observation Combination	Observation combination method for processing: - IONO_FREE : IF, supports dual-frequency - RAW_ALL : UC, supports dual-frequency and multi-frequency	<obs_combination>
Maximum Posterior Residual	Threshold for posterior residual editing.	<max_res_norm>
Coordinate Constraint	Coordinate constraint method: - est: estimated constraint - fix: fixed constraint - kin: kinematic constraint	<crd_constr>
Kinematic Mode	Whether to enable kinematic mode: - true: yes - false: no	<pos_kin>
Minimum Number of Satellites	Minimum threshold for the number of satellites in the solution.	<min_sat>
Observation Weighting	Method for determining observation weights.	<obs_weight>
BDS Code Bias Correction	Whether to correct BDS satellite code biases: - true: yes - false: no	<bds_code_bias_corr>
Cycle Slip Detection	Cycle slip detection model: - default: default model	<slip_model>

Item	Descriptions	Element in XML File
Observation Frequency	Number of observation frequencies used in the solution.	<frequency>
Filter Settings (First Level Element)		<filter>
Filter Algorithm	Filter algorithm to be used: - srcf: square-root filter - kalman: Kalman filter	method_flt
A Priori Noise of Estimated Parameters	White noise for station coordinates.	noise_crd
	White noise for station velocity.	noise_vel
	White noise for receiver clock drift.	noise_dclk
	White noise for receiver clock offset.	noise_clk
	White noise for ionospheric parameters.	noise_vion
	Random walk noise for tropospheric parameters.	rndwk_ztd
	Random walk noise for ambiguities.	rndwk_amb
	Random walk noise for GLONASS ISB/IFB.	rndwk_glo
	Random walk noise for Galileo ISB/IFB.	rndwk_gal
	Random walk noise for BDS ISB/IFB.	rndwk_bds
	Random walk noise for GPS IFB.	rndwk_gps
Ambiguity Resolution Settings (First Level Element)		<ambiguity>
Fixing Mode	Whether to perform ambiguity resolution: - NO: float - SEARCH: fix	<fix_mode>
UPD Mode	Use UPD products for ambiguity resolution.	<upd_mode>
Partial Ambiguity Fixing	Whether to perform partial ambiguity fixing: - YES: yes - NO: no	<part_fix>
Minimum Number of Fixed Ambiguities	Minimum number of fixed ambiguities in partial ambiguity fixing mode.	<part_fix_num>
Ratio Value	Ratio test value for ambiguity resolution using the LAMBDA method.	<ratio>
Reference Satellite	Whether to set a reference satellite: - YES: yes - NO: no	<set_refsat>
Minimum Common Time	Minimum common observation time for ambiguity resolution.	<min_common_time>
Extra Wide-Lane Observations	Settings for ambiguity resolution with different observation combinations: - alpha&maxdev: confidence interval parameters - maxsig: maximum sigma value	<extra_widelane_decision>
Wide-Lane Observations		<widelane_decision>
Narrow-Lane Observations		<narrowlane_decision>
Satellite Settings (First Level Element)		<gps>/<bds>/<gal>/ <glo>
A Priori Sigma of Observations	Sigma for pseudorange observations.	sigma_C
	Sigma for carrier phase observations.	sigma_L

Item	Descriptions	Element in XML File
Frequency	Satellite frequency, corresponding to the frequency band (options: 1/2/3/4/5).	<freq>
Satellite	Satellite PRN number.	<sat>
Frequency Band	Observation frequency bands for different satellite systems: - GPS: 1->L1, 2->L2, 5->L5 (Currently, only L1 and L2 frequencies are supported.) - GAL: 1->E1, 5->E5a, 7->E5b, 8->E5, 6->E6 (Currently, only E1 and E5a frequencies are supported.) - BDS: 2->B1I, 7->B2I, 6->B3I, 1->B1C, 5->B2a, 9->B2b, 8->B2a+b (Currently, only B1I B2I and B3I frequencies are supported.) - GLO: 1->G1, 2->G2	<band>

5.1 Running Example for CCL_STEC Methoda

```
<gen>
    <function>CCL_STEC     </function>   <!--> function: CCL_STEC PPP_STEC ROTI AATR  <!-->
    <beg> 2023-04-10 00:00:00 </beg>     <!--> begin time <!-->
    <end> 2023-04-10 23:59:30 </end>     <!--> end time <!-->
    <int> 30                  </int>     <!--> sampling interval <!-->
    <sys> GPS                 </sys>     <!--> system ex: GPS GAL GLO BDS <!-->
    <rec> HRAO POTS           </rec>     <!--> site (4-char upper) <!-->
    <est> FLT                 </est>     <!--> Estimator : FLT <!-->
</gen>
```

Figure 5.1.1 XML configuration example for the CCL_STEC mode

```
<inputs>
     <rinexo>  <!--> rinex obs file <!-->
         obs\hrao1000.23o
         obs\CORD00ARG_R_20231000000_01D_30S_MO.rnx
         obs\POTS00DEU_R_20231000000_01D_30S_MO.rnx
     </rinexo>
    <rinexn> gnss\brdc1000.23n </rinexn>  <!--> rinex nav file <!-->
    <rinexc> gnss\CODOMGXFIN_20231000000_01D_30S_CLK.CLK </rinexc>  <!--> precise satellite clock offset file <!-->
    <sp3> gnss\CODOMGXFIN_20231000000_01D_05M_ORB.SP3 </sp3>  <!--> precise orbit file <!-->

    <THREE_SP3>  <!--> CCL ROTI AATR precise orbit file <!-->
        gnss\CODOMGXFIN_20230990000_01D_05M_ORB.SP3
        gnss\CODOMGXFIN_20231000000_01D_05M_ORB.SP3
        gnss\CODOMGXFIN_20231010000_01D_05M_ORB.SP3

    </THREE_SP3>
    <de> model\jpleph_de405_great </de>  <!--> Planetary ephemeris file <!-->
    <atx> model\igs20_2290.atx </atx>  <!--> Antenna correction file <!-->
    <blq> model\oceanload </blq>  <!--> oceanload file <!-->
    <eop> model\poleut1 </eop>  <!--> ERP file <!-->
    <bias> gnss\CASOMGXRAP_20231000000_01D_01D_DCB.BSX </bias>  <!--> DCB file <!-->
</inputs>
```

Figure 5.1.2 XML configuration example for the CCL_STEC mode

```
    <gps sigma_C="0.6" sigma_L="0.01">  <!--> GPS pseudorange sigma & phase sigma <!-->
    <sat>  <!--> satellite prn <!-->
        G01 G02 G03 G04 G05 G06 G07 G08 G09 G10
        G11 G12 G13 G14 G15 G16 G17 G18 G19 G20
        G21 G22 G23 G24 G25 G26 G27 G28 G29 G30
        G31 G32
    </sat>
    <band> 1 2 </band>  <!--> the satellite frequency, currently only 1 and 2 frequencies are supported. <!-->
    <freq> 1 2 </freq>  <!--> satellite frequency, corresponding to band, optional value 1/2/3/4/5 <!-->
</gps>
<glo sigma_C="0.6" sigma_L="0.01">  <!--> GLO pseudorange sigma & phase sigma <!-->
    <sat>
        R01 R02 R03 R04 R05 R06 R07 R08 R09 R10
        R11 R12 R13 R14 R15 R16 R17 R18 R19 R20
        R21 R22 R23 R24
    </sat>
    <band> 1 2 </band>
    <freq> 1 2 </freq>
```

