



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

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



README: 4-Star Matching Algorithm

Guidance, Navigation and Controls Subsystem

Contents

1	st_main_4SM_v2.m	1
2	st_gnrt_bi.m	3
3	st_4SM.m	3
4	st_gnrt_ip_4SM.m	4
5	st_gnrt_SIM.m	4
6	st_gnrt_CSPA.m	5
7	st_gnrt_SMM.m	5
8	st_update_match_matrix.m	6
9	st_verify_4SM.m	7
10	st_check_ID_err.m	7
11	st_gnrt_op_4SM_v2.m	8
12	st_pseudo_input.m	8
13	st_main_4SM_deprecated.m	9
14	Variable Description	10

1 st_main_4SM_v2.m

Code Type: MATLAB - Script

Code author: KT Prajwal Prathiksh

Created on: 08/06/2020

Last modified: {/| -

Reviewed by: NOT YET REVIEWED!

Description:

This is the main script, which runs the 4-Star Matching Algorithm, with the verification step.

Note: This script can work with n -stars identified by *Feature Extraction block*

Formula & References:

Reference:

1. 4-Star Matching Algorithm [1]
2. Geometric Voting Algorithm [2]

Input parameters:

1. **st_constants_4SM.mat**: The contents of which are -
 - (a) **Focal_Length**
 - (b) **st_M_EPS**
 - (c) **st_n_GC**
 - (d) **st_n_RC**
 - (e) **st_DELTA**
 - (f) **st_M**
 - (g) **st_Q**
 - (h) **es_N_EST**
 - (i) **st_ITER_MAX_4SM**
 - (j) **st_verify_tol**
2. **st_input.mat**: The contents of which are -
 - (a) **fe_output**
 - (b) **fe_n_str**
3. **st_test_input.mat**: The contents of the test cases which are -
 - (a) **fe_output** - Test input
 - (b) **fe_n_str** - Test input

Output:

1. Writes the following output variables of Star-Matching in [./Star_Matching/4_Star_Matching/Output/st_output.mat](#) directory
 - (a) **st_iter_total**
 - (b) **st_N_Match**
 - (c) **st_Match**
 - (d) **st_N_UnMatch**
 - (e) **st_UnMatch**
 - (f) **st_op_bi**
 - (g) **st_op_ri**
 - (h) **st_N_Fail**
 - (i) **st_Fail**
 - (j) **st_N_Verify**

(k) `st_Verify`

2. Writes the following output variables of Star-Matching as input for Estimation in `./Estimation/Input/es.input.mat` directory

(a) `st_N_Verify`

(b) `st_op_bi`

(c) `st_op_ri`

2 `st_gnrt_bi.m`

Code Type: MATLAB - Function

Code author: KT Prajwal Prathiksh

Created on: 26/04/2020

Last modified: 29/04/2020

Reviewed by: NOT YET REVIEWED!

Description:

This function generates the body-frame vectors of the stars from the centroids identified through means of Feature Extraction

Formula & References:

References:

1. Refer Appendix [3]

Input parameters:

1. `fe_output`
2. `fe_n_str`
3. `Focal_Length`

Output:

1. `st_bi`

3 `st_4SM.m`

Code Type: MATLAB - Function

Code author: KT Prajwal Prathiksh

Created on: 29/04/2020

Last modified: {/{/| -

Reviewed by: NOT YET REVIEWED!

Description:

This function runs the 4-Star Matching Algorithm. It executes the algorithm to match stars, and returns the result.

Formula & References:

Reference:

1. 4-Star Matching Algorithm [1]

Input parameters:

1. `st_4SM_input`
2. `st_RF_SC`
3. `st_4SM_constants`

Output:

1. `st_n_match`
2. `st_result`

4 `st_gnrt_ip_4SM.m`

Code Type: MATLAB - Function

Code author: KT Prajwal Prathiksh

Created on: 26/04/2020

Last modified: 29/04/2020

Reviewed by: NOT YET REVIEWED!

Description:

This function generates the input for 4-Star Matching Algorithm, using the body-frame vectors of 4 identified stars from the image

Formula & References:

|

Input parameters:

1. `st_4SM_input`

Output:

1. `st_c_img_AngDst`
2. `st_c_fe_ID`

5 `st_gnrt_SIM.m`

Code Type: MATLAB - Function

Code author: KT Prajwal Prathiksh

Created on: 26/04/2020

Last modified: 29/04/2020

Reviewed by: NOT YET REVIEWED!

