

Sample rst2pdf doc

version

Your Name

February 09, 2021

Inhalt

Dokumentation zur Micropython Kalman Filter Library	1
Stichwortverzeichnis	3
Python-Modulindex	5

Dokumentation zur Micropython Kalman Filter Library

`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:

self: Object Pointer des KF

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 gestartet werden!

Parameters:

self: Object Pointer des KF

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 empfiehlt sich eine 1

n: Dimension des Zustandsraums. Muss ein Integer sein

m: Dimension des Beobachtungs- oder Messraums. Muss ein Integer sein

A: Koeffizienten der Zustandsübergangsmatrix. Muss eine $n \times n$ Matrix sein

At: Optionale Matrix für die Zeitabhängigkeiten in der Zustandsübergangsmatrix. Muss eine $n \times n$ Matrix sein.

C: Beobachtungsmatrix für die Überführung vom Zustand zur Messung. Muss eine $m \times n$ Matrix sein.

Q: Zustandskovarianzmatrix für die Modellierung der Zustandsunsicherheit. Dieser Teil ist zeitunabhängig. Muss eine $n \times n$ Matrix sein.

Q_coeff: Matrix für Koeffizienten der zeitabhängigen Anteile in der Zustandskovarianzmatrix. Muss eine $n \times n$ Matrix sein.

Q_variance: Skalärer 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 $n \times n$ Matrix sein.

R: Sensorkovarianzmatrix für die Modellierung des Sensorrauschens. Muss eine $m \times m$ 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 $n \times n$ Matrix sein.

P0: Initialer Wert der Fehlermatrix, falls man diese zurücksetzen möchte. Ist ein optionaler Parameter, muss aber beim Setzen eine $n \times n$ 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:

self: Object Pointer des KF

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.

Parameters:

self: Object Pointer des KF

getErrorCovarianceMatrix ()

Getter Funktion für die Fehlermatrix.

Parameters:

self: Object Pointer des KF

Returns:

Die P Matrix

likelihood (data_in)

Funktion um die Likelihood für eine Messung zu berechnen.

Parameters:

self: Object Pointer des KF

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:

self: Object Pointer des KF

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:

self: Object Pointer des KF

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 \hat{x} geupdated.

Parameters:

self: Object Pointer des KF

data_in: Aktuelle Messung. Muss ein $m \times 1$ 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

C

`computePrediction()` (Methode von `Kalman_Filter.Kalman_Filter.Kalman_Filter`)

`configure()` (Methode von `Kalman_Filter.Kalman_Filter.Kalman_Filter`)

F

`factorial()` (Methode von `Kalman_Filter.Kalman_Filter.Kalman_Filter`)

G

`getCurrentState()` (Methode von `Kalman_Filter.Kalman_Filter.Kalman_Filter`)

`getErrorCovarianceMatrix()` (Methode von `Kalman_Filter.Kalman_Filter.Kalman_Filter`)

K

`Kalman_Filter` (Klasse in `Kalman_Filter.Kalman_Filter`)

`Kalman_Filter.Kalman_Filter`

Modul

L

`likelihood()` (Methode von `Kalman_Filter.Kalman_Filter.Kalman_Filter`)

M

Modul

`Kalman_Filter.Kalman_Filter`

P

`predict()` (Methode von `Kalman_Filter.Kalman_Filter.Kalman_Filter`)

R

`resetErrorCovAndState()` (Methode von `Kalman_Filter.Kalman_Filter.Kalman_Filter`)

U

`update()` (Methode von `Kalman_Filter.Kalman_Filter.Kalman_Filter`)

Python-Modulindex

k

Kalman_Filter

[Kalman_Filter.Kalman_Filter](#)