Figure 5.1.3 XML configuration example for the CCL_STEC mode

```
<gal sigma_C="0.6" sigma_L="0.01">  <!--> GAL pseudorange sigma & phase sigma <!-->
    <sat>
        E01 E02 E03 E04 E05 E06 E07 E08 E09 E10
        E11 E12 E13 E14 E15 E16 E17 E18 E19 E20
        E21 E22 E23 E24 E25 E26 E27 E28 E29 E30
        E31 E32 E33 E34 E35 E36
    </sat>
    <band> 1 5 </band>  <!--> the satellite frequency, currently only 1 and 5 frequencies are supported. <!-->
    <freq> 1 2 </freq>
</gal>
<bds sigma_C="0.6" sigma_L="0.01">  <!--> BDS pseudorange sigma & phase sigma <!-->
    <sat>
        C01 C02 C03 C04 C05 C06 C07 C08 C09 C10
        C11 C12 C13 C14 C15 C16 C17 C18 C19 C20
        C21 C22 C23 C24 C25 C26 C27 C28 C29 C30
        C31 C32 C33 C34 C35 C36 C37 C38 C39 C40
        C41 C42 C43 C44 C45 C46
    </sat>
    <band> 2 7 </band>  <!--> the satellite frequency, currently only 2 and 6, 7 frequencies are supported. <!-->
    <freq> 1 2 </freq>
</bds>
```

Figure 5.1.4 XML configuration example for the CCL_STEC mode

```
<process>
    <roti_window>          10              </roti_window><!--> sliding window size (in epochs)<!-->
    <rot_unit>             min             </rot_unit><!--> unit of ROT (sec/min)<!-->
    <aatr_interval>        60             </aatr_interval><!--> AATR calculation interval (in epochs) <!-->
    <arc_min_length>       10              </arc_min_length><!--> minimum arc length threshold (in epochs)<!-->
    <ion_height>           350000          </ion_height><!--> height of the ionospheric layer (in meters) <!-->
    <mapping_function>     0               </mapping_function><!--> type of mapping function used for VTEC conve
    <minimum_elev>         7               </minimum_elev><!--> cut-off satellite elevation(deg) <!<!-->
```

Figure 5.1.5 XML configuration example for the CCL_STEC mode

5.2 Running Example for PPP_STEC Method

The settings for each satellite system are specified in Figures 5.1.3 and 5.1.4.

```
<gen>
    <function>PPP_STEC     </function>  <!--> function: CCL_STEC PPP_STEC ROTI AATR  <!-->
    <beg> 2023-04-10 00:00:00 </beg>    <!--> begin time <!-->
    <end> 2023-04-10 23:59:30 </end>    <!--> end time <!-->
    <int> 30                  </int>    <!--> sampling interval <!-->
    <sys> GPS                 </sys>    <!--> system ex: GPS GAL GLO BDS <!-->
    <rec> HRAO POTS           </rec>    <!--> site (4-char upper) <!-->
    <est> FLT                 </est>    <!--> Estimator : FLT <!-->
</gen>
```

Figure 5.2.1 XML configuration example for the PPP_STEC mode

```xml
<inputs>
      <rinexo>  <!--> rinex obs file <!-->
          obs\hrao1000.23o
          obs\CORD00ARG_R_20231000000_01D_30S_MO.rnx
          obs\POTS00DEU_R_20231000000_01D_30S_MO.rnx
      </rinexo>
    <rinexn> gnss\brdc1000.23n </rinexn>  <!--> rinex nav file <!-->
    <rinexc> gnss\COD0MGXFIN_20231000000_01D_30S_CLK.CLK </rinexc>  <!--> precise satellite clock offset file <!-->
    <sp3> gnss\COD0MGXFIN_20231000000_01D_05M_ORB.SP3 </sp3>  <!--> precise orbit file <!-->

    <THREE_SP3>  <!--> CCL ROTI AATR precise orbit file <!-->
        gnss\COD0MGXFIN_20230990000_01D_05M_ORB.SP3
        gnss\COD0MGXFIN_20231000000_01D_05M_ORB.SP3
        gnss\COD0MGXFIN_20231010000_01D_05M_ORB.SP3

    </THREE_SP3>
    <de> model\jpleph_de405_great </de>  <!--> Planetary ephemeris file <!-->
    <atx> model\igs20_2290.atx </atx>  <!--> Antenna correction file <!-->
    <blq> model\oceanload </blq>  <!--> oceanload file <!-->
    <eop> model\poleut1 </eop>  <!--> ERP file <!-->
    <bias> gnss\CAS0MGXRAP_20231000000_01D_01D_DCB.BSX </bias>  <!--> DCB file <!-->
</inputs>
```

Figure 5.2.2 XML configuration example for the PPP_STEC mode

```xml
<minimum_elev>          7                   </minimum_elev><!--> cut-off satellite elevation(deg) <!<!-->

<phase>                 true                </phase>  <!--> use phase obs (true/false) <!-->
<tropo>                 true                </tropo>  <!--> estimate trop param (true/false) <!-->
<iono>                  true                </iono>  <!--> estimate iono param (true/false) <!-->
<doppler>               false               </doppler>  <!--> use doppler obs (true/false) <!-->
<tropo_model>           saastamoinen        </tropo_model>  <!--> trop model <!-->
<sig_init_crd>          30                  </sig_init_crd>  <!--> initial sigma of coordinate <!-->
<sig_init_vel>          10                  </sig_init_vel>  <!--> initial sigma of velocity <!-->
<sig_init_ztd>          10                  </sig_init_ztd>  <!--> initial sigma of ztd <!-->
<sig_init_amb>          30                  </sig_init_amb>  <!--> initial sigma of ambiguity <!-->
<sig_init_gal>          10                  </sig_init_gal>  <!--> initial sigma of Galileo isb/ifb <!-->
<sig_init_glo>          10                  </sig_init_glo>  <!--> initial sigma of GLONASS isb/ifb <!-->
<sig_init_bds>          10                  </sig_init_bds>  <!--> initial sigma of BDS isb/ifb <!-->
<sig_init_vion>         100                 </sig_init_vion>  <!--> initial sigma of slant iono <!-->
<obs_combination>       RAW_ALL             </obs_combination>  <!--> obs comb type (IONO_FREE/RAW_ALL) <!-->
<max_res_norm>          3                   </max_res_norm>  <!--> posterior residual threshold <!-->
<crd_constr>            est                 </crd_constr>  <!--> coordinate constraint method (EST/FIX/KIN/SIMU_KIN) <!-->
<pos_kin>               false               </pos_kin>  <!--> kinematic mode (true/false) <!-->
<min_sat>               5                   </min_sat>  <!--> min satellite number <!-->
<obs_weight>            SINEL               </obs_weight>  <!--> weigh model of obs <!-->
<bds_code_bias_corr>    true                </bds_code_bias_corr>  <!--> whether to correct BDS codeBias (true/false) <!-->
<slip_model>            default             </slip_model>  <!--> cycle slip detect method <!-->
<frequency>             2                   </frequency>  <!--> frequency number <!-->
```