Description:

This function generates the the Star Identification Matrix (SIM) from a given array of angular distances between those of four stars, using the Reference Star Catalogue and the Guide Star Catalogue

Formula & References:

|

Input parameters:

1. `st_c_img_AngDst`
2. `st_RF_SC`

3. st_4SM_constants

Output:

1. st_SIM

6 st_gnrt_CSPA.m

Code Type: MATLAB - Function

Code author: KT Prajwal Prathiksh

Created on: 26/04/2020

Last modified: 29/04/2020

Reviewed by: NOT YET REVIEWED!

Description:

This function generates the candidate star pair array (CSPA) from the value of angular distance calculated between two image stars

Formula & References:

Reference:

1. Refer Section - 2 [1]

Input parameters:

1. st_AngDst
2. st_DELTA
3. st_M
4. st_Q
5. st_RF_SC

Output:

1. st_CSPA
2. st_INDEX

7 st_gnrt_SMM.m

Code Type: MATLAB - Function

Code author: KT Prajwal Prathiksh

Created on: 26/04/2020

Last modified: 29/04/2020

Reviewed by: NOT YET REVIEWED!

Description:

This function evaluates the Star Identification Matrix to identify the stars that have been matched through the 4-Star Matching Algorithm to generate the Star Matched Matrix (SMM).

Of the four stars that are provided as input to the 4-Star Matching Algorithm, the S_i star is said to have been matched, if only one such row of the Star Identification Matrix as given below is present:

1. $S_1 = [1, 1, 1, 0, 0, 0]$
2. $S_2 = [1, 0, 0, 1, 1, 0]$
3. $S_3 = [0, 1, 0, 1, 0, 1]$
4. $S_4 = [0, 0, 1, 0, 1, 1]$

This pattern arises out of the fact that angular distances that are provided as input follow the order: $(S_1, S_2); (S_1, S_3); (S_1, S_4); (S_2, S_3); (S_2, S_4); (S_3, S_4)$

Formula & References:

Reference:

1. Refer Section - 2 [\[1\]](#)

Input parameters:

1. `st_SIM`
2. `st_n_GC`
3. `st_c_fe_ID`

Output:

1. `st_SMM`
2. `st_mtch_rows`

8 `st_update_match_matrix.m`

Code Type: MATLAB - Function

Code author: KT Prajwal Prathiksh

Created on: 29/04/2020

Last modified: {/{/| -

Reviewed by: NOT YET REVIEWED!

Description:

This function updates the Match Matrices associated with 4-Star Matching Algorithm.

Formula & References:

{

Input parameters:

1. `st_result`
2. `st_Match`
3. `st_UnMatch`
4. `st_N_Match`
5. `st_N_UnMatch`

Output:

1. `st_Match`
2. `st_UnMatch`
3. `st_N_Match`
4. `st_N_UnMatch`

9 st_verify_4SM.m

Code Type: MATLAB - Function

Code author: KT Prajwal Prathiksh

Created on: 09/06/2020

Last modified: {/{}| -

Reviewed by: NOT YET REVIEWED!

Description:

Performs the verification step for 4-Star Matching Algorithm. It assigns votes to the matched stars that get verified. Returns only those stars which have 100% votes

Formula & References:

References:

1. Geometric voting algorithm, verification step [2]

Input parameters:

1. st]_Match
2. st_N_Match
3. st_GD_SC
4. st_verify_tol

Output:

1. st_N_Verify
2. st_Verify
3. st_N_Fail
4. st_Fail

10 st_check_ID_err.m

Code Type: MATLAB - Function

Code author: KT Prajwal Prathiksh

Created on: 29/04/2020

Last modified: {/{}| -

Reviewed by: NOT YET REVIEWED!

Description:

This function evaluates the result of Star-Matching with one of the input stars having already been matched, and checks for discrepancy error (if any) in the previously matched SSP-ID, and newly generated SSP-ID.

NOTE: This code is currently under development! DO NOT USE IT!

Formula & References:

|

Input parameters:

1. st_result

2. `st_ref_ID`

3. `st_N`

Output:

1. `st_flag_ID_err`

11 `st_gnrt_op_4SM_v2.m`

Code Type: MATLAB - Function

Code author: KT Prajwal Prathiksh

Created on: 26/04/2020

Last modified: 09/06/2020

Reviewed by: NOT YET REVIEWED!

