Autonomous Vehicle Simulation (AVS) Laboratory, University of Colorado

Basilisk Technical Memorandum

Document ID: Basilisk-inertialUKF

INERTIAL UNSCENTED FILTER

Prepared by	T. Teil

Status: First Version
Scope/Contents
This module filters incoming star tracker measurements and reaction wheel data in order to get the best possible inertial attitude estimate. Measurements can be coming in from several camera heads.

Rev	Change Description	By	Date
1.0	First documentation	T. Teil	03/02/2019

Contents

---

1 Model Description

This module implements a square-root unscented Kalman Filter in order to achieve it's best state estimate. The estimated state is the attitude (MRPs) and the spacecraft rotation rate in the body frame.

1.1 Filter Setup

The equations and algorithm for the square root uKF are given in "inertialUKF_DesignBasis.pdf" [1] alongside this document.

The filter is therefore derived with the states being $\boldsymbol{X} = \begin{bmatrix}\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{N}} & \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}\end{bmatrix}^T$

The dynamics of the filter are given in Equations (1). τ is the total torque read in by the wheels.

$$\dot{\boldsymbol{\sigma}} = \frac{1}{4}[B]\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \tag{1}$$

$$\dot{\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}} = [I]^{-1}\tau \tag{2}$$

The following square-root uKF coefficients are used: $\alpha = 0.02$, and $\beta = 2$.

1.2 Measurements

The measurement model is given in equation 4. Since the input MRP may or may not be in the same "shadow" set as the state estimate, they are assured to be in the same representation. This prevents from getting residuals of 360°.

This is done following these steps:

- Current state estimate and measurements turned to quaternions
- State estimate is transposed (scaled by -1)
- Both quaternions are added and the sum turned to an MRP
- If the sum is greater than one the MRPs were not in the same representation and the measurement is shadowed

$$G_i(X) = \sigma \tag{3}$$

1.3 Measurements

The measurement model is given in equation 4. Since the input MRP may or may not be in the same "shadow" set as the state estimate, they are assured to be in the same representation. This prevents from getting residuals of 360°.

This is done following these steps:

- Current state estimate and measurements turned to quaternions
- State estimate is transposed (scaled by -1)
- Both quaternions are added and the sum turned to an MRP
- If the sum is greater than one the MRPs were not in the same representation and the measurement is shadowed

$$G_i(X) = \sigma \tag{4}$$

2 Module Functions

- **Read ST Messages:** Read in the messages from all available star trackers and orders them with respect to time of measurement
- **InertialUKFAggGyrData:** Aggregate the input gyro data into a combined total quaternion rotation to push the state forward. This information is stored in the main data structure for use in the propagation routines.

inertialUKFTimeUpdate, inertialUKFMeasUpdate, inertialStateProp, inertialUKFMeasModel

3 Module Assumptions and Limitations

4 Test Description and Success Criteria

4.1 Test 1

test StatePropInertialAttitude

4.2 Test 2

test StatePropRateInertialAttitude

4.3 Test 3

test StateUpdateInertialAttitude

4.4 Test 4

test StateUpdateRWInertialAttitude(

5 Test Parameters

Output Value Tested	Tolerated Value
$\boldsymbol{\sigma}_{R_1N}$	1e-12
${}^{\mathcal{N}}\boldsymbol{\omega}_{RN}$	1e-09
${}^{\mathcal{N}}\dot{\boldsymbol{\omega}}_{RN}$	1e-12
$\boldsymbol{\sigma}_{BA}$	1e-12

6 Test Results

Table 2: Test results

Check	Pass/Fail
Test 1	PASSED
Test 2	PASSED
Test 3	PASSED
Test 4	PASSED

7 User Guide

7.1 Filter Set-up, initialization, and I/O

In order for the filter to run, the user must set a few parameters:

- The unscented filter has 3 parameters that need to be set, and are best as:
```
filterObject.alpha = 0.02
filterObject.beta = 2.0
filterObject.kappa = 0.0
```
- The angle threshold under which the coarse sun sensors do not read the measurement:
```
FilterContainer.sensorUseThresh = 0.
```
- The process noise matrix:
```
qNoiseIn = numpy.identity(5)
qNoiseIn[0:3, 0:3] = qNoiseIn[0:3, 0:3]*0.01*0.01
qNoiseIn[3:5, 3:5] = qNoiseIn[3:5, 3:5]*0.001*0.001
filterObject.qNoise = qNoiseIn.reshape(25).tolist()
```

- The measurement noise value, for instance:
  `FilterContainer.qObsVal = 0.001`
- The initial covariance:
```
Filter.covar =
[1., 0.0, 0.0, 0.0, 0.0,
0.0, 1., 0.0, 0.0, 0.0,
0.0, 0.0, 1., 0.0, 0.0,
0.0, 0.0, 0.0, 0.02, 0.0,
0.0, 0.0, 0.0, 0.0, 0.02]
```
- The initial state :
  `Filter.state =[0.0, 0.0, 1.0, 0.0, 0.0]`

The messages must also be set as such:

- `filterObject.navStateOutMsgName = "inertial_state_estimate"`
- `filterObject.filtDataOutMsgName = "inertial_filter_data"`
- `filterObject.massPropsInMsgName = "adcs_config_data"`
- `filterObject.rwSpeedsInMsgName = "reactionwheel_output_states"`
- `filterObject.rwParamsInMsgName = "rwa_config_data_parsed"`
- `filterObject.gyrBuffInMsgName = "gyro_buffer_data"`

REFERENCES

[1] R. van der Merwe. The square-root unscented kalman filter for state and parameter-estimation. Acoustics, Speech, and Signal Processing, 2001.