

User Manual 4.14 Measure and Filtering

De Wiki

Aller à : [navigation](#), [rechercher](#)

[User Manual 4.14 Measure and Filtering](#)

Sommaire

- [1 Introduction](#)
 - [1.1 Scope](#)
 - [1.2 Javadoc](#)
 - [1.3 Links](#)
 - [1.4 Useful Documents](#)
 - [1.5 Package Overview](#)
- [2 Features Description](#)
 - [2.1 Signal Propagation](#)
 - [2.2 Tropospheric correction](#)
 - [2.3 Ionospheric correction](#)
 - [2.3.1 Bent ionospheric model](#)
 - [2.3.1.1 What it does](#)
 - [2.3.1.2 What it does not](#)
 - [2.3.1.3 About thread-safety](#)
- [3 Getting Started](#)
 - [3.1 Signal propagation](#)
 - [3.2 Tropospheric correction](#)
 - [3.3 Ionospheric correction](#)
- [4 Contents](#)
 - [4.1 Interfaces](#)
 - [4.2 Classes](#)

Introduction

Scope

This section describes all the building blocks available for measure functions computation and measures management.

For now are available in the "signal propagation" package the tools to compute a signal propagation object (emission and reception dates, propagation vector, signal derivatives) and modify it (tropospheric effects correction, ionospheric effects correction...)

Javadoc

The packages associated to the "measure and filtering" theme are :

Library

Orekit [Package fr.cnes.sirius.patrius.models.earth](#)

Patrius [Package fr.cnes.sirius.patrius.signalpropagation](#)

Javadoc

Patrius [Package fr.cnes.sirius.patrius.signalpropagation.iono](http://package.fr.cnes.sirius.patrius.signalpropagation.iono)

Links

None as of now.

Useful Documents

- Bent model

<http://modelweb.gsfc.nasa.gov/ionos/bent.html>

Package Overview

Signal propagation

The Signal Propagation package contains the classes for signal propagation computation :

- VacuumSignalPropagation, that represents the signal itself (vector, emission date, reception date, emitter & receiver velocities) and its derivatives.
- VacuumSignalPropagationModel, a toolbox that provides methods to compute the signal propagation (first the initial commutation in the void, and then potentially apply to it some correction).

It also contains the classes for signal corrections: the correction models (AzoulayModel and FixedDelayModel) and their interfaces TroposphericCorrection and AngularCorrection.

Iono

For now, the Iono package contains the generic interface for ionospheric corrections (IonosphericCorrection) and the classes relevant to computing the electronic content of the ionosphere, according to the Bent ionosphere model :

- BentModel, the Bent model for the electronic content used in ionospheric corrections.
- R12Loader, data loader for the CCIR12 data file.
- USKLoader, data loader for the USK data file ("NEWUSK" type).

Features Description

Signal Propagation

The signal propagation model is a toolbox.

Its main method "computeSignalPropagation" is able to compute and create a propagation object (propagation vector, emission and reception dates, emitter & receiver velocities) out of the two PVCoordinatesProvider, that represent the emitter and the receiver, and one of the two dates (emission or reception).

It also provides method to get the corrections to be applied to this signal, like the tropospheric effects correction (see next paragraph).

Tropospheric correction

Interface

The tropospheric correction interface (TroposphericCorrection) returns the signal correction (in seconds) due to tropospheric effects to be applied to a given signal propagation vector.

NB : to get directly the correction associated to a VacuumSignalPropagation object, a method is available in the VacuumSignalPropagationModel, using any TroposphericCorrection.

Azoulay model

The Azoulay model is a particular model of tropospheric correction, under the TroposphericCorrection and AngularCorrection interfaces.

For its computations are also needed the local pressure, temperature, moisture and geodetic altitude of the involved station.

Ionospheric correction

Currently, the provided ionospheric correction capabilities are :

- an implementation of the Bent ionospheric model (electronic content computation)

Other are scheduled to appear eventually.