Description:

This function evaluates the Star Matched Matrix (SMM), to generate the output in the format required by Estimation - Body-frame vectors, and Inertial-frame vectors of the matched stars

Formula & References:

|

Input parameters:

1. `st_Match`

2. `st_GD_SC`

Output:

1. `st_op_bi`

2. `st_op_ri`

12 `st_pseudo_input.m`

Code Type: MATLAB - Script

Code author: KT Prajwal Prathiksh

Created on: 26/04/2020

Last modified: 29/04/2020

Reviewed by: NOT YET REVIEWED!

Description:

This script generates fictitious input, that resembles the output expected out of Feature Extraction.

NOTE: The values are random, and therefore should not be treated as test-cases

Formula & References:

|

Input parameters:

1. `fe_n_str`

Output:

1. Writes the following output variable in `./Star_Matching/4_Star_Matching/Input/st_input.mat` directory:

- (a) **st_input.mat**: The contents of which are -
 - i. **fe_output**
 - ii. **fe_n_str** :

13 st_main_4SM_deprecated.m

Code Type: MATLAB - Script

Code author: KT Prajwal Prathiksh

Created on: 26/04/2020

Last modified: 29/04/2020

Reviewed by: NOT YET REVIEWED!

Description:

This is the main script, which runs the 4-Star Matching Algorithm.

Note: This script can only work with 4-stars identified by *Feature Extraction block*, and therefore has been **DEPRECATED!**

Formula & References:

Reference:

1. 4-Star Matching Algorithm [1]

Input parameters:

1. **st_constants_4SM.mat**: The contents of which are -
 - (a) **Focal_Length**
 - (b) **st_M_EPS**
 - (c) **st_n_GC**
 - (d) **st_n_RC**
 - (e) **st_DELTA**
 - (f) **st_M**
 - (g) **st_Q**
 - (h) **es_N_EST**
 - (i) **st_ITER_MAX_4SM**
2. **st_input.mat**: The contents of which are -
 - (a) **fe_output**
 - (b) **fe_n_str**

Output:

1. Writes the following output variables of Star-Matching in [./Star_Matching/4_Star_Matching/Output/st_output_deprecated.mat](#) directory
 - (a) **n_st_strs**
 - (b) **st_mtch_rows**
 - (c) **st_op_bi**
 - (d) **st_op_ri**

- (e) **st_SIM**
 - (f) **st_SMM**
2. Writes the following output variables of Star-Matching as input for Estimation in [./Estimation/Input/es_input_deprecated.mat](#) directory
 - (a) **n_st_strs**
 - (b) **st_op_bi**
 - (c) **st_op_ri**

14 Variable Description

1. **Focal_Length** : (Float) The focal length of the optics system. Units in mm
2. **st_M_EPS** : (Float) The machine epsilon of the platform where the algorithm will be executed
3. **st_n_GC** : (Integer) The number of stars (= number of rows) in the Guide Star Catalogue
4. **st_n_RC** : (Integer) The number of star-pairs (= number of rows) in the Reference Star Catalogue
5. **st_DELTA** : (Float) The δ constant that determines the tolerance of the size window when searching for an angular distance value in the Reference Star Catalogue
6. **st_M** : (Float) The slope of the Z-vector line
7. **st_Q** : (Float) The y-intercept of the Z-vector line
8. **es_N_EST** : (Integer) The minimum number of pairs of body-frame (b_i) and inertial-frame (r_i) vectors required by *Estimation block* to provide the attitude within the accuracy requirements
9. **st_ITER_MAX_4SM** : (Integer) The max number of iterations of 4-Star Matching that will be allowed to execute irregardless of whether all the stars match or not - owing to the time constraint on the *Star-Matching block*
10. **st_verify_tol** : (Double) The tolerance within which the angular distance (in $\cos(\theta)$) of a pair of image stars should have match with the corresponding pair of stars from the Star Catalogue. Units in *percentage*
11. **fe_output** : ((N, 3) - Matrix) - The output of Feature Extraction block - which contains the centroids of the identified stars as well ID of the identified star. The centroid is represented in (X,Y) format, with the origin at the center of the sensor. Unit of centroid: mm
12. **fe_n_str** : (Integer) - The number of stars identified by feature extraction
13. **st_iter_total** : (Integer) - Counter variable which keeps track of the number of iterations of 4-Star Matching that has occurred
14. **st_N_Match** : (Integer) - Number of matched stars in st_Match matrix