Figure 5.2.3 XML configuration example for the PPP_STEC mode

```xml
<process>
    <roti_window>           10              </roti_window><!--> sliding window size (in epochs)<!-->
    <rot_unit>              min             </rot_unit><!--> unit of ROT (sec/min)<!-->
    <aatr_interval>         60              </aatr_interval><!--> AATR calculation interval (in epochs) <!-->
    <arc_min_length>        10              </arc_min_length><!--> minimum arc length threshold (in epochs)<!-->
    <ion_height>            350000          </ion_height><!--> height of the ionospheric layer (in meters) <!-->
    <mapping_function>      0               </mapping_function><!--> type of mapping function used for VTEC conve
    <minimum_elev>          7               </minimum_elev><!--> cut-off satellite elevation(deg) <!<!-->
```

Figure 5.2.4 XML configuration example for the PPP_STEC mode

```xml
<filter
    method_flt="kalman"
    noise_crd="0"
    noise_vel="1"
    noise_clk = "1000"
    noise_dclk="100"
    noise_vion="100"
    rndwk_ztd="6"
    rndwk_amb="0"
    rndwk_glo = "20"
            rndwk_gal = "20"
            rndwk_bds = "20"
            rndwk_gps = "20"
/>
<ambiguity>
    <fix_mode> NO </fix_mode>  <!--> ambiguity fixed mode (NO/SEARCH) <!-->
    <upd_mode> UPD </upd_mode>  <!--> upd mode <!-->
    <part_fix> YES </part_fix>  <!--> part_fix (YES/NO) <!-->
    <part_fix_num> 4 </part_fix_num>  <!--> threshold in partial ambiguity fixing <!-->
    <ratio> 2.0 </ratio>  <!--> threshold in LAMBDA method <!-->
    <set_refsat> YES </set_refsat>  <!--> set_refsat (YES/NO) <!-->
    <min_common_time> 1 </min_common_time>  <!--> minimum common time/amblonds <!-->
    <extra_widelane_decision maxdev = "0.07" maxsig = "0.10" alpha = "1000" />  <!--> ex
    <widelane_decision maxdev = "0.25" maxsig = "0.12" alpha = "1000" />  <!--> widelane
    <narrowlane_decision maxdev = "0.35" maxsig = "0.12" alpha = "1000" />  <!--> narrow
</ambiguity>
```

Figure 5.2.5 XML configuration example for the PPP_STEC mode

5.3 Running Example for ROTI Index

The settings for each satellite system are specified in Figures 5.1.3 and 5.1.4.

```xml
<gen>
    <function>ROTI             </function>     <!--> function: CCL_STEC PPP_STEC ROTI AATR  <!-
    <beg> 2023-04-10 00:00:00 </beg>     <!--> begin time <!-->
    <end> 2023-04-10 23:59:30 </end>     <!--> end time <!-->
    <int> 30                  </int>     <!--> sampling interval <!-->
    <sys> GPS                 </sys>     <!--> system ex: GPS GAL GLO BDS <!-->
    <rec> HRAO POTS           </rec>     <!--> site (4-char upper) <!-->
    <est> FLT                 </est>     <!--> Estimator : FLT <!-->
</gen>
```

Figure 5.3.1 XML configuration example for the ROTI mode

```xml
<inputs>
    <rinexo>  <!--> rinex obs file <!-->
        obs\hrao1000.23o
        obs\CORD00ARG_R_20231000000_01D_30S_MO.rnx
        obs\POTS00DEU_R_20231000000_01D_30S_MO.rnx
    </rinexo>
    <rinexn> gnss\brdc1000.23n </rinexn>  <!--> rinex nav file <!-->
    <rinexc> gnss\CODOMGXFIN_20231000000_01D_30S_CLK.CLK </rinexc>  <!--> precise satellite clock offset file <!-->
    <sp3> gnss\CODOMGXFIN_20231000000_01D_05M_ORB.SP3 </sp3>  <!--> precise orbit file <!-->

    <THREE_SP3>  <!--> CCL ROTI AATR precise orbit file <!-->
        gnss\CODOMGXFIN_20230990000_01D_05M_ORB.SP3
        gnss\CODOMGXFIN_20231000000_01D_05M_ORB.SP3
        gnss\CODOMGXFIN_20231010000_01D_05M_ORB.SP3

    </THREE_SP3>
    <de> model\jpleph_de405_great </de>  <!--> Planetary ephemeris file <!-->
    <atx> model\igs20_2290.atx </atx>  <!--> Antenna correction file <!-->
    <blq> model\oceanload </blq>  <!--> oceanload file <!-->
    <eop> model\poleut1 </eop>  <!--> ERP file <!-->
    <bias> gnss\CASOMGXRAP_20231000000_01D_01D_DCB.BSX </bias>  <!--> DCB file <!-->
</inputs>
```

Figure 5.3.2 XML configuration example for the ROTI mode

```xml
<process>
    <roti_window>            10            </roti_window><!--> sliding window size (in epochs)<!-->
    <rot_unit>               min           </rot_unit><!--> unit of ROT (sec/min)<!-->
    <aatr_interval>          60           </aatr_interval><!--> AATR calculation interval (in epochs) <!-->
    <arc_min_length>         10            </arc_min_length><!--> minimum arc length threshold (in epochs)<!-->
    <ion_height>             350000        </ion_height><!--> height of the ionospheric layer (in meters) <!-->
    <mapping_function>       0             </mapping_function><!--> type of mapping function used for VTEC conve
    <minimum_elev>           7             </minimum_elev><!--> cut-off satellite elevation(deg) <!<!-->
```

Figure 5.3.3. XML configuration example for the ROTI mode

5.4 Running Example for AATR Index

The settings for each satellite system are specified in Figures 5.1.3 and 5.1.4.

```xml
<gen>
    <function>AATR              </function>   <!--> function: CCL_STEC PPP_STEC ROTI AATR  <!-->
    <beg> 2023-04-10 00:00:00 </beg>     <!--> begin time <!-->
    <end> 2023-04-10 23:59:30 </end>     <!--> end time <!-->
    <int> 30                  </int>     <!--> sampling interval <!-->
    <sys> GPS                 </sys>     <!--> system ex: GPS GAL GLO BDS <!-->
    <rec> HRAO POTS           </rec>     <!--> site (4-char upper) <!-->
    <est> FLT                 </est>     <!--> Estimator : FLT <!-->
</gen>
```

