11th ANKARA INTERNATIONAL AEROSPACE CONFERENCE **AIAC-2021-053**
8-10 September 2021 - METU, Ankara TURKEY

INVESTIGATION OF DIFFERENT APPROACHES FOR STAR IDENTIFICATION AND TRACKING IN LOW-COST STAR TRACKERS

Ahmet Emre Açıkgöz[1], Emre Oksal[2], Esra Kaplan [3]and Halil Ersin Söken[4]
Middle East Technical University
Ankara, Turkey

ABSTRACT

A star tracker can provide attitude measurement once the detected stars are successfully identified and matched with the catalog information. To increase efficiency and reliability, a technique for identifying stars and estimating attitude in the general lost-in-space case while also tracking the stars in successive measurement frames is presented in this study. First, the Pyramid algorithm is used for star identification. QUEST algorithm is utilized to estimate attitude statically. To increase the accuracy of estimation and estimate the angular velocities in addition, Multiplicative Extended Kalman Filter (MEKF) is used. After achieving successive initial estimations, the identification is bypassed and only QUEST and MEKF are used to estimate the attitude, while the initially identified stars are tracked. Comparison between algorithms with and without MEKF is done, and robustness of the algorithms is tested.

Keywords: Star sensor, Pyramid algorithm, Kosik algorithm, Attitude tracking, QUEST, Extended Kalman Filtering

INTRODUCTION

Star trackers are commonly used for spacecraft attitude determination for both Earth orbiting and interplanetary missions due to their high accuracy. A star sensor has two modes when solving the attitude using the detected stars: lost-in-space and tracking. In a lost-in-space scenario, no attitude information is available. Initially, the star tracker scans the whole field of view (FOV) to receive and map the stars and match them with the stars in the sensor's star catalog. Different algorithms can be used for the star identification process. In this paper, Kosik and Pyramid algorithms are presented and used as star identification algorithms. Pyramid algorithm [Mortari, Samaan, Bruccoleri, & Junkins, 2004], as a lost-in-space algorithm, uses angular distances between stars as a star identification feature and tries to match star patterns including four and more stars by starting from a triangular pattern. On the other hand, Kosik algorithm [Kosik, 1991] requires coarse attitude information of the spacecraft and uses angular separation and orientation differences of the star pairs as star identification requires patterns including two or more stars.

[1] Student in Aerospace Engineering Department: ahmet.e.acikgoz@gmail.com
[2] Student in Aerospace Engineering Department: emreoksal97@gmail.com
[3] Student in Aerospace Engineering Department: esrakaplan501@gmail.com
[4] Assistant Professor in Aerospace Engineering Department: esoken@metu.edu.tr

There are various other algorithms for star identification which are not used in this study but briefly discussed here. Liebe's algorithm [Liebe, Christian, 1995] requires at least three stars and selects one star and takes two closest stars to the first selected star. Then uses angular separations between the first and second selected stars and the angle between them. Baldini's algorithm [Baldini, Barni, Foggi, Bernelli & Mecocci, 1993] first identifies N number of brightest stars in the image and determines the angular separation of the five-star sequences. After that the algorithm makes a linear search through the cataloged stars, which fit in the acceptable range, by comparing the distances of each star to eliminate the stars having angular separation out of the tolerance of the observed angular separation. In the end it has reduced lists of possible matches for the stars and compares the combination of stars in these lists that could fit in the acceptable range for the identification., Padgett's Grid Algorithm [Padgett, Delgado, 1997] selects a star in the image and uses the location of the neighboring stars as points on a loose grid and this list goes on. According to the survey made by Spatling and Mortari [Spatling & Mortari, 2009] Pyramid algorithm requires less database search time and gives results with high validation compared to all these discussed algorithms. This is why Pyramid method is chosen to be the main Lost-In-Space algorithm in this paper. Kosik algorithm, on the other hand, gives robust results since it uses the coarse attitude knowledge to reduce the catalog search interval and is thought to be efficient where the tracking sequence is lost, but a coarse attitude information is available.

The star identification process is a critical stage of measurement processing for star trackers because it is time-consuming, and the star catalog requires large memory. To improve the situation, a method is designed and presented here. After sufficient number of stars are matched, attitude and angular rate information become available by processing the measurement and reference data. Attitude determination in terms of the quaternion is done by the QUEST method first. Once we have QUEST attitude estimates, the angular rate can be estimated as well using quaternion derivatives. After that the star sensor switches to tracking mode in which the already identified stars are tracked by using available attitude rate information. A new direction vector for stars is projected to the next frame by using the estimated angular velocity. To ensure the accuracy of the tracking mode, the Multiplicative Extended Kalman Filter (MEKF) can be further added to the scheme. Different approaches are tested in terms of accuracy and robustness.

METHOD

Figure 1 summarizes the investigated methods in this paper. The overall method is presented in five main sections:

1. Star Identification Process with Kosik and Pyramid Algorithms
2. QUEST
3. Star Tracking
4. Extended Kalman Filter
5. The Proposed Method

In Figure 1, Block 1 in both scenarios represent the star identification and tracking features. Block 2 is for the attitude estimation, such that the star tracker can autonomously provide the quaternion estimate. Scenarios 1 and 2 differs due to the used MEKF is scenario 2 for finer attitude and attitude rate estimation.

