Dokumentation Kalman Filter

version

Philipp Pelcz

February 11, 2021

Inhalt

Dokumentation zur Micropython Kalman Filter Library

<iframe id="ID" style="border:1px solid #666CCC" title="PDF" src="../../Documents/document.pdf" frameborder="1" scrolling="auto" height="1600" width="1300" align="middle"> </iframe>

class `Kalman_Filter.Kalman_Filter.`**`Kalman_Filter`**

Die Klasse des Kalman Filters (KF)

`computePrediction` **(**delta_t=None**)**

Funktion um eine Vorhersage des Kalman Filters zu generieren. Dabei wird die P-Matrix nicht geupdated
Parameters:
delta_t: Optionaler Parameter für die Zeitdauer zwischen 2 Funktionsaufrufen. Falls delta_t im Methodenkopf nicht genutzt wird, wird dt aus der Konfigurationsdatei genutzt.
Returns:
Der nächste erwartete Zustand.

`configure` **(**path_to_config_file: str**,** x0=None**)**

Funktion um die Konfiguration des Kalman Filters vorzunehmen.
Hier werden alle wichtigen Parameter gesetzt. Daher kann der Kalman Filter nicht ohne die Konfiguration gestarted werden!
Parameters:
path_to_config_file: String der den Pfad zur Konfigurationsdatei angibt. Die Datei muss eine .json Datei sein
x0: Initialer Zustandsvektor des Kalman Filters. x0 muss vom Typ ulab.ndarray() sein
Folgende Parameter können/müssen in der .json Datei gesetzt werden
dt: Zeiteinheit zwischen 2 KF Updates. Als Default Wert empfielt sich eine 1
n: Dimension des Zustandraums. Muss ein Integer sein
m: Dimension des Beobachtungs- oder Messraums. Muss ein Integer sein
A: Koeffizienten der Zustandsübergangsmatrix. Muss eine nXn Matrix sein
At: Optionale Matrix für die Zeitabhängigkeiten in der Zustandsübergangsmatrix. Muss eine nXn Matrix sein.
C: Beobachtungsmatrix für die Überführung vom Zustand zur Messung. Muss eine mxn Matrix sein.
Q: Zustandskovarianzmatrix für die Modelierung der Zustandsunsicherheit. Dieser Teil ist Zeitunabhängig. Muss eine nxn Matrix sein.
Q_coeff: Matrix für Koeffizienten der zeitabhängigen Anteile in der Zustandskovarianzmatrix. Muss eine nxn Matrix sein.
Q_variance: Skalarer Wert für die Varianz der Zustandsunsicherheit. Wird im Programm mit der Zustandskovarianzmatrix multipliziert. Muss ein Skalar sein.
Q_exponent: Matrix für die Exponenten von t für die Zustandsunsicherheit. Muss eine nxn Matrix sein.
R: Sensorkovarianzmatrix für die Modelierung des Sensorrauschens. Muss eine mxm Matrix sein.
P: Fehlermatrix. Sollte als Identitätsmatrix oder Nullmatrix gewählt werden und hält die aktuelle Fehlermatrix. Wird bei Berechnungen überschrieben und ist daher ein optionaler Parameter. Muss beim Setzen aber eine nxn Matrix sein.
P0: Initialer Wert der Fehlermatrix, falls man diese zurücksetzen möchte. Ist ein optionaler Parameter, muss aber beim Setzen eine nxn Matrix sein.
x0: Anfangszustand für den Kalman Filter. Wird der x0 Wert des Methodenkopfes nicht genutzt muss dieser in der .json Datei gesetzt werden.

`factorial` **(**x: int**)**

Private Funktion um die Fakultät auszurechnen. Sollte eigentlich nicht benutzt werden
Parameters:
x: Integer Wert, dessen Fakultät ausgerechnet werden soll
Returns:
int: x! wird zurückgegeben

`getCurrentState` **()**

Funktion für das Rücksetzen der Fehlermatrix und des geschätzten Zustandes.
Diese Funktion kann genutzt werden, falls der Kalman Filter wegen schlechten Eingabewerten, einer ungenauen Systemgleichung o.ä. von der gewünschten Lösung divergiert.
Returns:
Den aktuellen Zustand

`getErrorCovarianceMatrix` ()
Getter Funktion für die Fehlermatrix.
Returns:
Die P Matrix

`likelihood` (data_in**)**
Funktion um die Likelihood für eine Messung zu berechnen.
Parameters:
data_in: Messung deren Likelihood berechnet werden soll.
Returns:
Likelihood der gegebenen Messung

`predict` ()
Funktion um eine Vorhersage des Kalman Filters zu generieren. Dabei wird die P-Matrix geupdated
Parameters:
Returns:
Der nächste erwartete Zustand.

`resetErrorCovAndState` ()
Funktion für das Rücksetzen der Fehlermatrix und des geschätzten Zustandes.
Diese Funktion kann genutzt werden, falls der Kalman Filter wegen schlechten Eingabewerten, einer ungenauen Systemgleichung o.ä. von der gewünschten Lösung divergiert.
Parameters:

`update` (data_in**,** delta_t=None**,** update_Q_Matrix=False**,** R_loc=None**)**
Funktion für einen Update-Schritt des Kalman Filters. In dieser Funktion wird sowohl die A-, Q-, P- und die Gatingmatrix und der Zustand x_hat geupdated.
Parameters:
data_in: Aktuelle Messung. Muss ein mx1 Vektor sein.
delta_t: Optionaler Parameter für die vergangene Zeit zwischen dem letzten Methodenaufruf. Falls dieser Wert nicht genutzt wird, wird stattdessen der dt Wert aus der Konfigurationsdatei verwendet.
update_Q_Matrix: Optionaler Parameter für das Überschreiben der Q Matrix.
R_Loc: Optionaler Parameter für das Setzen der R Matrix. Dieser Parameter sollte genutzt werden, falls die R Matrix nicht konstant ist. Ansonsten sollte die R Matrix in der Konfigurationsdatei gesetzt werden.
Returns:
Der gefilterte geschätzte Zustand

Stichwortverzeichnis

Python-Modulindex

k