Figure 5.4.1 XML configuration example for the AATR mode

```xml
<inputs>
    <rinexo>  <!--> rinex obs file <!-->
        obs\hrao1000.23o
        obs\CORD00ARG_R_20231000000_01D_30S_MO.rnx
        obs\POTS00DEU_R_20231000000_01D_30S_MO.rnx
    </rinexo>
    <rinexn> gnss\brdc1000.23n </rinexn>  <!--> rinex nav file <!-->
    <rinexc> gnss\CODOMGXFIN_20231000000_01D_30S_CLK.CLK </rinexc>  <!--> precise satellite clock offset file <!-->
    <sp3> gnss\CODOMGXFIN_20231000000_01D_05M_ORB.SP3 </sp3>  <!--> precise orbit file <!-->

    <THREE_SP3>  <!--> CCL ROTI AATR precise orbit file <!-->
        gnss\CODOMGXFIN_20230990000_01D_05M_ORB.SP3
        gnss\CODOMGXFIN_20231000000_01D_05M_ORB.SP3
        gnss\CODOMGXFIN_20231010000_01D_05M_ORB.SP3

    </THREE_SP3>
    <de> model\jpleph_de405_great </de>  <!--> Planetary ephemeris file <!-->
    <atx> model\igs20_2290.atx </atx>  <!--> Antenna correction file <!-->
    <blq> model\oceanload </blq>  <!--> oceanload file <!-->
    <eop> model\poleut1 </eop>  <!--> ERP file <!-->
    <bias> gnss\CASOMGXRAP_20231000000_01D_01D_DCB.BSX </bias>  <!--> DCB file <!-->
</inputs>
```

Figure 5.4.2 XML configuration example for the AATR mode

```xml
<process>
    <roti_window>          10          </roti_window><!--> sliding window size (in epochs)<!-->
    <rot_unit>             min         </rot_unit><!--> unit of ROT (sec/min)<!-->
    <aatr_interval>        60          </aatr_interval><!--> AATR calculation interval (in epochs) <!-->
    <arc_min_length>       10          </arc_min_length><!--> minimum arc length threshold (in epochs)<!-->
    <ion_height>           350000      </ion_height><!--> height of the ionospheric layer (in meters) <!-->
    <mapping_function>     0           </mapping_function><!--> type of mapping function used for VTEC conve
    <minimum_elev>         7           </minimum_elev><!--> cut-off satellite elevation(deg) <!<!-->
```

Figure 5.4.3 XML configuration example for the AATR mode

5.5 Execution and Results

After configuring the input files and XML, set ***IonoMoni.cpp*** as the startup item and run the program, as shown in Figures 5.5.1 and 5.5.2. The output files will be automatically saved in the result folder, which will be created under the ***./out/build/release-with-xml/*** directory.

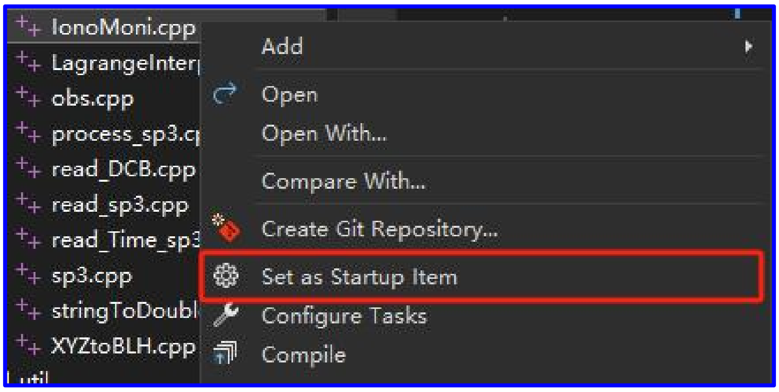

Figure 5.5.1 Set IonoMoni.cpp as the start

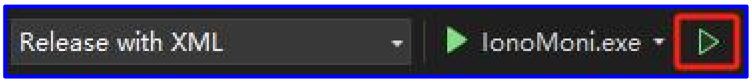

Figure 5.5.2 Run the IonoMoni program

The ROT, ROTI, and STEC results are saved in a matrix format to text files named after the corresponding station and satellite system. Each row represents one epoch, and each column corresponds to one satellite. Figure 5.5.3 shows an example of the ROT, ROTI, and STEC results.

Epoch \ PRN	G01	G02	G03	G04	G05	G06	G07	G08	G09	G10	G11
Epoch 0001:	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Epoch 0002:	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Epoch 0003:	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Epoch 0004:	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Epoch 0005:	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Epoch 0006:	0.00000	170.41060	0.00000	0.00000	0.00000	0.00000	17.18452	37.71762	47.81325	0.00000	0.00000
Epoch 0007:	0.00000	169.21639	0.00000	0.00000	0.00000	0.00000	17.14016	37.98050	47.05335	0.00000	0.00000
Epoch 0008:	0.00000	166.74931	0.00000	0.00000	0.00000	0.00000	17.20619	37.90358	46.77237	0.00000	0.00000
Epoch 0009:	0.00000	165.25314	0.00000	0.00000	0.00000	0.00000	15.73836	37.74839	46.46997	0.00000	0.00000
Epoch 0010:	0.00000	164.09089	0.00000	0.00000	0.00000	0.00000	16.22180	37.96126	47.20903	0.00000	0.00000
Epoch 0011:	0.00000	163.11176	0.00000	0.00000	0.00000	0.00000	16.83380	37.81412	47.71941	0.00000	0.00000
Epoch 0012:	0.00000	162.25150	0.00000	0.00000	0.00000	0.00000	16.69609	37.79785	47.53581	0.00000	0.00000
Epoch 0013:	0.00000	161.66234	0.00000	0.00000	0.00000	0.00000	16.01677	38.03709	47.04900	0.00000	0.00000
Epoch 0014:	0.00000	161.15183	0.00000	0.00000	0.00000	0.00000	15.68226	37.77123	46.67346	0.00000	0.00000
Epoch 0015:	0.00000	160.38521	0.00000	0.00000	0.00000	0.00000	16.50542	37.63392	46.33859	0.00000	0.00000
Epoch 0016:	0.00000	159.12480	0.00000	0.00000	0.00000	0.00000	16.65976	37.50459	46.41475	0.00000	0.00000
Epoch 0017:	0.00000	157.83829	0.00000	0.00000	0.00000	0.00000	16.48456	37.40403	46.30438	0.00000	0.00000
Epoch 0018:	0.00000	157.00238	0.00000	0.00000	0.00000	0.00000	16.25472	37.24393	46.27686	0.00000	0.00000
Epoch 0019:	0.00000	156.12873	0.00000	0.00000	0.00000	0.00000	16.07373	37.04107	46.05675	0.00000	0.00000
Epoch 0020:	0.00000	155.15940	0.00000	0.00000	0.00000	0.00000	15.93329	36.92658	45.88945	0.00000	0.00000
Epoch 0021:	0.00000	154.18936	0.00000	0.00000	0.00000	0.00000	16.05841	36.82520	45.68417	0.00000	0.00000
Epoch 0022:	0.00000	153.02225	0.00000	0.00000	0.00000	0.00000	16.24846	36.68029	45.68116	0.00000	0.00000
Epoch 0023:	0.00000	151.87403	0.00000	0.00000	0.00000	0.00000	16.34929	36.52656	45.74338	0.00000	0.00000
Epoch 0024:	0.00000	151.00554	0.00000	0.00000	0.00000	0.00000	16.36874	36.32905	45.60834	0.00000	0.00000
Epoch 0025:	0.00000	150.17355	0.00000	0.00000	0.00000	0.00000	16.53409	35.93875	45.31820	0.00000	0.00000
Epoch 0026:	0.00000	149.45661	0.00000	0.00000	0.00000	0.00000	16.53821	35.74039	45.20635	0.00000	0.00000
Epoch 0027:	0.00000	148.76614	0.00000	0.00000	0.00000	0.00000	16.87359	35.49386	44.96444	0.00000	0.00000
Epoch 0028:	0.00000	147.95507	0.00000	0.00000	0.00000	0.00000	16.94641	35.17642	44.91592	0.00000	0.00000