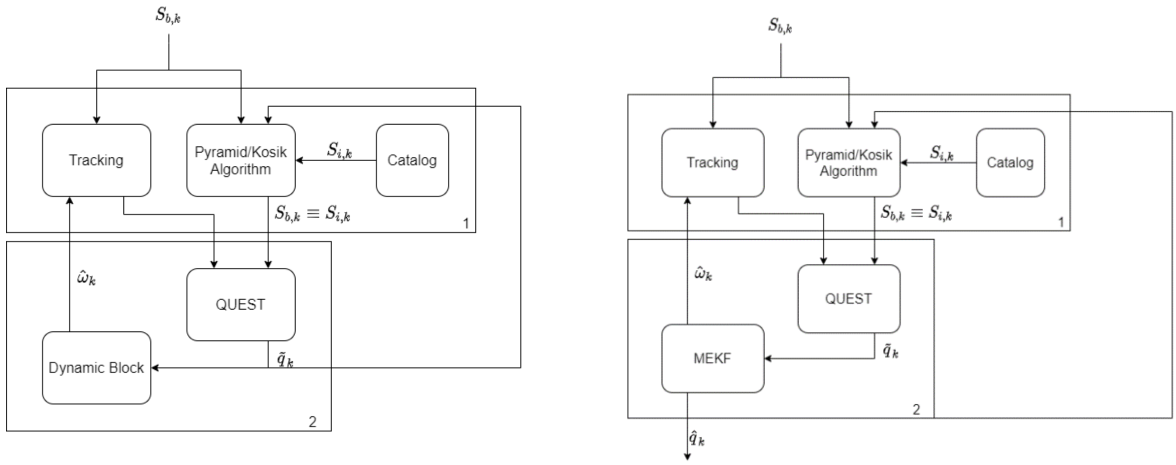

Figure 1: Flow Charts of the Proposed Method: Scenario 1 (Left) and Scenario 2 (Right)

Star Identification Process

Pyramid Algorithm

The pyramid Algorithm [Mortari, Samaan, Bruccoleri, & Junkins, 2004] is based on forming a pyramid star pattern with four stars or a triangle pattern for the cases that star tracker captures only three stars. After it forms the pattern, it looks at the inter-star angles between the stars of the pattern. The algorithm first looks for a star triangle and after that, it tries to identify the stars forming that triangle by using the 'k' vector approach. If it cannot find a unique solution it tries a different triangle that can be formed from the stars in the FOV. If it finds a solution, it seeks a reference star to form a pyramid. Then, it finds the inter-star angles of the pyramid and tries to match these angles with the catalog using the same approach. If it finds a unique solution that is consistent with the previous triangle solution the algorithm completes its mission for that time step. The key for the Pyramid star identification method is the 'k' vector range searching technique which makes the code able to know where to search in the catalog, therefore decreasing the operation time and cost.

Formation of the 'k-vector' is as follows:

The interstar angles between the star pairs which can fit in the field of view are found and sorted. Let the sorted vector is described as $s(N)$ where N is the number of elements and I is the integer vector that stores the indices of unsorted star angle vector y where

$$y\big(I(i)\big) = s(i) \quad (1)$$
$$y_{min} = \min\big(y(i)\big) = s(1)$$
$$y_{max} = \max\big(y(N)\big) = s(N)$$

This approach is about constructing a line that is slightly steeper than the line that connects $[1, y_{min}]$ and $[N, y_{max}]$ points. This new line connects $[1, y_{min} - \xi]$ and $[N, y_{max} + \xi]$ points where

$$\xi = \varepsilon\, max[|y_{min}|, |y_{max}|] \quad (2)$$

Here ε which is machine epsilon can be taken as $2.22\ 10^{-16}$ for double precision algorithms. Thus, the equation of the new line can be formed as:

$$z(x) = mx + q, \qquad where\ m = \frac{y_{max} - y_{min} + 2\xi}{N-1}\ \&\ q = y_{min} - m - \xi \quad (3)$$

Then, the k vector which starts from 0 ($ie. k(1) = 0$) is constructed considering the following condition:

$$k(i) = j \; if \; s(j) \leq z(i) \leq s(j+1) \tag{4}$$

Note that i varies from 1 to $N-1$. By looking at above relation it can be seen that the k vector gives the number of elements of 's' vector below the i^{th} element of 'z' vector.

Pyramid algorithm gets the array of interstar angles of a star pattern which can be triangle or pyramid. Then, it gets the range of the array which is $[y_{min}\,,\; y_{max}]$ and the usage algorithm of k vector is as follows:

$$[y_a\,, y_b] = [y_{min}\,, y_{max}] \tag{5}$$

$$j_b = \left\lfloor \frac{y_a - q}{m} \right\rfloor \;\&\; j_t = \left\lceil \frac{y_a - q}{m} \right\rceil \tag{6}$$

$$k_{start} = k(j_b) + 1 \;\&\; k_{end} = \; k(j_t) \tag{7}$$

Here $\lfloor x \rfloor$ is the maximum integer number that is smaller than x and $\lceil x \rceil$ is the minimum integer number that is greater than x. Note that in order to simulate the method an array which stores the suitable star pairs for different angular distances is formed [Spatling & Mortari, 2009].

The pyramid algorithm can be summarized as in Algorithm 1.

Algorithm 1: Pyramid Algorithm General Flow

1	**FOR** *# of Possible Triangles*
2	*Get a triangle*
3	*Identify*
4	**If** *it cannot be identified* **Goto** 2
5	*Save it as a reference triangle*
6	*Get another star*
7	**If** *it can be identified* **Update** *reference triangle as brightest 3 stars &* **Goto** 6
8	**If** *4 or more stars are identified* **Stop** *Identification*
9	**END**

Note that this approach is successful when there is not any error in the body vector measurements coming from the image processing. Therefore, two algorithms are developed in order to help the identification algorithm to overcome the failure due to errors. They can be separated as triangle match (Algorithm 2) and pyramid match (Algorithm 3) algorithms which are explained below.