Bent ionospheric model

What it does

The Bent ionospheric model implementation in Patrius is issued from the existing MSLIB Fortran implementation.

It computes the electronic content for the signal propagated between a ground station and the satellite.

The Bent model needs for initialization :

- a [SolarActivityDataProvider](#)
- a R12Provider : sunspots data (usually from a file)
- a USKProvider : custom data for the Bent Model (usually from a file)
- the BodyShape representing the Earth
- the coordinates of the station position and the associated frame

The R12Provider and USKProvider implementations are part of the package; they use the Patrius data loading features.

What it does not

The Bent model implementation is strictly for the computation of the electronic content. The following is missing :

- taking into account the direction of the signal
- taking into account multiple paths for the signal
- taking into account the position and date corrections for the movement during the signal propagation

All these issues will have to be addressed before the model is really usable, i.e. used in a signal propagation model like the tropospheric corrections already are.

About thread-safety

The BentModel class is thread-safe, so a same instance of this class can now be shared between threads in a multi-threaded environment.

Getting Started

Signal propagation

Here is a code sample of the computation of a signal propagation.

```
// the transmitter (here, the station) is supposed to be a well defined
PVCoordinatesProvider
PVCoordinatesProvider station;
// the receiver (here, the spacecraft) is supposed to be a well defined
PVCoordinatesProvider
PVCoordinatesProvider spacecraft;

// the emission date is supposed to be known
AbsoluteDate emissionDate;

// creation of the signal propagation model
// Reference frame : must be an inertial frame
Frame refFrame = FramesFactory.getGCRF();
// signal propagation computation threshold (see the javadoc for details)
double threshold = 1.0e-13;
// model
VacuumSignalPropagationModel model = new
VacuumSignalPropagationModel(refFrame, threshold);

// computation of the signal propagation
// the FixedDate informs the method that the given date is the EMISSION date
VacuumSignalPropagation goSignal = model.computeSignalPropagation(
    station, spacecraft, emissionDate, FixedDate.EMISSION);
```

Tropospheric correction

Here is a code sample for how to get the correction associated to a given VacuumSignalPropagation object using the Azoulay model.

```
// Topocentric frame of the station
// NB : this object is a PVCoordinatesProvider and can be used to compute the
VacuumSignalPropagation
// object, cf. previous paragraph
TopocentricFrame stationTopoFrame = new TopocentricFrame(earthShape,
stationPosition, "station");
```

```
// tropospheric correction model creation
TroposphericCorrection correctionModel = new AzoulayModel(pressure,
temperature, moisture, stationAltitude);

// Computation of the corrections directly from the signal object
(VacuumSignalPropagation class)
double corrections = model.getSignalTropoCorrection(correctionModel, signal,
stationTopoFrame);
```

Ionospheric correction

Here is a code sample on how to use the Bent ionospheric model.

```
// Input needed for the Bent model
// 1) solar activity
final SolarActivityDataProvider cstSolarActivity = new
ConstantSolarActivity(84., 15);
// 2) R12 file reader (sunspots data)
final R12Provider r12Prov = new R12Loader("CCIR12");
// 3) USK file reader (Bent model parameters)
final USKProvider uskProv = new USKLoader("NEWUSK");
// 4) earth shape
final CelestialBodyFrame earthFrame = FramesFactory.getITRF();
final BodyShape earth = new OneAxisEllipsoid(6378137, 1. / 298.257,
earthFrame);

// We can then create a Bent model instance
final IonosphericCorrection ionoBent = new BentModel(r12Prov,
cstSolarActivity,
uskProv, earth, stationPosition, stationFrame, frequency);
// and call it with parameters :
// satellite position and frame,
// date
final double delay = ionoBent.computeSignalDelay(date,
satellitePosition, satelliteFrame);
```

For those knowing the MSLIB version of this model : the computed value is the TOTNA variable in the original Fortran code.

Contents

Interfaces

Interface	Summary	Javadoc
TroposphericCorrection	Interface for all the signal propagation corrections due to the troposphere.	...