Figure 5.5.3 Example of output format for ROT, ROTI, and STEC results

The AATR results are saved in a list format to text files named after the corresponding station and satellite system. Each row represents one time block, and the number of epochs spanned by each block depends on the settings in the XML file.Figure 5.5.4 shows an example of the AATR results.

```
Block          AATR
Block 01:    0.41920
Block 02:    0.25623
Block 03:    0.25531
Block 04:    0.17505
Block 05:    0.20425
Block 06:    0.22489
Block 07:    0.19512
Block 08:    0.17170
Block 09:    0.17960
Block 10:    0.17544
Block 11:    0.32271
Block 12:    0.24659
Block 13:    0.12649
Block 14:    0.20507
Block 15:    0.25380
Block 16:    0.16091
Block 17:    0.17800
Block 18:    0.21411
Block 19:    0.11007
Block 20:    0.06791
Block 21:    0.07264
Block 22:    0.04325
Block 23:    0.13730
Block 24:    0.17746
```

Figure 5.5.4 Example of output format for AATR results

6. Visualization

The Python plotting scripts provided with IonoMoni support multi-station plotting for a single day. The directory structure under the plot folder is as follows:

Table 6.1 Plot Folder Structure

plot	
./data_aatr	Input folder for AATR plot data
./data_ccl	Input folder for CCL_STEC plot data
./data_ppp	Input folder for PPP_STEC plot data
./data_roti	Input folder for ROTI plot data
./output	Output folder for plot results
./PythonScripts	Plot script folder

6.1CCL_STEC and PPP_STEC Results Plotting

To plot the STEC results, the text output files generated by IonoMoni in either CCL_STEC or PPP_STEC mode must be used as input data.　Fanyi1 These files should be placed in the ***./data_ccl*** or ***./data_ppp*** folder.

The following takes the processing of CCL_STEC mode results as an example:

Open the ***ccl_stec_plot.py*** script located in the ***./PythonScripts*** folder, and modify the parameters in the main function as shown in Figure 6.1.1. The ***site_list*** should include all stations to be plotted. The ***year*** and ***doy*** should correspond to the year and day of year of the input data, and site_list should include all station names corresponding to the input data in the ***./data_ccl*** folder.

```python
# ======== Main entry point ========
if __name__ == "__main__":
    site_list = ["HKWS", "ABPO"]
    input_path = "../data_ccl"          # Relative input path
    output_path = "../output"           # Relative output path
    year = 2023
    doy = 330

    batch_plot(site_list, input_path, output_path, year, doy)
```

Figure 6.1.1 Example of the main function in ccl_stec_plot.py

After running the script, the generated plotting results will be saved in PNG format to the ***./output/ccl_stec_plot/<year>_<doy>/<site>*** directory.

6.2VTEC Map Plotting

To plot the VTEC map, the text output files generated by IonoMoni in either CCL_STEC or PPP_STEC mode must be used as input data. These files should be placed in the ***./data_vtec*** folder. (The data for one station on a single day should include text output files for IPP longitude and latitude, as well as a text output file for VTEC.).

Open the ***vtec_map_plot.py*** script located in the ***./PythonScripts*** folder, and modify the parameters in the main function as shown in Figure 6.2.1. The ***site_list*** should include all stations to be plotted.The system selected should match the input text files (GPS, BDS, GAL, GLO). The ***year*** and ***doy*** should correspond to the year and day of year of the input data, and site_list should include all station names corresponding to the input data in the ***./data_vtec*** folder.

```python
# ======== Main entry point ========
if __name__ == "__main__":
    input_folder = r"../data_vtec"  # Input data directory
    output_base = r"../output"  # Output root directory
    sitelists = ["ABPO"]  # Station list
    year = 2023
    doy = 330
    system = "GPS"
    plot_multi_station_vtec(
        input_folder, sitelists, system,
        output_base=output_base, year=year, doy=doy
    )
```

Figure 6.2.1 Example of the main function in vtec_map_plot.py

After running the script, the generated plotting results will be saved in PNG format to the ***./output/vtec_map_plot/<year>_<doy>/***directory.The output file is named following the convention ***VTEC_<system>_map_lon<lon_min><E/W>-<lon_max><E/W>_lat<lat_min><N/S>-<lat_max><N/S>.png***, where ***lon_min*** and ***lon_max*** represent the minimum and maximum longitudes with ***E*** or ***W*** denoting east or west, and ***lat_min*** and ***lat_max*** represent the minimum and maximum latitudes with ***N*** or ***S*** denoting north or south.

6.3ROTI and AATR Results Plotting

To plot the results of ROTI or AATR, the text output files generated by IonoMoni in ROTI or AATR mode must be used as input data and placed in the ***./data_roti*** or ***./data_aatr*** folder, respectively.

The following takes the processing of ROTI mode results as an example:

Open the ***roti_plot.py*** script located in the ***./PythonScripts*** folder, and modify the parameters in the main function as shown in Figure 6.3.1. The ***year*** and ***doy*** should match the year and day of year of the input data, ***rot_unit*** should be consistent with the unit setting used in the XML configuration when the input data was generated, and ***site_list*** should include all station names corresponding to the input data in the ***./data_ccl*** folder.

```
if __name__ == "__main__":
    base_input_path = "../data_roti"
    base_output_path = "../output"
    year = 2023
    doy = 330
    rot_unit = "min"   # "sec" means ROT unit is TECU/sec; "min" means TECU/min
    site_list = ["HKWS", "ABPO"]  # Add your station names here

    batch_plot_roti_rot(base_input_path, base_output_path, year, doy, rot_unit, site_list)
```

Figure 6.3.1 Example of the main function in roti_plot.py

After running the script, the generated plotting results will be saved in PNG format to the ***./output/roti_plot/<year>_<doy>/<site>*** directory.