Algorithm 2: Triangle Match Algorithm

```
1  FOR range of narrowed possible pairs for 1st Angle
2  Look at the angle of the pair
3  IF angle is in between ±0.45 deg range Hold it as a candidate 1st pair
4  ELSE Goto 22
5  END IF
6  FOR range of narrowed possible pairs for 2nd Angle
7  Look at the angle of the pair
8  IF angle is in between ±0.45 deg range AND the first star of the pair is same with that of 1st pair
9  Hold it as a candidate 2nd pair
10 ELSE Goto 21
11 END IF
12 FOR range of narrowed possible pairs for 3rd Angle
13 Look at the angle of the pair
14 IF angle is in between ±0.45 deg range
15   IF the 1st star of the pair is same with 2nd of the 1st pair AND the 2nd star of the pair is same with 2nd of 2nd pair
16     Save it as the 3rd pair
17     STOP matching
18   END IF
19 END IF
20 END
21 END
22 END
```

Algorithm 3: Pyramid Match Algorithm

```
1  FOR 1 range of narrowed possible pairs for 1st Angle
2  Look at the angle of the pair
3  IF angle is in between ±0.6 deg range AND the pair includes the first star of the triangle
4  Hold uncommon star as the candidate
5  ELSE Goto 21
6  END IF
7  FOR 2 range of narrowed possible pairs for 2nd Angle
8  Look at the angle of the pair
9  IF angle is in between ±0.6 deg range
10   IF the pair includes the second star of the ref triangle AND the pair includes the candidate
11     FOR 3 range of narrowed possible pairs for 3rd Angle
12     Look at the angle of the pair
13     IF the pair includes the third star of the ref triangle AND the pair includes the candidate
14       Save the candidate as the Match
15       STOP matching
16     END IF
17     END 3
18   END IF
19 END IF
20 END 2
21 END 1
```

Kosik Algorithm

Kosik Algorithm [Kosik, 1991] is a star identification algorithm that requires coarse attitude information about the spacecraft. The working principle of the Kosik Star Identification Algorithm is based on two different criteria which are angular separation and orientation differences of the stars. Two criteria depending on star body vectors are given with below relations.

$$\left\|\vec{V}_c^{12}\right\| - \left\|\vec{V}_m^{12}\right\| \leq \varepsilon \tag{8}$$

$$\frac{\vec{V}_c^{12} \bullet \vec{V}_m^{12}}{\left\|\vec{V}_m^{12}\right\|\left\|\vec{V}_c^{12}\right\|} \leq 1 - \Delta \tag{9}$$

In above relations, subscript c stands for catalogue, subscript m stands for the measured vector. Also, superscript 1 and 2 are defined as numbers of star body vectors that are examined at a given moment where:

$$\vec{V}_c^{12} = \overrightarrow{V_c^2} - \overrightarrow{V_c^1} \tag{10}$$

Actually, the first criteria given in (8) measures and checks angular separation values between stars in the field of view and stars in the catalogue. The second criteria given in (9) measures and checks orientation differences between the star pairs in the field of view and star pairs in the star catalogue. During this process, all-star pair combinations in the catalogue is evaluated by the algorithm. To have a better perception, orientation difference between star pairs in the field of view and star pairs in the catalogue is illustrated in below Figure (2).

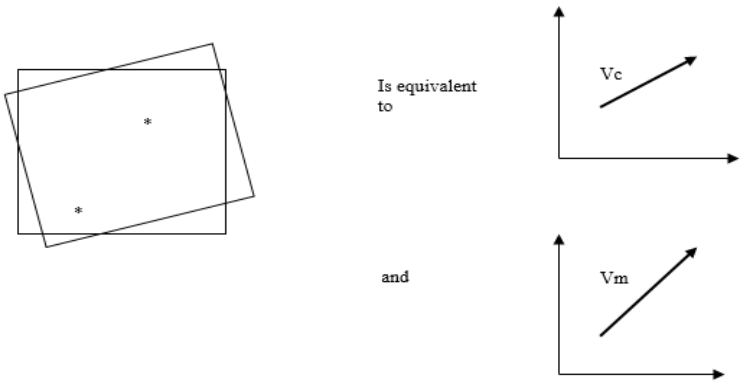

Figure 2: Orientation Criteria Demonstration of Star Pairs

As can be understood from inequalities (8) and (9), if the corresponding values are less than or equal to certain thresholds ε and $1 - \Delta$, in other words, if both conditions are satisfied simultaneously, Kosik Algorithm matches the stars in the field of view with the stars in the catalogue. Finally, once the stars in the field of view are matched, the matched stars can be used to determine attitude of the spacecraft.

Please note that if a pair satisfies both conditions (8) and (9), still there are two options while making star matching. This problem can be handled with the following statement. When the first measured star corresponds to the first star in the qualified pair from the catalogue, and the second measured star corresponds to the second star in the qualified pair, their orientation difference is near 0 degree. So, matching can be done directly as $V_m^1 = V_c^1$ and $V_m^2 = V_c^2$. On the other hand, if their orientation difference is near 180-degree, matching can be done as $V_m^1 = V_c^2$ and $V_m^2 = V_c^1$.

Coarse attitude information can be used in Kosik Algorithm in different ways. In this study, the narrowed star catalogue is generated in a way that it includes all the stars that that are making an angle of 35-degree or less with the -z axis of the spacecraft's body frame, which is actually the boresight axis of the star tracker. During this process, quaternion values estimated by the QUEST or the MEKF in the previous recursive step is used. Thus, the coarse attitude is used to map all the catalog stars to the body frame and these mapped star vectors are used in the identification (e.g. $\overrightarrow{V_c^1}, \overrightarrow{V_c^2}$) in Eq.10.

In order to improve the accuracy of Kosik Algorithm, following method is applied. Since the algorithm runs by considering stars in the FOV by pairs, first, it tries to match a selected pair with a pair in the catalogue according to (8) and (9). If no match is achieved, it constructs another star pair which consists of another star in the field of view. This generally corresponds to the brightest star in the field of view, and the star coming from the unmatched pair in the first try. Then, the algorithm tries to match this newly created pair. After matching this newly created star pair, the algorithm again constructs a new pair which includes other stars coming from the unmatched pair in the first try. Then, it tries to match this newly created star pair. In brief, both two stars in the first unmatched star pair could be matched by constructing two additional pairs having another star in the field of view. Below, in Algorithm 4, the general star identification process of Kosik Algorithm used in this study is given.

