Extended Kalman Filter Algorithm: Line-By-Line

**Lines 1-17: Set Up**

A computer screen shot of a code

AI-generated content may be incorrect.

Define function, with state ‘XREF’ and time ‘tk’ reference trajectory input.

Import necessary functions and packages, noting that ‘NRHOmotion’ and ‘STM’ functions are explained in *CR3BP: Line-By-Line*.

Empty arrays set up to store results, ‘filter\_results’ to store state estimates, and ‘covariance\_results’ to store corresponding covariance determined by the EKF. 'residual\_results’ stores the difference between the state estimate and the actual read for later analysis.

The rest of the function reads XREF state by state, storing the resulting estimates in filter\_results.

**Lines 18-33: Initialisation**

**A computer screen shot of a program code

AI-generated content may be incorrect.**

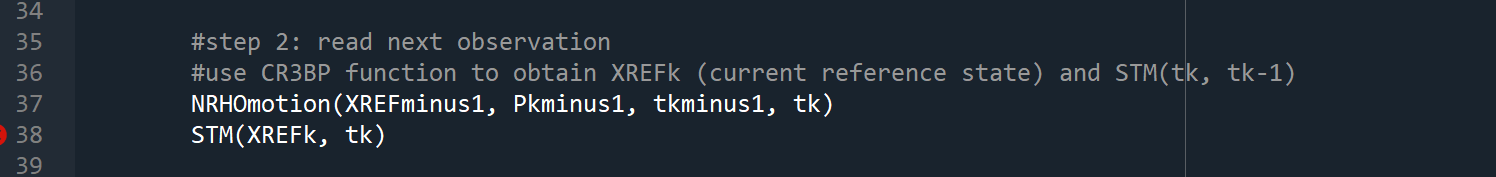
The first step in the algorithm is to set up the ‘a priori’ state. The filter takes this a priori ‘ith’ reference state provided in XREF array ‘XREFminus1’ (each row provides x, y, z coordinate) and corresponding time from tk ‘tkminus1’ and uses this to determine the next state at the next time read tk. How close the resulting state is to the corresponding truth state is the point of comparison for this experiment.

As the covariance is determined and carried on from the previous state, it is set to zero for the first read.

Below is a table outlining the naming conventions for the equations and the code.

|  |  |  |
| --- | --- | --- |
| Mathematical Notation | Python Variable Name | Definition |
|  | XREFminus1 | Reference state used to propagate from, most recent measurement of state |
|  | Pkminus1 | Last covariance (metric of how much we trust the estimate vs the measurement) determined |
|  | tkminus1 | Time at most recent measurement of state |
|  | tk | Time at which the future state estimate is being made to |

**Lines 34-39: Obtain Next Observation**

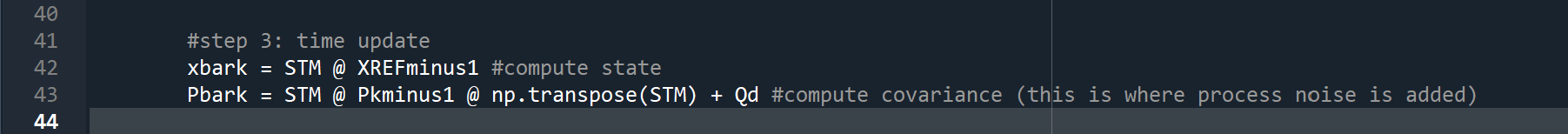


Line 37 uses the NRHOmotion function to propagate the a priori to obtain the next observation ‘XREFk’.

This result is then used to obtain the state transition matrix (STM) via the STM function to allow for a computation of a state estimate.

|  |  |  |
| --- | --- | --- |
| Mathematical Notation | Python Variable Name | Definition |
|  | XREFk | State estimate from propagation via equations of motion |
|  | STM | State transition matrix, used to transition between state between two times via matrix multiplication. |

**Lines 40-44: Time Update**



Then the state estimate is determined via the STM (line 42):

Then the estimated covariance is also determined (line 43):

|  |  |  |
| --- | --- | --- |
| Mathematical Notation | Python Variable Name | Definition |
|  | xbark | Resulting state estimate from time update. |
|  | Pbark | Resulting covariance from STM and time update. |
|  | Qd | Process noise matrix, an estimate of the errors accompanied with obtaining the results. |

**Lines 45-53: Measurement Update**

A computer screen shot of text

AI-generated content may be incorrect.

State deviation is updated in the measurement update and thus the final state and covariance estimate is produced.

|  |  |  |
| --- | --- | --- |
| Mathematical Notation | Python Variable Name | Definition |
|  | Hk | Observation mapping matrix, note that as the coordinates are in x, y, z already this is just an identity matrix. |
|  | epsilon\_k | Measurement residual, resulting from mapping between measurement and coordinates. Is zero for this experiment. |
|  | yk | Actual measured state at time k. |
|  | Rk | Measurement noise matrix. |
|  | Kk | Kalman gain, weighting factor determining how much a measurement is trusted versus the model prediction. |
|  | xxk | State deviation from reference state initially propagated from equations of motion. |
|  | Xk | Final state estimate for this iteration of the filter. |
|  | Pk | Final covariance estimate for this iteration of the filter. |

Kalman gain determined:

State deviation:

State estimate:

Covariance:

**Lines 54-65: EKF Update**

**A computer screen shot of text

AI-generated content may be incorrect.**

Essentially a condition is set that if the estimated propagated state and the state estimate converge, the state deviation is then reset to zero to ensure the filter continues to accept new estimate updates.

These results are then added to the initialised empty matrices for storage. Filter outputs all recorded results after all iterations are completed.