6.4Example of Plotting Results

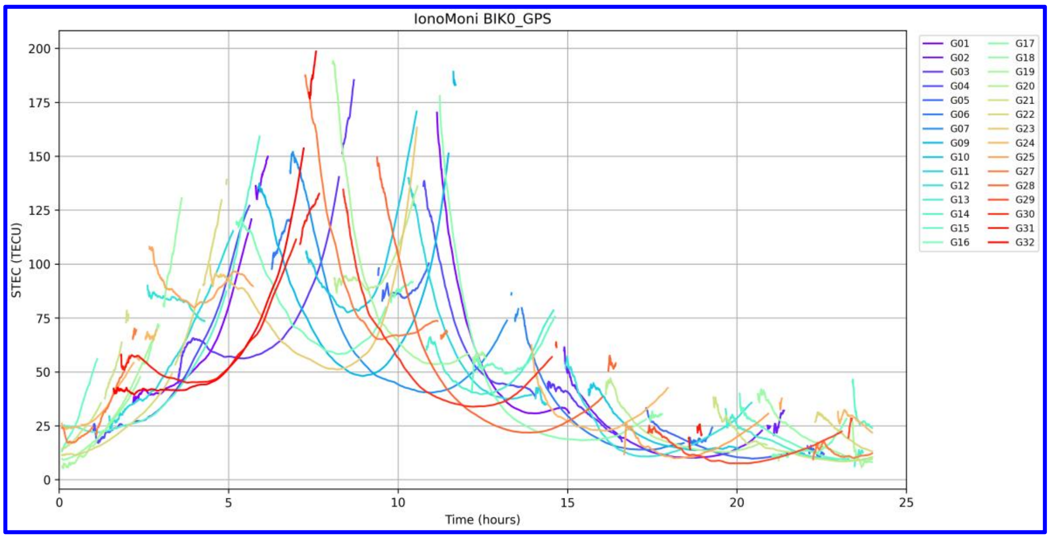

Figure 6.4.1 Plotting Example Based on CCL_STEC Results

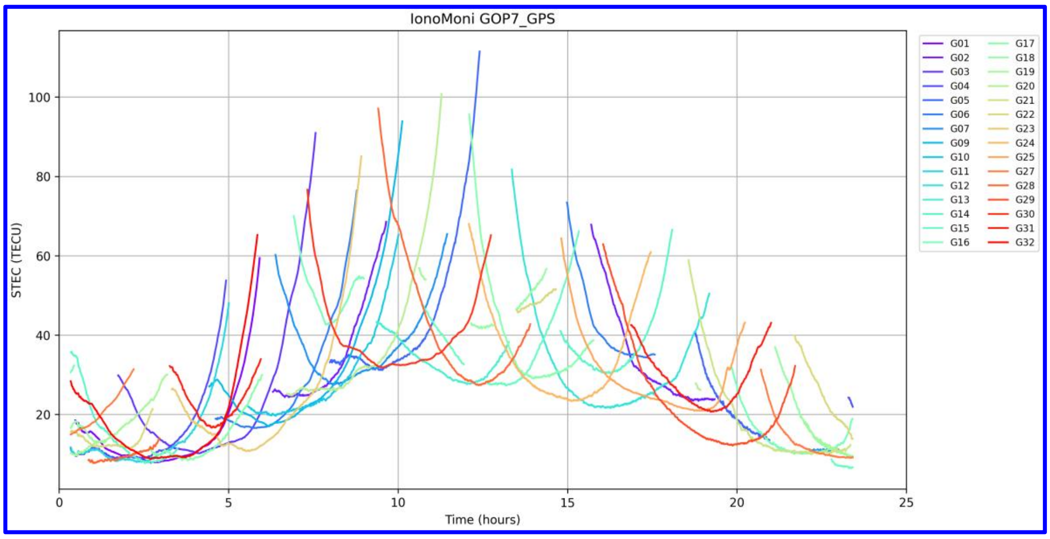

Figure 6.4.2 Plotting Example Based on PPP_STEC Results

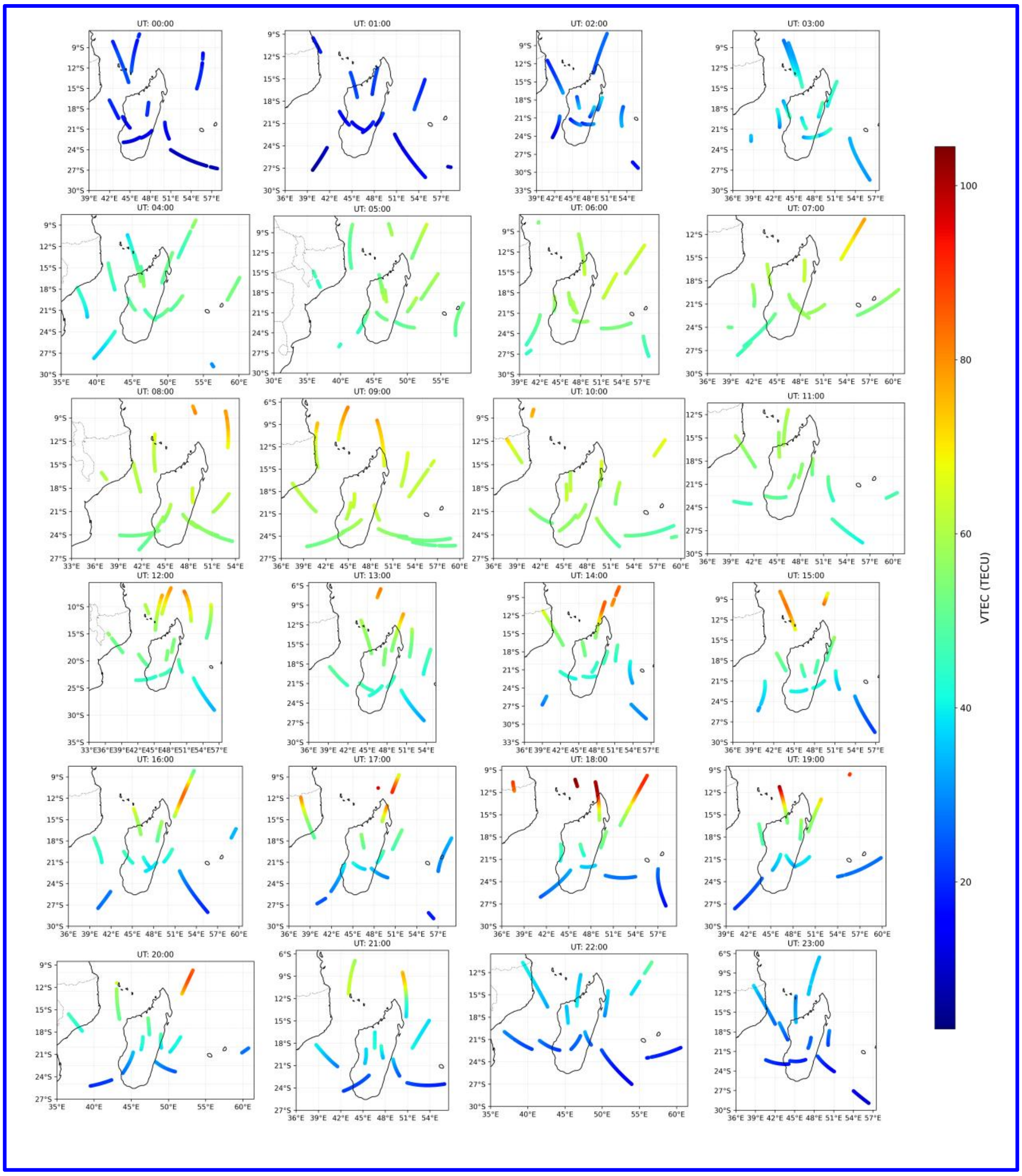

Figure 6.4.3 VTEC Map Plotting Example Based on CCL_STEC Results

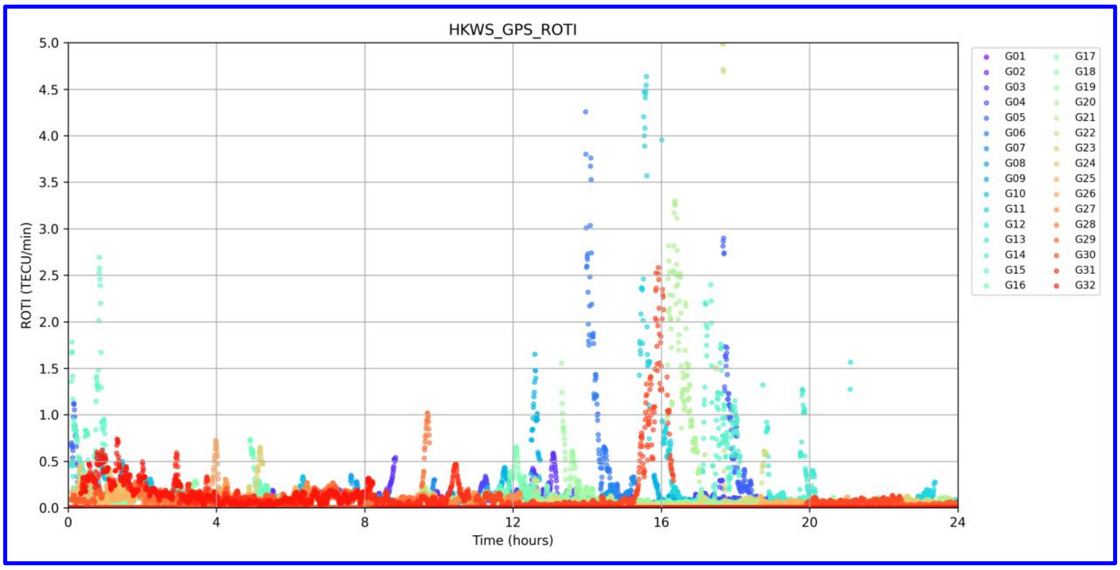

Figure 6.3.3 Plotting Example Based on ROTI Results

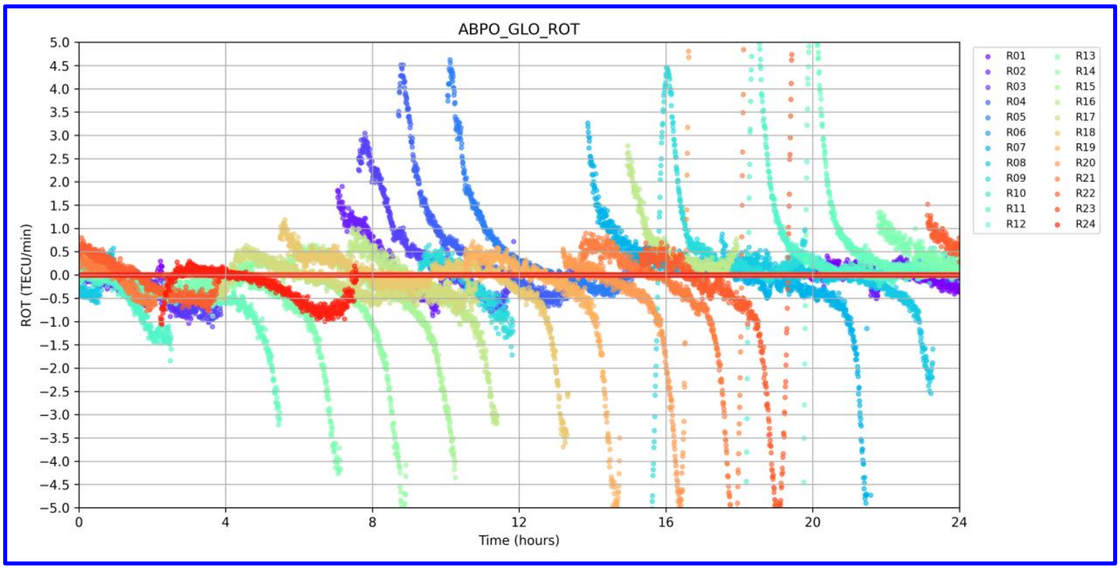

Figure 6.3.4 Plotting Example Based on ROT Results

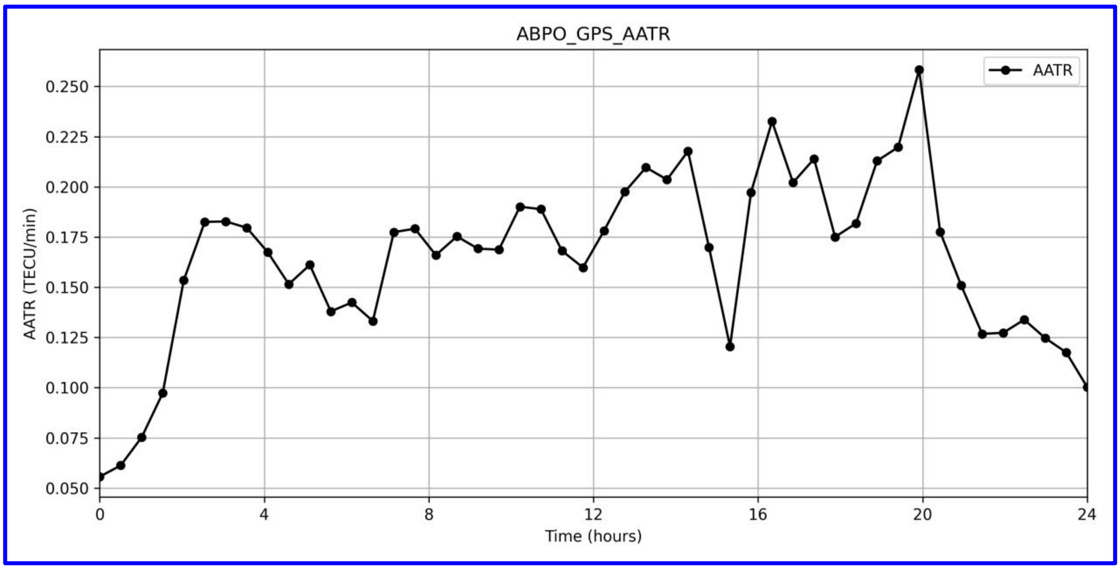

Figure 6.3.4 Plotting Example Based on AATR Results

7. Batch Processing

The Python batch processing scripts provided with IonoMoni support multi-day and multi-station data processing. The directory structure under the batch_process folder is as follows:

Table 7.1 Directory Structure of the batch_process Folder

batch_process	
./data	Input data folder
./ini	INI configuration folder
./project	Output results folder
./PythonScripts	Python batch processing scripts folder
./sitelist	Station list folder
./xml	XML folder

Descriptions of the scripts in the PythonScripts folder are as follows:

Table 7.2 Directory Structure of the batch_process Folder

Python Script	Description
run_cmd_IonoMoni_multi.py	Main script for IonoMoni batch processing
gnss_sitelist_io.py	Provides a general interface for reading GNSS station files
gnss_ini_tool.py	Manages the data pool and reads input data
gnss_print_tool.py	Outputs key information to the terminal
gnss_run_tool.py	Provides command-line operation and execution functions
gnss_timestran_tool.py	Handles time format conversion and calculation in processing