15. **st_Match** : ((st_N_Match, 5) - Matrix) - This matrix contains the entries of the stars that have been matched so far. The columns are as follows:
 - (a) 1st column - Feature Extraction ID
 - (b) 2nd, 3rd, 4th columns - (X, Y, Z) unit body-frame vector
 - (c) 5th column - The matched SSP-ID
 16. **st_N_UnMatch** : (Integer) - Number of stars yet to be matched in st_UnMatch matrix
 17. **st_UnMatch** : ((st_N_UnMatch, 4) - Matrix) - This matrix contains the entries of the stars that are yet to be matched so far. The columns are as follows:
 - (a) 1st column - Feature Extraction ID
 - (b) 2nd, 3rd, 4th columns - (X, Y, Z) unit body-frame vector
 18. **st_op_bi** : ((st_N_Verify, 4) - Matrix) - The body-frame vectors b_i of the image stars that have been matched through Star-Matching. The columns include:
 - (a) 1st column - Feature Extraction ID
 - (b) 2nd, 3rd, 4th columns - (X, Y, Z) unit body-frame vector
 19. **st_op_ri** : ((st_N_Verify, 4) - Matrix) - The corresponding inertial-frame vectors r_i of the matched stars obtained from the Guide Star Catalogue. The columns include:
 - (a) 1st column - Corresponding SSP-ID of the matched star
 - (b) 2nd, 3rd, 4th columns - (X, Y, Z) unit inertial-frame vector
- NOTE:**
- (a) (N) in both cases should be equal!
 - (b) The i^{th} body-frame and the i^{th} inertial-frame vector should correspond to the same star!
20. **st_N_Fail** : (Integer) - Number of stars that failed the verification step
 21. **st_Fail** : ((st_N_Fail, 6) - Matrix) - This matrix contains the entries of the stars that failed the verification step. The columns are as follows:
 - (a) 1st column - Feature Extraction ID
 - (b) 2nd, 3rd, 4th columns - (X, Y, Z) unit body-frame vector
 - (c) 5th column - The matched SSP-ID
 - (d) 6th column - Number of votes
 22. **st_N_Verify** : (Integer) - Number of stars that passed the verification step
 23. **st_Verify** : ((st_N_Verify, 6) - Matrix) - This matrix contains the entries of the stars that have been verified. The columns are as follows:
 - (a) 1st column - Feature Extraction ID
 - (b) 2nd, 3rd, 4th columns - (X, Y, Z) unit body-frame vector
 - (c) 5th column - The matched SSP-ID

- (d) 6th column - Number of votes
24. **st_bi** : ((N, 4) - Matrix) - Generates the body-frame vectors for all the stars identified by Feature Extraction using the centroids. The body-frame vectors are unit-vectors represented in (X,Y,Z) format, with the origin at the center of the sensor and positive z-axis pointing out of the lens. The columns of matrix are as follows:
- (a) 1st column - Feature Extraction ID
 - (b) 2nd, 3rd, 4th columns - (X,Y,Z) unit body-frame vector
25. **st_4SM_input** : ((4, 4) - Matrix) - The input to 4-Star Matching Algorithm - four stars identified by Feature Extraction, along with their body-frame vectors (a unit-vector represented in (X,Y,Z) format, with the origin at the center of the sensor and positive z-axis pointing out of the lens). The columns of the matrix are as follows:
- (a) 1st column - Feature Extraction ID
 - (b) 2nd, 3rd, 4th columns - (X,Y,Z) unit body-frame vector
26. **st_RF_SC** : ((st_n_RC, 3) - Matrix) - The Reference Star Catalogue, which has the following columns:
- (a) **SSP_ID_1** - The SSP-ID of i^{th} star
 - (b) **SSP_ID_2** - The SSP-ID of j^{th} star
 - (c) **K_Vec** - The K-Vector value determined uniquely using the dot product of the Cartesian unit vector corresponding to the i^{th} and j^{th} star ($i \neq j, \forall i, j$)
27. **st_4SM_constants** : ((4, 1) - Matrix) - An array that contains the following constants:
- (a) **st_n_GC**
 - (b) **st_M**
 - (c) **st_Q**
 - (d) **st_DELTA**
28. **st_n_match** : (Integer) - The number of stars that have been matched
29. **st_result** : ((4,2) - Matrix) - Output of 4-Star Matching. The columns of the matrix are as follows:
- (a) 1st column - Feature Extraction ID
 - (b) 2nd column - The matched SSP-ID (*If no match was made - 0 is assigned*)
30. **st_c_img_AngDst** : ((6,1) - Matrix) - Has the angular distances (in $\cos(\theta)$) between those of four stars in the following order:
 $(S_1, S_2); (S_1, S_3); (S_1, S_4); (S_2, S_3); (S_2, S_4); (S_3, S_4);$
NOTE: THE ABOVE ORDER IS IMPORTANT, AND SHOULD BE FOLLOWED!
31. **st_c_fe_ID** : ((4,1) - Matrix) - Has the Feature Extraction IDs of stars that are used to generate st_c_img_AngDst, in the following order:
 $[S_1; S_2; S_3; S_4]$

