User Manual 4.14 Directions

[User Manual 4.14 Directions]()

Introduction

Scope

The "direction" objects are used for attitude computation purposes to describe different types of axis in space that can evolve in time. A direction provides a vector in space at any date in a given frame. A direction can be for example "spacecraft-moon" or "nadir direction of the spacecraft". An attitude law can then be defined using two of those directions.

All available directions belong to the following categories:

- Defined by a "target" moving point, and a given origin point;

- Defined by a vector only known to be constant in time in a particular frame, or defined by a physical property.

The available features are listed in the description of the general `IDirection` interface or of the more specific `ITargetDirection` interface (for directions pointing a target).

Javadoc

The directions are available in the package `fr.cnes.sirius.patrius.attitudes.directions` of the PATRIUS library.

Library	Javadoc
Patrius	[Package fr.cnes.sirius.patrius.attitudes.directions]()

Links

None as of now.

Useful Documents

None as of now.

Package Overview

The following diagram represents the two main interfaces of the "directions" package, and for each interface some of the implemented directions.

Please note that not all implementations are present in the following diagram for the sake of clarity.

Features Description

Directions

The available directions are:

Basic direction

This direction is defined at any date by a vector; this vector can be constant in a frame, or it can be defined by a physical property (for instance the axis of the poles of a celestial body, the orbit momentum, the velocity vector…).

The `getVector` and `getLine` methods of this interface use a `PVCoordinatesProvider` to represent the origin of the direction, but for some of them this input parameter is unused (for instance for the poles of a celestial body direction). See each javadoc for the associated behavior.

All directions can also provide the line containing the given origin and directed by the vector, expressed in any frame of the tree.

Ground velocity direction

This direction is used for instance when it comes to take pictures of the ground. To avoid distortion, the satellite has to compensate the earth rotation around its yaw axis. On the following drawing, the red direction is the one that has to be followed to compensate the earth rotation.

Target direction

This direction also implements the methods of basic directions. It is defined at any date by the position of two points of space: the target (that defines the direction) and the origin given by the user. The associated vector (and line) is computed from the origin to the target.

Aberration corrections

The available aberration corrections are:

- NONE: no aberration correction is applied.
- LIGHT_TIME: the light-time aberration correction is applied. This aberration correction takes into account the apparent displacement of the source (from the receiver point of view) or of the observer (from the transmitter point of view) during the time of propagation.
- STELLAR: the stellar aberration correction is applied. This aberration correction takes into account the observer velocity at reception time (from the receiver point of view) or the source velocity at transmission time (from the transmitter point of view).
- ALL: both the light-time and the stellar aberration corrections are applied.

Getting Started

Use

For all directions

All the directions implement the "IDirection" interface and so provide the "getVector" method. This method needs a date and a frame to express the vector, and the PV coordinates of the origin point (unused for some of them). It shall be used this way :

```
Vector3D directionVector = myDirection.getVector(pvCoordinatesProvider, date,
frame);
```

They also provide the "getLine" method, with the same arguments.

```
Line directionLine = myDirection.getLine(pvCoordinatesProvider, date, frame);
```

NB : when the PVCoordinatesProvider is null then the origin of the direction is tacitly the frame origin.

For target directions

The directions that implement the "ITargetDirection" interface also provide the "getTarget" method :

```
PVCoordinates directionTarget =
myTargetDirection.getTargetPVCoordinates(date, frame);
```

Contents

Interfaces

Interface	Summary	Javadoc
IDirection	General interface for all directions classes. All basic directions implement it.	...
ITargetDirection	This interface extends IDirection for the particular case of directions defined by a target point. The vector's direction is oriented from the given origin to the target.	...

Classes

Class	Summary	Javadoc
ConstantVectorDirection	Vector constant in one of the frames of the tree.	...
VelocityDirection	Velocity of the origin point expressed in a reference frame, projected in the given one.	...
MomentumDirection	Momentum : normal vector to the trajectory plane of the origin point around a celestial body.	...
CelestialBodyPolesAxisDirection	Axis of the poles of a celestial body.	...
ToCelestialBodyCenterDirection	Vector from the origin to the center of a celestial body.	...
EarthCenterDirection	Vector from the origin to the center of the Earth central body.	...

GenericTargetDirection	Defined by a PVCoordinatesProvider target.	...
BodyPointTargetDirection	Defined by target which is a GeodeticPoint on a BodyShape.	...
GroundVelocityDirection	Defined by the location of the ground target point given by the pointing direction of the satellite and the body shape.	...
NadirDirection	Defined by the location of the nadir point and the satellite position.	...
CrossProductDirection	Cross Product of two directions.	...
GlintApproximatePointingDirection	Direction to the Sun specular reflexion point.	...
EarthToCelestialBodyCenterDirection	Direction from Earth to the celestial body center.	...

Récupérée de « http://patrius.cnes.fr/index.php?title=User_Manual_4.14_Directions&oldid=3743 »

Catégorie :

Menu de navigation

Outils personnels

Espaces de noms

Variantes

Affichages

Plus

Rechercher

PATRIUS

- [Welcome](#)

Evolutions

- [Main differences between V4.15 and V4.14](#)
- [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 4.15](#)
- [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 4.15](#)
* [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 14:43.