The batch processing Python scripts of IonoMoni use INI files to configure corresponding XML files for each station on each day, enabling automated data processing. The ***run_IonoMoni.ini*** configuration file is described as follows:

Table 7.3 Directory Structure of the batch_process Folder

Configuration Node	Description
function	CCL_STEC PPP_STEC ROTI AATR
beg_time	Start time <YYYY DOY>
end_time	End time <YYYY DOY>
int_time	Calculation interval (in days)
site_path	Path to the station list file
project_path	Path to the output result files
IonoMoni	Path to the IonoMoni executable program
IonoMoni_xml	Path to the batch processing template XML file
rinexo_name	Observation file name, e.g., <SITE><DOY>0.<YY>o
rinexc_name	Precise clock offset file name, e.g., COD0MGXFIN_<YYYY><DOY>0000_01D_30S_CLK.CLK
rinexn_name	Broadcast ephemeris file name, e.g., brdc<DOY>0.<YY>p
bia_name	DCB file name, e.g., CAS0MGXRAP_<YYYY><DOY>0000_01D_01D_DCB.BSX
sp3_name	Precise ephemeris file name, e.g., COD0MGXFIN_<YYYY><DOY>0000_01D_05M_ORB.SP3
three_sp3_name	3Days Precise ephemeris file name, e.g., COD0MGXFIN_<YYYY><DOY>0000_01D_05M_ORB.SP3
ifcb_name	IFCB file name, e.g., ifcb_<YYYY><DOY>
upd_name	UPD file name, e.g., upd_wl_<YYYY><DOY>_GREC upd_nl_<YYYY><DOY>_GREC
systerm_de	Planetary ephemeris file, e.g., jpleph_de405_great
systerm_atx	Antenna file, e.g., igs20_2290.atx
systerm_blq	Ocean tide loading file, e.g., oceanload
systerm_eop	Earth orientation parameters file, e.g., poleut1
systerm_path	Path to system files
rinexo_path	Path to observation files
rinexc_path	Path to precise clock offset files
rinexn_path	Path to broadcast ephemeris files
bia_path	Path to differential code bias files
sp3_path	Path to precise ephemeris files
three_sp3_path	Path to 3days precise ephemeris files
upd_path	Path to uncalibrated phase delay files

After preparing the data in the ./data folder, configure the run_IonoMoni.ini file as shown in Figure 7.3. To perform multi-day data processing, users need to enter the command:

>> cd /d <install_dir>\IonoMoni\batch_process\PythonScripts

>> python run_cmd_IonoMoni_multi.py ^

-year <YYYY> -beg <DOY_START> -end <DOY_END> ^

-ini ..\ini\run_IonoMoni.ini

After the batch processing is completed, a corresponding output directory will be automatically created, and the batch processing results will be saved in the ./project/<function>/<YYYY>_<DOY> folder, which contains the daily XML configuration files and text result files (PPP_STEC mode will additionally include FLT result files).

```
[IonoMoni]
# function
function = PPP_STEC
# begin time
beg_time = 2023 305
# end time
end_time = 2023 306
# interval (day)
int_time = 1
# site list
site_path = ../sitelist/site_list.txt
# output path
project_path = ../project/PPP_STEC
# program path
IonoMoni = ../../out/build/release-with-xml/IonoMoni.exe
# template xml
IonoMoni_xml = ../xml/IonoMoni.xml

[data_pool]
# input files and path
systerm_de = jpleph_de405_great
systerm_atx = igs20_2290.atx
systerm_blq = oceanload
systerm_eop = poleut1
systerm_path = ../data/sys
rinexo_name = <SITE><DOY>0.<YY>o
rinexc_name = CODOMGXFIN_<YYYY><DOY>0000_01D_30S_CLK.CLK
rinexn_name = brdc<DOY>0.<YY>p
bia_name = CAS0MGXRAP_<YYYY><DOY>0000_01D_01D_DCB.BSX
sp3_name = CODOMGXFIN_<YYYY><DOY>0000_01D_05M_ORB.SP3
three_sp3_name = CODOMGXFIN_<YYYY><DOY>0000_01D_05M_ORB.SP3
ifcb_name = ifcb_<YYYY><DOY>
#The Configuration Node is retained but not used in computation.
upd_name = upd_wl_<YYYY><DOY>_GREC~upd_nl_<YYYY><DOY>_GREC
rinexo_path = ../data/obs/<YYYY>/<DOY>
rinexc_path = ../data/clk
rinexn_path = ../data/rinexn
bia_path = ../data/bia
sp3_path = ../data/sp3
three_sp3_path = ../data/sp3
ifcb_path = ../data/ifcb
#The Configuration Node is retained but not used in computation.
upd_path = ../data/upd
```

Table 7.3 Directory Structure of the batch_process Folder

8. Data preparation

Different modes of IonoMoni require different input files.

For **CCL_STEC** mode, the required input files include: observation data, precise orbit data, and Differential Code Bias (DCB) files.

For **PPP_STEC** mode, the required input files include: observation data, broadcast ephemeris, precise orbit data, precise clock products, Differential Code Bias (DCB), IGS antenna files, planetary ephemeris files, ocean tide loading files, and Earth Orientation Parameters (EOP) files.

For **ROTI** and **AATR** modes, the required input files include: observation data and precise orbit data.

The detailed format descriptions and acquisition methods for these input files are listed in Table 8.1.

Table 8.1 Format Descriptions and Acquisition Methods for Input Files

Input files	File Description	Format Description and Acquisition Methods
RINEXO	GNSS Observations	https://files.igs.org/pub/data/format/rinex304.pdf https://cddis.nasa.gov/archive/gnss/data/daily/
RINEXN	Broadcast Ephemeris	https://files.igs.org/pub/data/format/rinex304.pdf https://cddis.nasa.gov/archive/gnss/data/daily/2024/brdc/
SP3	Precise Orbit	https://files.igs.org/pub/data/format/sp3d.pdf https://cddis.nasa.gov/archive/gnss/products/
RINEXC	Precise Clock Products	https://files.igs.org/pub/data/format/rinex_clock304.txt https://cddis.nasa.gov/archive/gnss/products/
DCB	Differential Code Bias	https://files.igs.org/pub/data/format/sinex_bias_100.pdf https://cddis.nasa.gov/archive/gnss/products/bias/
jpleph_de405	Planetary Ephemeris	https://ssd.jpl.nasa.gov/planets/eph_export.html https://ssd.jpl.nasa.gov/ftp/eph/planets/Linux/
oceanload	Ocean Tide	http://holt.oso.chalmers.se/loading/example_blq.html http://holt.oso.chalmers.se/loading
poleut1	Earth Orientation Parameters	Refer to the GREAT-PVT_manual_1.0.pdf located in the ./poleut1 folder under the IonoMoni root directory.
atx	Antenna	https://files.igs.org/pub/station/general/antex14.txt https://files.igs.org/pub/station/general/pcv_archive/