	Algorithm 4: Kosik Algorithm General Flow
1.	*Generate narrowed catalogue*
2.	*Construct a pair in the FOV*
3.	*Try to identify the pair*
4.	**If** *it can be identified*
5.	Pass to *next pair in the FOV* **(Goto 2)**
6.	**Else**
7.	*Construct two additional pairs*
8.	*Try to identify these both pairs*
9.	*Pass to next pair in the FOV* **(Goto 2)**
10.	**END**

Quest

QUEST method [J., Damaren, & Forbes, 2013] is a very frequently used attitude determination algorithm that takes measurements in the body frame as input and gives estimated quaternion information of the spacecraft as output. It is based on Wahba's Cost Function J given in equation (12), which is needed to be minimized.

$$J = \sum_{k=1}^{N} w_k (1 - S_{b,k}^T A S_{i,k}) \tag{12}$$

Here w_k is the weight of the sensors, $S_{i,k}$ is star vector in Earth Centered Inertial (ECI) frame, $S_{b,k}$ is the measured star vector in spacecraft's body frame and $A$is the direction cosine matrix from ECI frame to the spacecraft's body frame.

Minimizing J, equals to maximizing g function defined as,

$$g(\bar{q}) = \bar{q}^T K \bar{q} \tag{13}$$

Where,

$$[K] = \begin{bmatrix} \sigma & Z^T \\ Z & S - \sigma I_{3x3} \end{bmatrix}, \quad \sigma = tr([B])$$

$$[B] = \sum_{k=1}^{N} w_k \, S_{b,k} S_{i,k}^T, \quad [S] = B + B^T, \quad [Z] = [B_{23} - B_{32} \quad B_{31} - B_{13} \quad B_{12} - B_{21}]^T$$

After some mathematical manipulations, it is seen that desired quaternion set corresponds to the largest eigenvector of matrix K.

In QUEST algorithm, largest eigenvalue of K matrix, firstly, assumed as sum of individual sensor weights,

$$\lambda_0 = \sum_{k=1}^{N} w_k \tag{14}$$

Then by using Newton-Raphson Iteration Method λ_{max} can be found as follows,

$$\lambda_{max} = \lambda_i = \lambda_{i-1} - \frac{f(\lambda_{i-1})}{f'(\lambda_{i-1})} \tag{15}$$

Where,

$$f(s) = \det(K - s[I_{4x4}]) \tag{16}$$

Therefore, estimated Rodrigues parameter vector R can be found as,

$$\bar{R} = ((\lambda_{opt} + \sigma) I_{3x3} - [S])^{-1}[Z] \tag{17}$$

At the end, quaternion vector estimated by QUEST method can be defined as,

$$\bar{q} = \frac{1}{\sqrt{1 + \bar{R}^T \bar{R}}} [\frac{1}{\bar{R}}] \tag{18}$$

Knowing that A is the direction cosine matrix, error of QUEST can be determined as

$$v_{err} = 2sin^{-1}\left(\frac{\|A - A^{true}\|_F}{\sqrt{8}}\right) \, where \, \|C\|_F^2 = trace(CC^T) \tag{19}$$

In calculation of three axis error, firstly quaternions of the QUEST is converted into Euler angles via following relations [D. Baldini,1985]:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} atan2(2(q_0 q_1 + q_2 q_3), 1 - 2(q_1^2 + q_2^2)) \\ \operatorname{asin}(2(q_0 q_2 - q_3 q_1)) \\ atan2(2(q_0 q_3 + q_1 q_2), 1 - 2(q_2^2 + q_3^2)) \end{bmatrix} \tag{20}$$

Taking the differences between the actual and simulated Euler angles gives the three-axis angle errors.

Tracking

The tracking mode of the star sensors becomes enabled after a certain period of a lost-in-space star identification algorithm operation. Once the system gets stable quaternion information provided by the star identification algorithms and the QUEST, it switches to tracking mode in order to reduce the computational load. With the knowledge of previous quaternion information, the angular rate of the spacecraft is estimated by using kinematic equations. Then, the body vectors of the stars in the previous time step which are matched with the catalog already, are projected into the next time step. The angular separation between these projected vectors and the current measured body vectors are compared with each other and they are stored if angular separation values are small enough. Thus, stars become matched without the need for a star identification algorithm which results in a significant reduction in computational load. For the cases of high angular rates, high measurement errors which cause the projection process to be flawed, the star identification algorithms are needed to operate again until reaching stable quaternion information.

The vector projection algorithm is as follows [Markley & Crassidis, 2014]:

First the angular velocity of the previous time step is calculated based on quaternion differentiation

$$\omega_{k-1} = 2\Xi^{T}(q_{k-1})\dot{q}_{k-1} \tag{21}$$

where

$$\Xi(q) = \begin{bmatrix} q(4) & -q(3) & q(2) \\ q(3) & q(4) & -q(1) \\ -q(2) & q(1) & q(4) \\ -q(1) & -q(2) & -q(3) \end{bmatrix} \;\&\; \dot{q}_{k-1} = \frac{q_{k-1}-q_{k-2}}{\Delta t} \tag{22}$$

Here Δt is timestep and subscript k describes the discrete time.

Angular separation between the body frames of the spacecraft in the current time and previous time is found by Euler integration.

$$\begin{bmatrix} \phi_{k,k-1} \\ \theta_{k,k-1} \\ \psi_{k,k-1} \end{bmatrix} = \begin{bmatrix} \omega_{k-1}\,(1) \\ \omega_{k-1}\,(2) \\ \omega_{k-1}\,(3) \end{bmatrix} \Delta t \tag{23}$$

After that the previous body vectors of the stars can be projected by following multiplication

$$b_k = C_{k,k-1}\big(\psi_{k,k-1}, \theta_{k,k-1}, \phi_{k,k-1}\big) b_{k-1} \tag{24}$$

Note that $C_{k,k-1}(\psi_{k,k-1}, \theta_{k,k-1}, \phi_{k,k-1})$ is the direction cosine matrix that projects the previous body frame to the current time by using the previous angular rate in Z-Y-X sequence.

