User Manual 4.15 Events: sensors

De Wiki

Aller à :

User Manual 4.15 Events: sensors

- Exemple.jpgDescription 1
- Exemple.jpgDescription 2

Sommaire

Introduction

Scope

Here are presented all the events detectors of the theme "sensors".

Javadoc

Those event detectors are available in the packages :

Library	Javadoc
Orekit	[[[:Modèle:JavaDoc4.15]]/fr/cnes/sirius/patrius/propagation/events/package-summary.html Package fr.cnes.sirius.patrius.propagation.event]
Patrius	[[[:Modèle:JavaDoc4.15]]/fr/cnes/sirius/patrius/events/package-summary.html Package fr.cnes.sirius.patrius.events]

Links

None as of now.

Useful Documents

None as of now.

Package Overview

None as of now.

Features Description

g functions

This section describes the meaning of the g switching function for the "sensors" event detectors :

CircularFieldOfViewDetector

The g switching function for the circular field of view is : $g=\alpha_{half aperture}-\alpha_{_{center-targetPosSat}}$.
$\vec{targetPosSat}$ is the satellite-target body vector in the satellite frame; $\alpha_{half aperture}$ is the fov half aperture angle and $\alpha_{center-targetPosSat}$ is the absolute value of the angle between target direction and field of view center.

- $g \lt 0 :$ the target body is outside the field of view:

- $g \gt 0 :$ the target body is inside the field of view:

CentralBodyMaskCircularFOVDetector

This detector is a combination of the previous circular field of view detector and the [MIS_ORB_Home#HPartialandTotaleclipsebyaspheroid eclipse detector] for an ellipsoid body.
It detects the visibility only when the target is not eclipsed by the central body.
The g switching function is :
$g=min(circularFieldOfView, ellipsoidEclipse)$.

DihedralFieldOfViewDetector

The g switching function computes the distance between the target and the nearest boundary of the field of view.
The double dihedral field of view is identified by the two following axes:

- $axis_{1}$ is the axis of the first dihedral ($\vec n_{1}$ is its normal vector)
- $axis_{2}$ is the axis of the second dihedral ($\vec n_{2}$ is its normal vector)

If these 2 axes are orthogonal, we have a rectangular field of view. Otherwise, we have a trapezoidal field of view.

- [math]center[/math] is the vector representing the fov center (intersection between the two dihedrals central planes).

This axis must be obviously orthogonal to the 2 previous axis, but not necessarily in direct order.

The value of the g function is computed in two steps:

1. computing the angular distance between the target body and the first dihedral, and then subtracting it from the fov first half-aperture angle;
2. computing the angular distance between the target body and the second dihedral, and then subtracting it from the fov second half-aperture angle;

The lowest distance is then chosen to compute the g function.

- [math]g\lt0 :[/math] the target body is ouside the field of view;
- [math]g\gt0 :[/math] the target body is inside the field of view.

TargetInFieldOfViewDetector

This detector is defined from a sensor of an assembly part (see the Spacecraft user manual for details on sensor models).
A sensor can contain a spherical main target (PVCoordinatesProvider + radius) and a field of view (IFieldOfView) defined in the local frame of the sensor.
The g function of this sensor is positive when AT LEAST A PART this target is in the field, negative otherwise. So it detects the times when the target first enters and totaly leaves the field.
In order to proceed to this computation, the MAIN frame (and so all of the part's frames) is updated from the current SpacecraftState.

SensorInhibitionDetector

This detector is defined from a sensor of an assembly part (see the Spacecraft user manual for details on sensor models).
A sensor can contain an array of couples "inhibition target point" (PVCoordinatesProvider) / "inhibition field" (IFieldOfView) defined in the local frame of the sensor.
The g function of this detector is positive when AT LEAST ONE of the inhibition target is AT LEAST PARTIALLY in its associated inhibition field. So it detects the first and last times of inhibition.
In order to proceed to this computation, the MAIN frame (and so all of the part's frames) is updated from the current SpacecraftState.

MaskingDetector

This detector is defined from a sensor of an assembly part (see the Spacecraft user manual for details on sensor models).
A sensor model can have a list of masking objects. Those objects are :

- Parts of the same spacecraft (that must have A GEOMETRY property)
- Parts of others spacecrafts, built as [MIS_SENSORS_SecondSpc SecondarySpacecraft] objects (that must have A GEOMETRY property)
- Celestial bodies, built as a [MIS_SENSORS_PatriusBodySpheroid BodyShape] objects.
  The g function of this detector is positive when AT LEAST ONE of the potentially masking bodies

masks the target, negative otherwise.
The masking happens when the line

segment between the sensor and the target's center is cut between them by the shape of the masking object.
The "getMaskingObjectName" method allows the user to know wich one of the masking objects realises the masking event when occuring.

