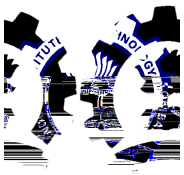
.



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

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


# README - Tracking Mode Algorithm

Guidance, Navigation and Controls Subsystem

---

## sm_TM_main.m()

**Code author:** Pranjal Gupta
**Created on:** 18/01/2021
**Last modified:** 18/01/2021
**Reviwed by:** NOT YET REVIEWED
**Description:**
This is the main script of the Tracking Mode algorithm, which calls the three blocks (Centroid Prediction (CP), Radius Based Matching (RBM) and the Star Neighbourhood Table (SNT)) in accordance with the main flowchart for the algorithm.
**Formula & References:**
**Input parameters:**
The input arguments to the function must be written here. The format would

1. **fe_output** : ((N,2) - Matrix) - The output matrix from Feature Extraction containing the centroids obtained from the image at an instant.

2. **sm_output_curr** : ((M,3) - Matrix) - The Star Matching output at the $K^{th}$ iteration, containing the (x,y) coordinates in the first two columns and the Star ID in the third column.

3. **sm_output_prev** : ((O,3) - Matrix) - The Star Matching output at the $(K-1)^{th}$ iteration, containing the (x,y) coordinates in the first two columns and the Star ID in the third column.

4. **sm_consts_TM** : (binary .mat file) - This binary MATLAB file contains all the constants required in the Tracking Mode algorithm (the current function requires the Focal Length, FOV width, RBM Radius, CP radius and $N_{th}$).

5. **sm_TM_SNT** : ((5060, N) - Matrix) - The Star Neighbourhood Table

6. **sm_catalogues** : (binary .mat file) - Contains the Guide star catalogue and the Preprocessed Star Catalogue

**Output:**

1. **sm_output** : ((K,4) - Matrix) - The final output of the Star Matching block (operating in Tracking Mode). Contains the Star IDs and the corresponding star unit vector from the Guide Star Catalogue.

## sm_TM_CP_main.m()

**Code author:** Pranjal Gupta
**Created on:** 15/10/2020
**Last modified:** 15/10/2020
**Reviwed by:** NOT YET REVIEWED
**Description:**
This script deploys centroid prediction (and angular velocity estimation) at the $(K+1)^{th}$ frame given the common centroids among the $(K-1)^{th}$ and $(K)^{th}$ frames.
**Formula & References:**
Refer Mark Knutson's thesis containg the matrix calculations for centroid prediction.
**Input parameters:**

1. **sm_TM_CP_prevmat** : ((N,2) - Matrix) - The common centroids from the Feature extraction output at the $(K-1)^{th}$ iteration, containing the (x,y) coordinates of the centroids.
2. **sm_TM_CP_prevmat** : ((N,2) - Matrix) - The common centroids from the Feature extraction output at the $(K)^{th}$ iteration, containing the (x,y) coordinates of the centroids.
3. **sm_TM_CP_F** : (double) - The focal length of the star sensor (Units - cm)

**Output:**

1. **sm_TM_CP_predmat** : ((N,2) - Matrix) - The predicted centroids at the $(K+1)^{th}$ frame.

## sm_TM_RBM_main.m()

**Code author:** Pranjal Gupta
**Created on:** 24/09/2020
**Last modified:** 24/09/2020
**Reviwed by:** NOT YET REVIEWED
**Description:**
This script deploys Radius Based Matching to match the true centroids (obtained from feature extraction) to the predicted centroids (obtained from the CP block).
**Formula & References:**
**Input parameters:**

1. **sm_TM_RBM_predmat** : ((N,2) - Matrix) - The predicted centroids at the $(K+1)^{th}$ frame.
2. **sm_TM_RBM_truemat** : ((M,2) - Matrix) - The true centroids at the $(K+1)^{th}$ frame obtained from feature extraction.
3. **sm_TM_RBM_R** : (double) - The radius value to be used for carrying out Radius based matching between predicted and true centroids (Units - cm)
4. **is_dx** : (boolean) - if True : predicted and true centroids are sorted according to their x-coordinates
5. **sort_before_match** : (boolean) - if True : Implement sorting before matching optimisation algorithm (to carry out faster RBM)

**Output:**

1. **sm_TM_RBM_matchmat** : ((L,4) - Matrix) - The matrix of matched true and predicted centroids

## sm_TM_CP_jacobian.m()

**Code author:** Pranjal Gupta
**Created on:** 15/10/2020
**Last modified:** 15/10/2020
**Reviwed by:** NOT YET REVIEWED
**Description:**
Constructs the Jacobian Matrix given a set of centroids and star sensor's focal length
**Formula & References:**
**Input parameters:**

1. **n_centroid** : (int) - Number of common centroids among the $(K)^{th}$ and the $(K-1)^{th}$ frame.
2. **sm_TM_CP_F** : (double) - The focal length of the star sensor (Units - cm)

**Output:**

1. **m_A** : ((2N,3) - Matrix) - Jacobian Matrix constructed using the Focal Length and the u,v coordinates from the set of centroids

## sm_TM_RBM_sortmat.m()

**Code author:** Pranjal Gupta
**Created on:** 24/09/2020
**Last modified:** 24/09/2020
**Reviwed by:** NOT YET REVIEWED
**Description:**
This script sorts the predicted and true centroids according to either their x-coordinates or y-coordinates (depends on the value of is_dx)
**Formula & References:**
**Input parameters:**