Multiplicative Extended Kalman Filter

The MEKF is a very compelling attitude filtering algorithm due to its flexibility and computational efficiency. After the star identification is completed and an initial attitude and angular velocity are established, the star tracker enters tracking mode. In the tracking mode, as discussed in previous section the algorithm can use the QUEST estimated quaternions and calculated angular rates (recall scenario 1 in Fig.1). Nonetheless, to provide finer attitude and attitude rates for the tracker the MEKF block can be added into the algorithm (Scenario 2 in Fig.1).

As a result, recursively QUEST and MEKF algorithms perform attitude calculation. Kalman filter integrates mathematical model of the spacecraft and measurements coming from the QUEST to estimate the attitude and attitude rate. MEKF is used when nonlinearity is present. In our problem, the system model is nonlinear. There are two steps for filtering [Spratling & Mortari,2009]:

1. Prediction: Mathematical model is used to propagate states to the time of measurements
2. Update: Predicted states are updated by using the measurements

State is chosen as:

$$x = [\widehat{\delta a_k} \ \ \widehat{\delta \beta_k}]$$

The following scheme summarizes the whole process.

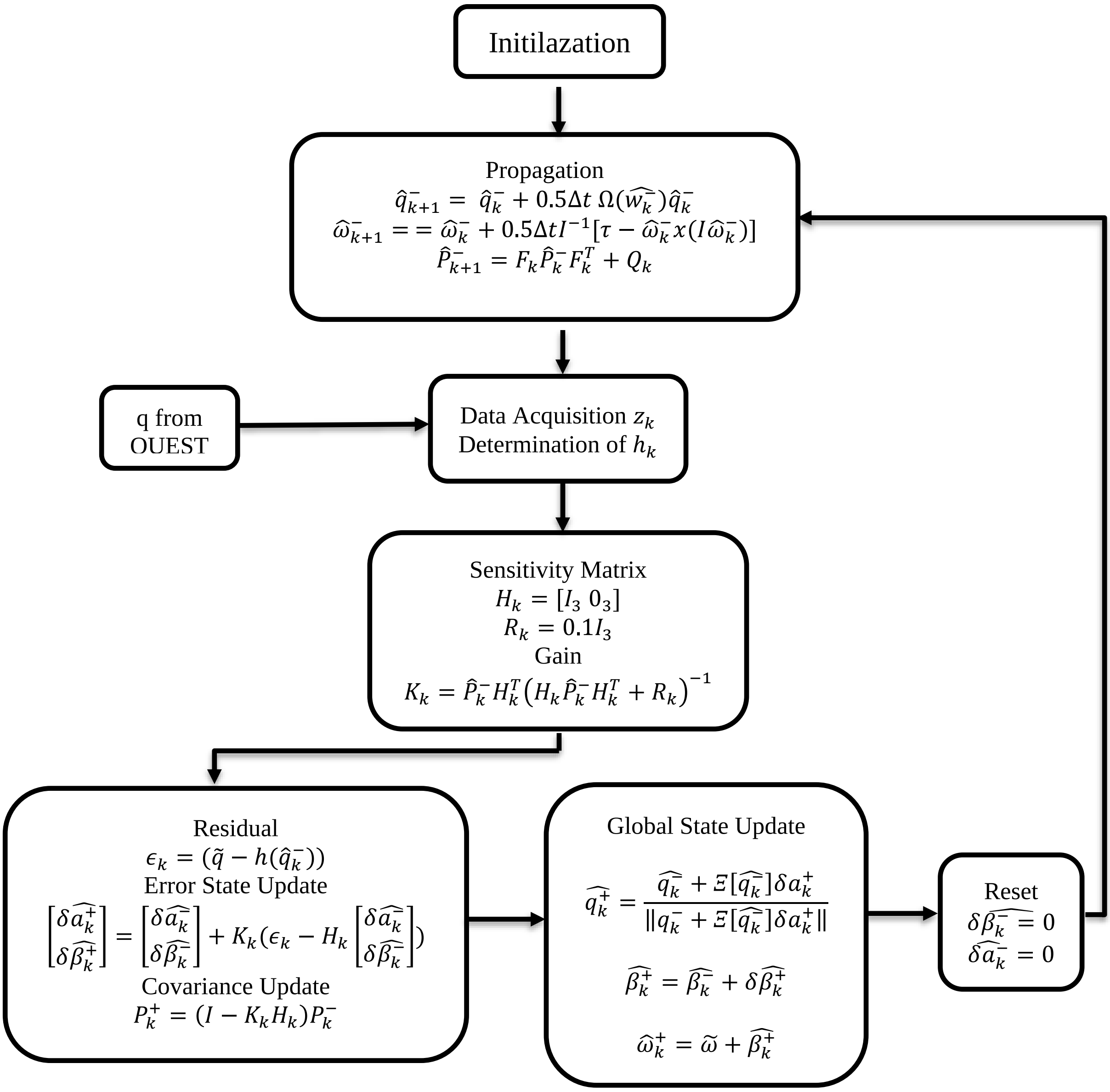

Figure 3: Multiplicative Extended Kalman Filter General Structure

Proposed Method

In this study, two different approaches for the integration of the introduced methods are compared and discussed. As it is shown in Fig. 1the main difference between these two approaches is as follows:

- In the first approach, in absence of MEKF, tracking and Kosik blocks are feedbacked with the quaternion output of the QUEST block and the calculated angular velocity
- In the second approach, those blocks are feedbacked with the angular velocity and quaternion information coming from the MEKF block,

In order to reduce the computational load, it is best to reduce the operation time of star ID algorithms by operating in tracking mode as much as possible. But tracking can be lost in time. Therefore, previously explained methods are combined. Note that, when they are called, Pyramid method runs for 10 seconds, Kosik algorithm runs for 3 seconds and the tracking function runs until the tracking is lost.