32. **st_SIM** : ((st_n_GC, 6) - Matrix) - The Star Identification Matrix (SIM). The i^{th} row of the matrix corresponds to the i^{th} SSP-ID star. There are thus as many rows in SIM as there are in the Guide Star Catalogue = (st_n_GC). The j^{th} column corresponds to a boolean value. If an element SIM_{ij} is 1 $\implies S_j^{th}$ input star matched to the i^{th} star of the Guide Star Catalogue
33. **st_AngDst** : (Float) - The angular distance calculated between two stars (in $\cos(\theta)$)
34. **st_CSPA** : ((X,1) - Matrix) - The possible SSP-ID matches for given angular distance
35. **st_INDEX** : ((1,2) - Matrix) - The start and stop indices of possible matches generated for given angular distance
36. **st_SMM** : ((4,3) - Matrix) - The Star Matched Matrix. The columns of the matrix are as follows:
 - (a) 1st column - Feature Extraction ID
 - (b) 2nd column - The matched SSP-ID (*If no match was made - 0 is assigned*)
 - (c) 3rd column - The number of rows in SIM that matched the condition of S_i star
37. **st_mtrch_rows** : ((4, X) - Cell Array) - The columns of the cell array are as follows:
 - (a) 1st column - Feature Extraction ID
 - (b) 2nd column - i^{th} element consists of a (X,1) - Matrix, that stores the SSP-IDs that matched the i^{th} condition
38. **st_result** : ((4,2) - Matrix) - Output of 4-Star Matching. The columns of the matrix are as follows:
 - (a) 1st column - Feature Extraction ID
 - (b) 2nd column - The matched SSP-ID (*If no match was made - 0 is assigned*)
39. **st_ref_ID** : ((4,2) - Matrix) - This matrix is stores the previously matched SSP-IDs of the stars taken from st.Match matrix. This is used to check if the newly matched SSP-ID is the same as the previous SSP-ID. The columns of this matrix are as follows:
 - (a) 1st column - Feature Extraction ID
 - (b) 2nd column - The previously matched SSP-ID (The unmatched stars have 0 as their ID)
40. **st_N** : (Integer) - Number of previously matched stars used as input
41. **st_flag_ID_err** : (boolean) - Returns 1 if the previous and the new SSP-ID do not match, or if the previously matched star was not matched in st_result. Returns 0 only if they match!
42. **st_GD_SC** : ((st_n_GC, 4) - Matrix) - The Guide Star Catalogue, which has the following columns:
 - (a) **SSP_ID** - The SSP-ID of Guide Stars

- (b) **X** - X component of the unit vector, corresponding to the star
- (c) **Y** - Y component of the unit vector, corresponding to the star
- (d) **Z** - Z component of the unit vector, corresponding to the star

References

- [1] Y. Dong, F. Xing, and Z. You, "Brightness independent 4-star matching algorithm for lost-in-space 3-axis attitude acquisition", *Tsinghua Science & Technology*, vol. 11, no. 5, pp. 543{548, 2006.
- [2] M. Kolomenkin, S. Pollak, I. Shimshoni, and M. Lindenbaum, "Geometric voting algorithm for star trackers", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, pp. 441{456, 2008.
- [3] A. O. Erlank, "Development of cubestar: A cubesat-compatible star tracker", PhD thesis, Stellenbosch: Stellenbosch University, 2013.