NavFuse Kalman Filter Class Design Description

Parker Barrett

February 6th, 2023

1 Class Overview

- Header File: NavFuse/include/filter/KalmanFilter.hpp
- Implementation: NavFuse/src/filter/KalmanFilter.cpp

The NavFuse Standard Kalman Filter class contains functions for linear and extended Kalman Filter initialization, prediction and measurement updates. The Kalman Filter class is inherited from the BaseFilter class.

2 Inherited Class Members

2.1 KalmanFilter::filterInitialize()

- Inputs
  - Eigen::VectorXd x0: nx1 dimensional vector containing the initial filter state
  - Eigen::MatrixXd P0: nxn dimensional matrix containing the initial filter covariance
- Outputs
  - No Outputs
- Algorithm
  - The private class member KalmanFilter::filterState is initialized to the value of the input state vector, x0
  - The private class member KalmanFilter::filterCovariance is initialized to the value of the input covariance matrix, P0

2.2 KalmanFilter::filterCovariance

- Data Type: Eigen::MatrixXd
- Description: nxn dimensional matrix which stores the most up to date filter covariance estimate

2.3 KalmanFilter::filterState

- Data Type: Eigen::VectorXd
- Description: nx1 dimensional vector which stores the most up to date filter state estimate

2.4 KalmanFilter::getCovariance()

- Inputs
  - No Inputs
- Outputs
  - Eigen::MatrixXd Pk: nxn dimensional filter covariance matrix
- Algorithm
  - Return the current filter covariance matrix

2.5 KalmanFilter::getState()

- Inputs
  - No Inputs
- Outputs
  - Eigen::VectorXd xk: nx1 dimensional filter state vector
- Algorithm
  - Return the current filter state vector

3 Public Class Members

The following sub-sections describe the inputs, outputs and internal algorithms used in the public interface of the Kalman Filter class.

3.1 KalmanFilter::filterPredict()

- Inputs
  - Eigen::MatrixXd Phik: nxn dimensional discrete time state transition matrix
  - Eigen::MatrixXd Qk: nxn dimensional discrete time process noise matrix
- Outputs
  - No Outputs
- Algorithm
  - The state vector mean is propagated forward in time via $\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k$
  - The covariance matrix is propagated forward in time via $\mathbf{P}_{k+1} = \Phi_k \mathbf{P}_k \Phi_k + \mathbf{Q}_k$

3.2 KalmanFilter::filterUpdate()

- Inputs
  - Eigen::MatrixXd zk: mx1 dimensional measurement vector
  - Eigen::MatrixXd Hk: mxn dimensional discrete time measurement Jacobian matrix
  - Eigen::MatrixXd Rk: mxm dimensional discrete time measurement noise matrix
- Outputs

  - No Outputs

- Algorithm
  - The measurement residual is computed as the difference between the actual and predicted measurements: $\nu = \mathbf{z}_k - \mathbf{H}_k\mathbf{x}_k$
  - The optimal Kalman Filter is computed by: $\mathbf{K} = \mathbf{P}_k\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k\mathbf{H}_k^T + \mathbf{R}_k)^{-1}$
  - The state is updated by: $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{K}\nu$
  - The covariance is updated using the stable Joseph Form by: $\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}\mathbf{H}_k)\mathbf{P}_k(\mathbf{I} - \mathbf{K}\mathbf{H}_k)^T + \mathbf{K}\mathbf{R}_k\mathbf{K}^T$