The more detailed explanation of the main algorithm is explained in the following flow chart.

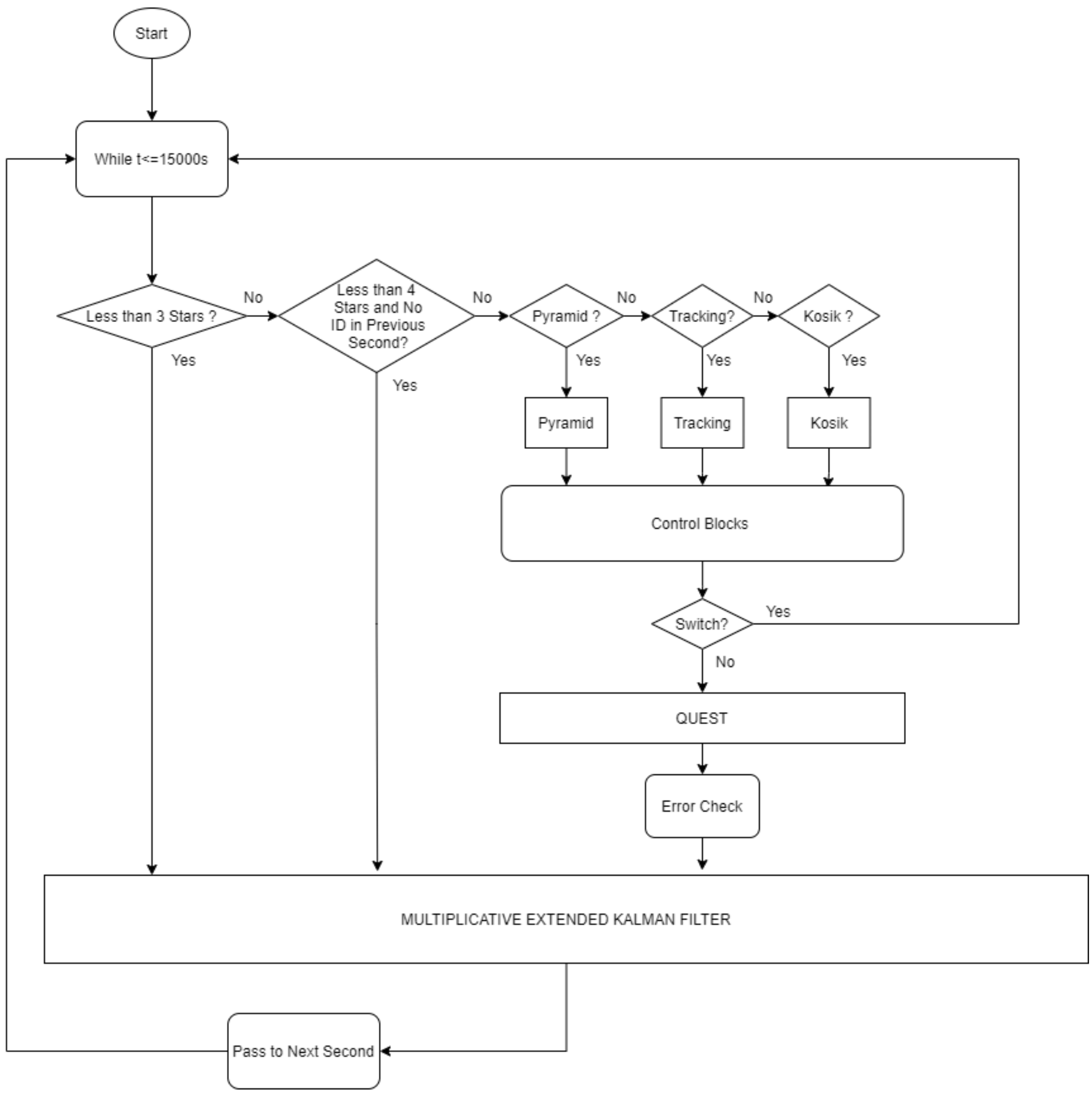

Figure 4: General Flow of Star Identification and Tracking Process

It should be noticed that the Pyramid algorithm may not match triangles correctly and, when there are 3 stars in the view the success rate just crashes to the zero. Also, in some rare cases where there are 4 stars in the FOV it may continue matching based on a wrong triangle. Thus, in these cases, the attitude error rapidly increases. As it can be seen from the Figure 4 the first two conditional operators are placed in the algorithm in order to prevent the failure.

Control and Error Check blocks shown in Fig.4 are explained below in detail (Algorithm 5 and Algorithm 6).

Algorithm 5: Control Blocks Pseudo Code

Pyramid Control Block

If *run_time*==10 sec
  switch to tracking
end

Tracking Control Block

If *less than 60% match or # of matches*==2
  switch to kosik
end

Kosik Control Block

If *less than 70% match*
  switch to pyramid
elseif *run_time==3 sec*
  switch to tracking
end

Algorithm 6: Error Check Block Pseudo Code

calculate angular error between t and t-1
If error > 4 deg
  switch to MEKF (dynamic model)
else
  continue to MEKF (with quest results)
end

Note that, error check is done under the assumption of that the motion of the spacecraft is slow so that the attitude does not change significantly. Also, it should be stated that the angular error is calculated in terms of Frobenius Norm as it is shown in Equation (19).

