.

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: 08/08/2020
Reviwed by: NOT YET REVIEWED!
Description:
This is the main script, which runs the QuEST - 1 Algorithm. It also runs the sequential rotation function, in case the QuEST - 1 fails in the given initial frame.
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-
   - **op_bi** : ((N, 4) - Matrix) - The body-frame vectors - (X,Y,Z), of the matched stars
   - **op_ri** : ((N, 4) - Matrix) - The inertial-frame vectors - (X,Y,Z), of the corresponding matched stars
   - **N** : (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 as well as the **Input** folder(to be used for Sequential Rotation later).

es_main_quest_2.m

Code Type: MATLAB - Script
Code author: Shashank Singh

Created on: 29/04/2020
Last modified: 08/08/2020
Reviwed by: NOT YET REVIEWED!
Description:
This is the main script, which runs the QuEST - 2 Algorithm. This algorithm different then QuEST-1 Algorithm in the sense that it uses the the fourth column of adjoint of the calculated H Matrix as the quaternion.It also runs the sequential rotation function, in case the QuEST - 2 fails in the given initial frame.
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-

    - **op_bi** : ((N, 4) - Matrix) - The body-frame vectors - (X,Y,Z), of the matched stars
    - **op_ri** : ((N, 4) - Matrix) - The inertial-frame vectors - (X,Y,Z), of the corresponding matched stars
    - **N** : (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-2 into **es_q_bi.csv** file in the **Output** folder as well as the **Input** folder(to be used for Sequential Rotation later).

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 second 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: 08/08/2020
Reviwed by: **NOT YET REVIEWED!**
Description:
This is the final and separate function for QuEST-1 Algorithm. This function calculates the **final estimated quaternion**. It also checks if **check_value** is close to zero. If **check_value** is smaller than the threshold value, then $q\_bi = [-1; -1; -1; -1]$ is returned, which indicates the main script that QUEST has failed in this frame and then sequential rotation is used.

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. **lam** : (Float) - The maximum eigenvalue of the K matrix
4. **es_seq_error** : (Float) - The minimum allowed value of **check_value**. If **check_value** is less than this value, then sequential rotation is used.

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: 08/08/2020
Reviwed by: NOT YET REVIEWED!
Description:
This is the final and separate function for QuEST-2 Algorithm. This function calculates the **final estimated quaternion**. It also checks if **check_value** is close to zero. If **check_value** is smaller than the threshold value, then $q\_bi = [-1; -1; -1; -1]$ is returned, which indicates the main script that QUEST has failed in this frame and then sequential rotation is used.
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. **lam** : (Float) - The maximum eigenvalue of the K matrix
4. **es_seq_error** : (Float) - The minimum allowed value of **check_value**. If **check_value** is less than this value, then sequential rotation is used.

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

es_quest_1_seq_rot.m

Code Type: MATLAB - Function
Code author: Shashank Singh
Created on: 08/08/2020
Last modified: –/–/—-

Reviwed by: NOT YET REVIEWED!
Description:
This function calculates the **final estimated quaternion.**This function first finds the preferred frame for sequential rotation and finds the estimated quaternion in the changed frame using QUEST - 1. This quaternion is later converted to quaternion in the original initial frame. If **check_value** is again smaller than the threshold value, then a new preferred frame is found and sequential rotation is used in the new frame. This process continues for all the three frames(the three frames are, inertial frame rotated by 180 degrees about x,y,z axes) until the correct quaternion is found.
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
4. **q_bi_prev** : ((4,1) - Vector) - The previous quaternion value

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

es_quest_2_seq_rot.m

Code Type: MATLAB - Function
Code author: Shashank Singh
Created on: 08/08/2020
Last modified: –/–/—-
Reviwed by: NOT YET REVIEWED!
Description:
This function calculates the **final estimated quaternion.**This function first finds the preferred frame for sequential rotation and finds the estimated quaternion in the changed frame using QUEST - 2. This quaternion is later converted to quaternion in the original initial frame. If **check_value** is again smaller than the threshold value, then a new preferred frame is found and sequential rotation is used in the new frame. This process continues for all the three frames(the three frames are, inertial frame rotated by 180 degrees about x,y,z axes) until the correct quaternion is found.
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

4. **q_bi_prev** : ((4,1) - Vector) - The previous quaternion value

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