IonosphericCorrection	Interface for all the signal propagation corrections due to the ionosphere.	...
AngularCorrection	Interface representing an angular correction (on the satellite elevation).	...
R12Provider	R12 value provider for the Bent model.	...
USKProvider	Interface for the providers of USK data for the Bent ionospheric correction.	...

Classes

Class	Summary	Javadoc
VacuumSignalPropagation	A propagation vector and its associated dates.	...
VacuumSignalPropagationModel	Toolbox to compute SignalPropagation objects and correct it with different models.	...
AzoulayModel	Azoulay tropospheric correction model.	...
BentModel	Bent model for the electronic content used in ionospheric corrections.	...
R12Loader	Data loader for the R12 values.	...
USKLoader	Reader for the USK data file (file of the "NEWUSK" type).	...
SaastamoinenModel	Saastamoinen model for tropospheric corrections.	...
FixedDelayModel	Fixed delay tropospheric correction model.	...

Récupérée de

« http://patrius.cnes.fr/index.php?title=User_Manual_4.14_Measure_and_Filtering&oldid=3775 »
 Catégorie :

- [User Manual 4.14 Orbit Propagation](#)

Menu de navigation

Outils personnels

- [3.147.51.16](#)
- [Discussion avec cette adresse IP](#)
- [Créer un compte](#)
- [Se connecter](#)

Espaces de noms

- [Page](#)
- [Discussion](#)

Variantes

Affichages

- [Lire](#)
- [Voir le texte source](#)
- [Historique](#)
- [Exporter en PDF](#)

Plus

Rechercher

PATRIUS

- [Welcome](#)

Evolutions

- [Main differences between V4.14 and V4.13](#)
- [Main differences between V4.13 and V4.12](#)
- [Main differences between V4.12 and V4.11](#)
- [Main differences between V4.11 and V4.10](#)
- [Main differences between V4.10 and V4.9](#)
- [Main differences between V4.9 and V4.8](#)
- [Main differences between V4.8 and V4.7](#)
- [Main differences between V4.7 and V4.6.1](#)
- [Main differences between V4.6.1 and V4.5.1](#)
- [Main differences between V4.5.1 and V4.4](#)
- [Main differences between V4.4 and V4.3](#)
- [Main differences between V4.3 and V4.2](#)
- [Main differences between V4.2 and V4.1.1](#)
- [Main differences between V4.1.1 and V4.1](#)
- [Main differences between V4.1 and V4.0](#)
- [Main differences between V4.0 and V3.4.1](#)

User Manual

- [User Manual 4.14](#)
- [User Manual 4.13](#)
- [User Manual 4.12](#)
- [User Manual 4.11](#)
- [User Manual 4.10](#)
- [User Manual 4.9](#)

- [User Manual 4.8](#)
- [User Manual 4.7](#)
- [User Manual 4.6.1](#)
- [User Manual 4.5.1](#)
- [User Manual 4.4](#)
- [User Manual 4.3](#)
- [User Manual 4.2](#)
- [User Manual 4.1](#)
- [User Manual 4.0](#)
- [User Manual 3.4.1](#)
- [User Manual 3.3](#)

Tutorials

- [Tutorials 4.14](#)
- [Tutorials 4.13.5](#)
- [Tutorials 4.12.1](#)
- [Tutorials 4.8.1](#)
- [Tutorials 4.5.1](#)
- [Tutorials 4.4](#)
- [Tutorials 4.1](#)
- [Tutorials 4.0](#)

Links

- [CNES freeware server](#)

Navigation

- [Accueil](#)
- [Modifications récentes](#)
- [Page au hasard](#)
- [Aide](#)

Outils

- [Pages liées](#)
- [Suivi des pages liées](#)
- [Pages spéciales](#)
- [Adresse de cette version](#)
- [Information sur la page](#)
- [Citer cette page](#)

• Dernière modification de cette page le 5 septembre 2024 à 15:31.

- [Politique de confidentialité](#)
- [À propos de Wiki](#)

- [Avertissements](#)

- 