Results and Discussions

As can be seen from Figure 5, adding MEKF block to the scheme decreases the attitude estimation error significantly. From scenario 1 to scenario 2, the maximum error angle dropped from 4 degrees to 2 degrees. It is necessary to notice that there is a distinct spike at nearly 3200th second of both graphs. This situation is examined, and it is concluded that a new relatively bright star entering into the FOV of the spacecraft causes tracking mode to reduce its success rate by shifting star order which is sent to QUEST. In this period, the algorithm runs in tracking mode without calling any of the star identification algorithms. Therefore, error in the outputs of the attitude estimations is getting larger since the matched star orders are faulty in the tracking mode. At a certain time, tracking mode requirements are not satisfied and Pyramid Algorithm is called. After calling star identification algorithm, the error can be again reduced as shown in Figure 5.

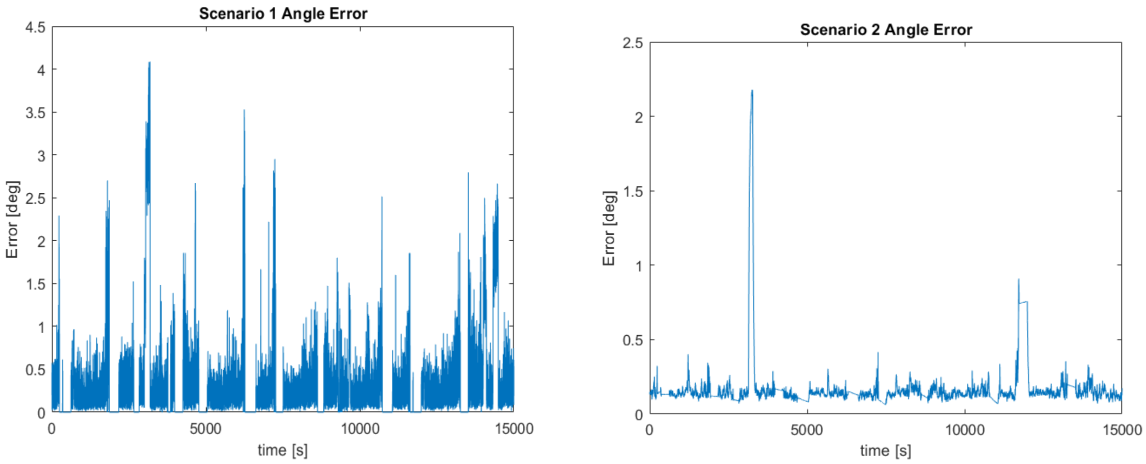

Figure 5: Angle Errors Comparison

Figure 6 shows that with the MEKF scenario, in every axis, the attitude estimation error is smaller. Also, the process is noisier in scenario 1. There are some gaps in the left figure. It means that no successful star matching is done, so there is no attitude information available. In the right figure, there is no gaps since, in no matching case, the mathematical model is used to estimate the attitude in the MEKF. Therefore, there is always attitude information available in scenario 2.

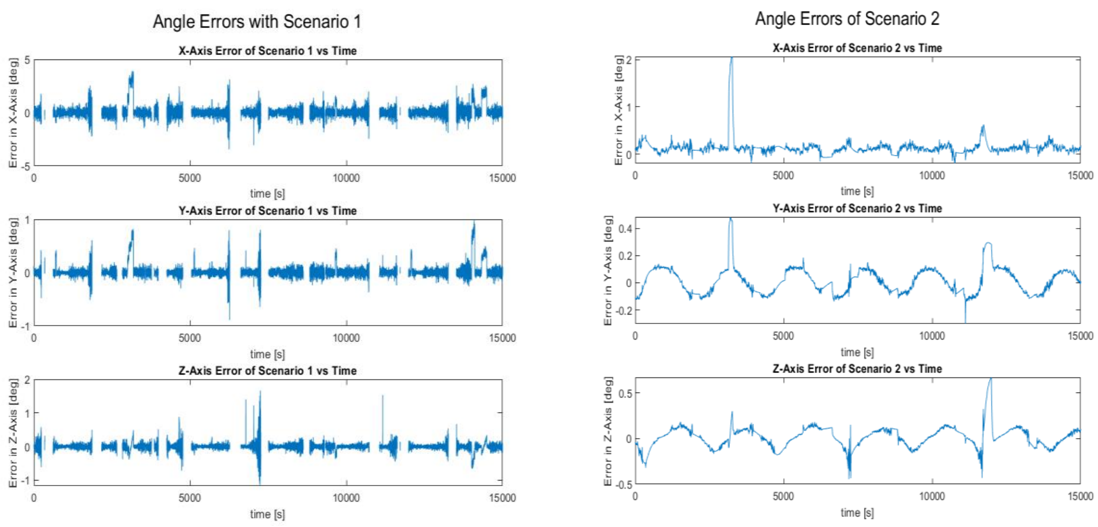

Figure 6: Angle Errors Comparison in Three-Axis

Next we check the number of referrals to each algorithm in the proposed schemes. In Figure 7, operation type 0 corresponds to time steps when there is no star identification, operation type 1 corresponds to the calls for Pyramid Algorithm, operation type 2 corresponds to the calls for tracking and operation type 3 corresponds to the calls for Kosik Algorithm. In both cases the majority of the time the tracking algorithm is running and this is a desired situation for reducing the load.

