.

Student Satellite Project
Indian Institute of Technology, Bombay
Powai, Mumbai - 400076, INDIA

Website: www.aero.iitb.ac.in/satlab

README - QuEST Algorithm

Guidance, Navigation and Controls Subsystem

---

es_main_quest_1.m

Code Type: MATLAB - Script
Code author: Shashank Singh
Created on: 29/04/2020
Last modified: –/–/—-
Reviwed by: NOT YET REVIEWED!
Description:
This is the main script, which runs the QuEST - 1 Algorithm.
Formula & References:
Reference: **Chapter 5,** Fundamentals of Spacecraft Attitude Determination and Control Authors: Markley, F. Landis, Crassidis, John L.
Input parameters:
The input arguments to the function are read from the **Input** folder. Here **N** refers to the number of input stars.

1. **es_input.mat** : The contents of which are-
   - **st_op_bi** : ((N, 3) - Matrix) - The body-frame vectors - (X,Y,Z), of the matched stars
   - **st_op_ri** : ((N, 3) - Matrix) - The inertial-frame vectors - (X,Y,Z), of the corresponding matched stars
   - **n_st_strs** : (Integer) - The number of stars matched by Star Matching
2. **es_epsilon.csv** : (Integer) - This is the the maximum value allowed for the for the characteristic equation. It would be used while doing iterations, using Newton Raphson method to find the maximum eigenvalue.

Output:
Writes the final estimated quaternion using QuEST-1 into **es_q_bi.csv** file in the **Output** folder.

es_main_quest_2.m

Code Type: MATLAB - Script
Code author: Shashank Singh
Created on: 29/04/2020
Last modified: –/–/—-

Reviwed by: NOT YET REVIEWED!
Description:
This is the main script, which runs the QuEST - 2 Algorithm.
Formula & References:
Reference: **Chapter 5,** Fundamentals of Spacecraft Attitude Determination and Control Authors: Markley, F. Landis, Crassidis, John L.
Input parameters:
The input arguments to the function are read from the **Input** folder. Here **N** refers to the number of input stars.

1. **es_input.mat** : The contents of which are-
   - **st_op_bi** : ((N, 3) - Matrix) - The body-frame vectors - (X,Y,Z), of the matched stars
   - **st_op_ri** : ((N, 3) - Matrix) - The inertial-frame vectors - (X,Y,Z), of the corresponding matched stars
   - **n_st_strs** : (Integer) - The number of stars matched by Star Matching
2. **es_epsilon.csv** : (Integer) - This file includes the maximum value allowed for the for the characteristic equation. It would be used while doing iterations, using Newton Raphson method to find the maximum eigenvalue.

Output:
Writes the final estimated quaternion using QuEST-2 into **es_q_bi.csv** file in the **Output** folder.

es_quest_common.m

Code Type: MATLAB - Function
Code author: Shashank Singh
Created on: 29/04/2020
Last modified: –/–/—-
Reviwed by: NOT YET REVIEWED!
Description:
This is the first and common function for both the QuEST-1 and QuEST-2 Algorithms. This function calculates the **B matrix** , **z vector** and the value for **lambda-not**, which are further used in finding the maximum eigenvalue of the K matrix and later used to calculate the final quaternion.
Formula & References:
Reference: **Chapter 5,** Fundamentals of Spacecraft Attitude Determination and Control Authors: Markley, F. Landis, Crassidis, John L.
Input parameters: Here **N** refers to the number of input stars.

1. **b_m** : ((N, 3) - Matrix) The body-frame vectors - (X,Y,Z), of the matched stars
2. **m_r** : ((N, 3) - Matrix) The inertial-frame vectors - (X,Y,Z), of the corresponding matched stars
3. **v_a** : ((N, 1) - Vector) The weights of the corresponding matched stars

Output:

1. **m_B** : ((3, 3) - Matrix) The **B Matrix**
2. **v_z** : ((3, 1) - Vector) The **z Vector**
3. **lamnot** : (Integer) - The sum of the weights of all stars

es_quest_newton.m

Code Type: MATLAB - Function
Code author: Shashank Singh
Created on: 29/04/2020
Last modified: –/–/—-
Reviwed by: NOT YET REVIEWED!
Description:
This is the first and common function for both the QuEST-1 and QuEST-2 Algorithms. This function calculates the **maximum eigenvalue of the K Matrix**, which is further used to calculate the final quaternion.
Formula & References:
Reference: **Chapter 5,** Fundamentals of Spacecraft Attitude Determination and Control Authors: Markley, F. Landis, Crassidis, John L.
Input parameters: Here **N** refers to the number of input stars.

1. **m_B** : ((3, 3) - Matrix) The **B Matrix**
2. **v_z** : ((3, 1) - Vector) The **z Vector**
3. **lamnot** : (Integer) - The sum of the weights of all stars
4. **epsilon** : (Float) - This is the the maximum value allowed for the for the characteristic equation. It would be used while doing iterations, using Newton Raphson method to find the maximum eigenvalue.

Output:
lam : (Float) - The maximum eigenvalue of the K matrix.

es_quest_1_final.m

Code Type: MATLAB - Function
Code author: Shashank Singh
Created on: 29/04/2020
Last modified: –/–/—-
Reviwed by: NOT YET REVIEWED!
Description:
This is the final and separate function for QuEST-1 Algorithm. This function calculates the **final estimated quaternion**.
Formula & References:
Reference: **Chapter 5,** Fundamentals of Spacecraft Attitude Determination and Control Authors: Markley, F. Landis, Crassidis, John L.
Input parameters:

1. **m_B** : ((3, 3) - Matrix) The **B Matrix**
2. **v_z** : ((3, 1) - Vector) The **z Vector**
3. **lam** : (Float) - The maximum eigenvalue of the K matrix

Output:
q_bi : ((4,1) - Vector) - The final estimated quaternion, using QuEST-1.

es_quest_2_final.m

Code Type: MATLAB - Function
Code author: Shashank Singh
Created on: 29/04/2020
Last modified: –/–/—-
Reviwed by: NOT YET REVIEWED!
Description:
This is the final and separate function for QuEST-2 Algorithm. This function calculates the **final estimated quaternion**.
Formula & References:
Reference: **Chapter 5,** Fundamentals of Spacecraft Attitude Determination and Control Authors: Markley, F. Landis, Crassidis, John L.
Input parameters:

1. **m_B** : ((3, 3) - Matrix) The **B Matrix**
2. **v_z** : ((3, 1) - Vector) The **z Vector**
3. **lam** : (Float) - The maximum eigenvalue of the K matrix

Output:
q_bi : ((4,1) - Vector) - The final estimated quaternion, using QuEST-2.