KaFi

0.1

Generated by Doxygen 1.8.14

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 kafi::jacobian_function< N, M > Class Template Reference

Public Types

- using **self_t** = jacobian_function< N, M >
- using **nx1_vector** = blaze::StaticMatrix< double, N, 1UL, blaze::rowMajor >
- using **mx1_vector** = blaze::StaticMatrix< double, M, 1UL, blaze::rowMajor >
- using **mxn_matrix** = blaze::StaticMatrix< double, M, N, blaze::rowMajor >
- using **nxm_matrix** = blaze::StaticMatrix< double, N, M, blaze::rowMajor >
- using **mxm_matrix** = blaze::StaticMatrix< double, M, M, blaze::rowMajor >
- using **nxn_matrix** = blaze::StaticMatrix< double, N, N, blaze::rowMajor >
- using **func** = std::function< void(nx1_vector &, mx1_vector &)>
- using **par_jacobi_func** = std::function< double(const nx1_vector &)>
- using **jacobi_func** = blaze::StaticMatrix< par_jacobi_func, M, N, blaze::rowMajor >

Public Member Functions

- constexpr **jacobian_function** (func f, jacobi_func F)
- constexpr **jacobian_function** (const self_t &other)=delete
- constexpr **jacobian_function** (const self_t &&other)
- constexpr void **operator()** (nx1_vector &state, mx1_vector &output) const
- constexpr mxn_matrix & **jacobian** (const nx1_vector &state, mxn_matrix &jacobi_temp) const

Private Attributes

- const func **_f**
- const jacobi_func **_F**

The documentation for this class was generated from the following file:

- library/jacobian_function.h

2.2 kafi::kafi< N, M > Class Template Reference

Public Types

- using **nx1_vector** = typename jacobian_function< N, M >::nx1_vector
- using **mx1_vector** = typename jacobian_function< N, M >::mx1_vector
- using **mxn_matrix** = typename jacobian_function< N, M >::mxn_matrix
- using **nxm_matrix** = typename jacobian_function< N, M >::nxm_matrix
- using **mxm_matrix** = typename jacobian_function< N, M >::mxm_matrix
- using **nxn_matrix** = typename jacobian_function< N, M >::nxn_matrix
- using **self_t** = kafi< N, M >
- using **return_t** = std::tuple< const nx1_vector &, const nxn_matrix &, const nxm_matrix & >

Public Member Functions

- **kafi** (const jacobian_function< N, N > f, const jacobian_function< N, M > h, nx1_vector starting_state, mx1_vector &observation, const nxn_matrix &process_noise, const mxm_matrix &sensor_noise)
- **kafi** (const jacobian_function< N, N > f, const jacobian_function< N, M > h, nx1_vector starting_state, mx1_vector &observation, const nxn_matrix &process_noise, const mxm_matrix &sensor_noise, const nxn↩ _matrix &prediction_error)
- **kafi** (const self_t &other)=delete
- **kafi** (const self_t &&other)=delete
- void **set_current_observation** (mx1_vector &observation)
- std::tuple< const nx1_vector &, const nxn_matrix &, const nxm_matrix & > **step** ()
- void **print_state_to** (std::ostream &stream)

Private Member Functions

- bool **new_data_available** ()
- void **apply_prediction** ()
- void **apply_update** ()

Private Attributes

- const jacobian_function< N, N > **_f**
- nxn_matrix **_f_jacobian_temp**
- const jacobian_function< N, M > **_h**
- mx1_vector **_h_temp**
- mxn_matrix **_h_jacobian_temp**
- const nxn_matrix **_process_noise**
- const mxm_matrix **_sensor_noise**
- const nxn_matrix **_identity**
- nx1_vector **_state**
- mx1_vector & **_observation**
- nxn_matrix **_prediction_error**
- nxm_matrix **_gain**
- bool **_new_data_available**
- size_t **_prediction_count**
- size_t **_update_count**

Friends

- std::ostream & **operator<<** (std::ostream &stream, const self_t &rhs)

The documentation for this class was generated from the following file:

- library/kafi.h

2.3 kafi::kalman_filter Class Reference

Public Member Functions

- **kalman_filter** (size_t state_dimension_n, size_t sensor_dimension_m, size_t control_dimension_l)
- void **kalman_step** (const blaze::DynamicMatrix< float > &jacobian_cf_m, const blaze::DynamicMatrix< float > &noise_q_m, const blaze::DynamicMatrix< float > &c_h_m, const blaze::DynamicMatrix< float > &cov_noise_m, const blaze::DynamicVector< float > &observed_v)

Private Member Functions

- void **_predict** (const blaze::DynamicMatrix< float > &jacobian_cf_m, const blaze::DynamicMatrix< float > &noise_q_m)
- void **_update** (const blaze::DynamicMatrix< float > &c_h_m, const blaze::DynamicMatrix< float > &cov↩_noise_m, const blaze::DynamicVector< float > &observed_v)

Private Attributes

- size_t **_state_dimension_n** {0}
- size_t **_sensor_dimension_m** {0}
- size_t **_control_dimension_l** {0}
- blaze::DiagonalMatrix< blaze::DynamicMatrix< float > > **_predict_old_m**
- blaze::DynamicVector< float, blaze::columnVector > **_est_state_old_v**
- blaze::DiagonalMatrix< blaze::DynamicMatrix< float > > **_predict_m**
- blaze::DynamicVector< float, blaze::columnVector > **_est_state_v**
- blaze::DynamicMatrix< float > **_gain_m**

The documentation for this class was generated from the following file:

- library/kalman_filter.h

2.4 kafi::prediction< N, M > Class Template Reference

Public Types

- using **self_t** = prediction< N, M >
- using **nxn_matrix** = blaze::StaticMatrix< double, N, N, blaze::rowMajor >
- using **nx1_vector** = blaze::StaticMatrix< double, N, 1UL, blaze::rowMajor >

Public Member Functions

- **prediction** (const self_t &other)=delete
- **prediction** (const self_t &&other)=delete
- std::tuple< nx1_vector &, nxn_matrix & > & **apply** (const jacobian_function< N, N > &state_transition, nxn_matrix &, const nxn_matrix &prediction_error, const nxn_matrix &&process_noise)

The documentation for this class was generated from the following file:

- library/prediction.h

2.5 kafi::update< N, M > Class Template Reference

Public Types

- using **nx1_vector** = blaze::StaticMatrix< double, N, 1UL, blaze::rowMajor >
- using **mx1_vector** = blaze::StaticMatrix< double, M, 1UL, blaze::rowMajor >
- using **mxn_matrix** = blaze::StaticMatrix< double, M, N, blaze::rowMajor >
- using **nxm_matrix** = blaze::StaticMatrix< double, N, M, blaze::rowMajor >
- using **mxm_matrix** = blaze::StaticMatrix< double, M, M, blaze::rowMajor >

Public Member Functions

- **update** (const jacobian_function< N, M > &prediction_scaling, const mxm_matrix &sensor_noise, nxn_↩ matrix &prediction_error, prediction< N, M > &prediction)
- std::tuple< nx1_vector, nxn_matrix > **apply** (nx1_vector new_state)

Private Attributes

- const jacobian_function< N, M > **_prediction_scaling**
- const mxm_matrix **_sensor_noise**
- nx1_vector & **_state**
- nxn_matrix **_prediction_error**
- prediction< N, M > & **_prediction**
- nxm_matrix **_gain**

The documentation for this class was generated from the following file:

- library/update.h

Index