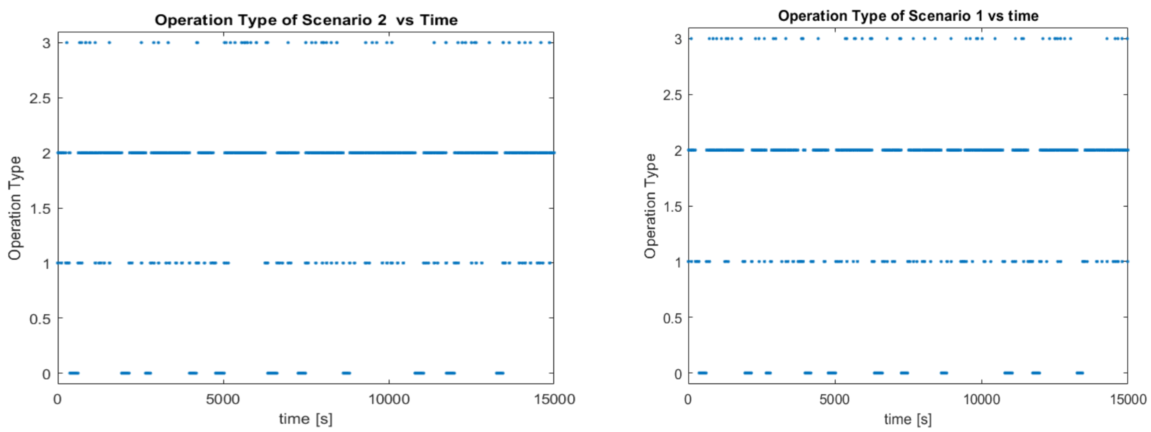

Figure 7: Operation Type Comparison

Table 1 further investigates the computational times. It can be seen that MEKF has increased the performance in such a way that the Pyramid operation time is decreased and tracking is increased. Also it is seen that Kosik has taken place more in MEKF case. These results are expected since the MEKF is reducing the error in attitude and angular velocity estimates which are necessary information for the Kosik and Tracking blocks. Accurate measurements led the Kosik and Tracking Blocks operate in success therefore their operation time is increasing when the MEKF is included in the block.

Table 1. Performance Comparison

Performance of The Methods		
	with MEKF	without MEKF
Pyramid Operation Time	1150s/15000s	1770s/15000s
Kosik Operation Time	141s/15000s	123s/15000s
Tracking Operation Time	11252s/15000s	10665s/15000s
No Identification	2442s/15000s	2442s/15000s

Figure 8 shows the Runtime comparison of the proposed method with the pyramid method in this simulation. It can be clearly observed that the runtime for the proposed method is almost 10% of the pyramid runtime thanks to the tracking block.

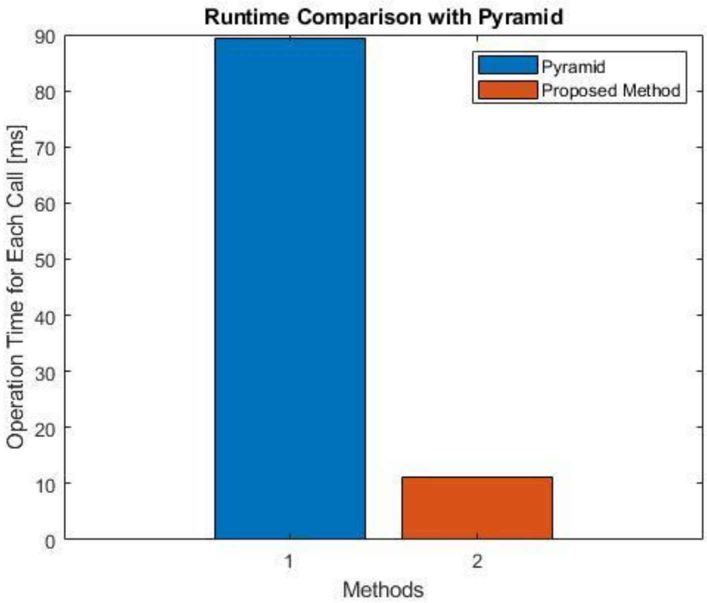

Figure 8: Runtime Comparison

CONCLUSIONS

In this study, a combined method is proposed for star identification and spacecraft attitude determination. The results are presented and discussed. It can be observed that the purpose of reduced usage of star identification algorithms by replacing them with tracking is achieved. Thus, computational load is reduced compared to computational effort consumed by Pyramid Method only. As it is proposed, there are two scenarios for comparison, which are with MEKF and without MEKF scenarios. Without MEKF, Kosik and Tracking blocks are sent information from QUEST while this information is provided by MEKF in the other scenario. It is observed that MEKF increased the accuracy of the attitude estimation, therefore, increased the tracking performance. It should be noted that, throughout the operation, error in attitude estimation does not exceed 0.5 degrees (2^{nd} Scenario) except two limited periods of time. The reason of these exceptions is understood that there are multiple mismatches done by tracking in those periods, causing attitude estimation not to be proper. Overall, proposed method is investigated and compared with the approaches in the literature and shown to be successful in terms of both accuracy and computational effort.

References

B. Spratling and D. Mortari, “A Survey on Star Identification Algorithms,” Algorithms, vol. 2, no. 1, pp. 93–107, 2009.

C. C. Liebe, “Star trackers for attitude determination,” IEEE Aerospace and Electronic Systems Magazine, vol. 10, no. 6, pp. 10–16, 1995.

C. Padgett and K. Kreutz-Delgado, “A grid algorithm for autonomous star identification,” IEEE Transactions on Aerospace and Electronic Systems, vol. 33, no. 1, pp. 202–213, 1997.

D. Baldini, M. Barni, A. Forgi, G. Benelli, and A. Mecocci, “A new star-constellation matching algorithm for satellite attitude determination,” ESA Journal, vol. 17, pp. 185–198, 1993

D. Mortari, M. A. Samaan, C. Bruccoleri, and J. L. Junking, “The Pyramid Star Identification Technique,” Navigation, vol. 51, no. 3, pp. 171–183, 2004.

D. R. A. H. J., C. Damaren, and J. R. Forbes, Spacecraft dynamics and control: an introduction. Chichester: John Wiley, 2013.

J. C. Kosik, “Star pattern identification aboard an inertially stabilized aircraft,” Journal of Guidance, Control, and Dynamics, vol. 14, no. 2, pp. 230–235, 1991.