1. **sm_TM_RBM_predmat** : ((N,2) - Matrix) - The unsorted matrix of predicted centroids obtained from Centroid Prediction.
2. **sm_TM_RBM_truemat** : ((M,2) - Matrix) - The unsorted matrix of true centroids obtained from feature extraction at the $(K+1)^{th}$ frame.
3. **is_dx** : (boolean) - if True - predicted and true centroids are sorted according to their x-coordinates

**Output:**

1. **sorted_predmat** : ((N,2) - Matrix) - The matrix of predicted centroids sorted according to either their x-coordinates or y-coordinates.
2. **sorted_truemat** : ((M,2) - Matrix) - The matrix of true centroids (obtained from feature extraction at the $(K+1)^{th}$ frame) sorted according to either their x-coordinates or y-coordinates.

## sm_TM_RBM_normmatch.m()

**Code author:** Pranjal Gupta
**Created on:** 24/09/2020
**Last modified:** 24/09/2020
**Reviwed by:** NOT YET REVIEWED
**Description:**
This script implements brute-force Radius Based Matching to match the predicted and true centroids without any optimisations
**Formula & References:**
**Input parameters:**

1. **sorted_predmat** : ((N,2) - Matrix) - The matrix of predicted centroids sorted according to either their x-coordinates or y-coordinates.
2. **sorted_truemat** : ((M,2) - Matrix) - The matrix of true centroids (obtained from feature extraction at the $(K+1)^{th}$ frame) sorted according to either their x-coordinates or y-coordinates.
3. **sm_TM_RBM_R** : (double) -The radius value to be used for carrying out Radius based matching between predicted and true centroids (obtained from sm_consts_TM)

**Output:**

1. **sm_TM_RBM_matchmat** : ((L,4) - Matrix) - The matrix of matched true and predicted centroids.

## sm_TM_RBM_sortmatch.m()

**Code author:** Pranjal Gupta
**Created on:** 24/09/2020
**Last modified:** 24/09/2020
**Reviwed by:** NOT YET REVIEWED
**Description:**
This script implements the Sorting before Matching optimisation technique for the Radius Based Matching algorithm
**Formula & References:** Refer Zhang2019 here.
**Input parameters:**

1. **sorted_predmat** : ((N,2) - Matrix) - The matrix of predicted centroids sorted according to either their x-coordinates or y-coordinates.
2. **sorted_truemat** : ((M,2) - Matrix) - The matrix of true centroids (obtained from feature extraction at the $(K+1)^{th}$ frame) sorted according to either their x-coordinates or y-coordinates.
3. **sm_TM_RBM_R** : (double) -The radius value to be used for carrying out Radius based matching between predicted and true centroids (obtained from sm_consts_TM)

**Output:**

1. **sm_TM_RBM_matchmat** : ((L,4) - Matrix) - The matrix of matched true and predicted centroids.

## sm_TM_SNT_match.m()

**Code author:** Pranjal Gupta
**Created on:** 18/01/2021
**Last modified:** 18/01/2021
**Reviwed by:** NOT YET REVIEWED
**Description:**
This script implements an algorithm to identify the unmatched centroids from fe_output using the identified stars and the Star Neighbourhood Table
**Formula & References:**
**Input parameters:**

1. **sm_TM_RBM_matchmat** : ((L,5) - Matrix) - The matrix of matched true and predicted centroids obtained from the RBM block along with the Star IDs of the centroids appended as the fifth column.
2. **fe_output** : ((N,2) - Matrix) - The output matrix from Feature Extraction containing the centroids obtained from the image at the current frame.
3. **sm_TM_SNT** : ((5060, N) - Matrix) - The Star Neighbourhood Table
4. **sm_GD_SC** : ((5060, K) - Matrix) - The Guide Star catalogue
5. **sm_TM_CP_F** : (double) - The focal length of the star sensor (Units - cm)
6. **sm_TM_Nth** : (int) - The minimum number of matched stars required at every time step of the star tracker to achieve the desired accuracy of attitude (imposed by the Estimation block).

**Output:**

1. **sm_TM_SNT_output** : ((M,1) - Vector) - The column vector of the newly identified Star IDs as an output of the SNT block of the algorithm.

## sm_TM_calc_angdist.m()

**Code author:** Pranjal Gupta
**Created on:** 18/01/2021
**Last modified:** 18/01/2021
**Reviwed by:** NOT YET REVIEWED
**Description:**
This script calculates the cosine of the angular distance between given pair of centroids or given pair of Star IDs.
**Formula & References:**
**Input parameters:**

1. **is_id** : (boolean) - If True : The input variables id_centroid_1 and id_centroid_1 represent two Star IDs. Else, they represent two sets of centroids.
2. **id_centroid_1** : (int or (1,2) - Vector) - The first Star ID or the first centroid's (x,y) coordinates [depends on the value of is_id]

3. **id_centroid_2** : (int or (1,2) - Vector) - The second Star ID or the second centroid's (x,y) coordinates [depends on the value of is_id]
4. **sm_GD_SC** : ((5060, K) - Matrix) - The Guide Star catalogue
5. **sm_TM_CP_F** : (double) - The focal length of the star sensor (Units - cm)

**Output:**

1. **cos_ang_dist** : (float) - The calculated cosine of the angular distance.