SensorVisibilityDetector

This detector is defined from a sensor of an assembly part (see the Spacecraft user manual for details on sensor models).
It is based on the three previous ones : the g function is positive when the main target is IN the field of view AND ALL of the inhibition targets are OUT of their associated inhibition field, AND NO masking object cuts the sensor / target center line segment. When used with earth ellipsoid, the computation of sight axis local elevation on earth uses a rapid algorithm with limited accuracy : typically 1.3e-4 degrees (worst case, satellite altitude 250 Km, latitude 45 deg., East sight direction) if sight axis is 25 degrees above horizon, and 1.4e-6 degrees (worst case) if sight axis is 0,25 degrees above horizon.

ExtremaSightAxisDetector

This detector detects local extrema angle between the sight axis of a part having the SENSOR property and a given target. The target is a PVCoordinatesProvder, it means that it can be a point on the central body (TopocentricFrame) or another vehicle (Propagator).

This detector doesn't take into account masking by the earth.

This detector has also been design to work without assembly and part. In this case, the sight axis is given as a Vector3D and is apply to the origin of the orbital local frame taking attitude into account.

The switching function is based on the extrema three body angle detector where P1 is the target, P2 is the origin of the frame (sensor part frame, or orbital local frame) and P3 is the sight axis.

See the [SPC_Home Spacecraft] user manual for details about sensor models and the [MIS_ORB_Home Orbit determination detectors] user manual for details about extrema three bodies angle detector.

Note: for the detection of events linked to the sight axis of an instrument, see also `CircularFieldOfViewDetector` and `CentralBodyMaskCircularFOVDetector`; they detect the passing of the sight axis close to a target (on ground, fixed or moving target, satellite, celestial body) taking account of the masking by the Earth. Events of the `ExtremaSigthAxisDetector` can also be filtered with phenomena detected with `CentralBodyMaskCircularFOVDetector`.

Getting Started

Modèle:SpecialInclusion prefix=$theme sub section="GettingStarted"/

Contents

Interfaces

All the detectors implement the interface :

Interface	Summary	Javadoc
EventDetector	This interface represents an event finder.	[[[:Modèle:JavaDoc4.15]]/fr/cnes/sirius/patrius/propagation/events/EventDetector.html EventDetector]

Classes

Class	Summary	Javadoc
CircularFieldOfViewDetector	This class handles target entry/exit events with respect to a satellite sensor circular field of view.	[[[:Modèle:JavaDoc4.15]]/fr/cnes/sirius/patrius/propagation/events/CircularFieldOfViewDetector.html CircularFieldOfViewDetector]
CentralBodyMaskCircularFOVDetector	This class handles target entry/exit events with respect to a satellite sensor circular field of view taking central body masking into account.	[[[:Modèle:JavaDoc4.15]]/fr/cnes/sirius/patrius/events/CentralBodyMaskCircularFOVDetector.html CentralBodyMaskCircularFOVDetector]
DihedralFieldOfViewDetector	This class handles target entry/exit events with respect to a satellite sensor dihedral field of view.	[[[:Modèle:JavaDoc4.15]]/fr/cnes/sirius/patrius/propagation/events/DihedralFieldOfViewDetector.html DihedralFieldOfViewDetector]
TargetInFieldOfViewDetector	This class handles events representing the time when the main target of a sensor enters the field of view.	[[[:Modèle:JavaDoc4.15]]/fr/cnes/sirius/patrius/events/sensor/TargetInFieldOfViewDetector.html TargetInFieldOfViewDetector]
SensorInhibitionDetector	This class handles events representing the time when at least one of the the inhibition target of a sensor enters the field of view.	[[[:Modèle:JavaDoc4.15]]/fr/cnes/sirius/patrius/events/sensor/SensorInhibitionDetector.html SensorInhibitionDetector]

MaskingDetector	This class handles events representing the time when at least one of the potentially masking objects that have been set to the sensor model is cutting the sensor / target center line segment.	[[[:Modèle:JavaDoc4.15]]/fr/cnes/sirius/patrius/events/sensor/MaskingDetector.html MaskingDetector]
SensorVisibilityDetector	This class handles events representing the time when, at the same time, 1) the main target of a sensor enters the field of view, 2) masking objects that have been set to the sensor model are not cutting the sensor/target center line segment and 3) no inhibition target is in its inhibition field.	[[[:Modèle:JavaDoc4.15]]/fr/cnes/sirius/patrius/events/sensor/SensorVisibilityDetector.html SensorVisibilityDetector]
ExtremaSightAxisDetector	This class handles events representing extrema of the angular angle between a sight axis on a vehicle and a target. This doesn't take into account eventual masking from any occulting body even from central body.	[[[:Modèle:JavaDoc4.15]]/fr/cnes/sirius/patrius/events/sensor/ExtremaSightAxisDetector.html ExtremaSightAxisDetector]

Récupérée de

Catégories :

- Pages avec des liens de fichiers brisés
- User Manual 4.15 Mission

Menu de navigation

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](#)

- Dernière modification de cette page le 21 novembre 2024 à 15:53.