

Фильтры Калмана

Программная документация



1 Результаты моделирования фильтров Калмана	1
2 Алфавитный указатель групп	7
2.1 Группы	7
3 Алфавитный указатель пространств имен	9
3.1 Пространства имен	9
4 Иерархический список классов	11
4.1 Иерархия классов	11
5 Алфавитный указатель классов	13
5.1 Классы	13
6 Группы	15
6.1 Kalman_filters	15
6.1.1 Подробное описание	15
7 Пространства имен	17
7.1 Пространство имен KalmanFilters	17
7.1.1 Подробное описание	18
7.1.2 Функции	18
8 Классы	19
8.1 Шаблон класса KalmanFilters::CKalmanCKF< SizeX, SizeY >	19
8.1.1 Подробное описание	21
8.1.2 Конструктор(ы)	21
8.1.3 Методы	22
8.1.4 Документация по друзьям класса и функциям, относящимся к классу	24
8.1.5 Данные класса	25
8.2 Шаблон класса KalmanFilters::CKalmanECKF< SizeX, SizeY >	26
8.2.1 Подробное описание	29
8.2.2 Конструктор(ы)	29
8.2.3 Методы	30
8.2.4 Документация по друзьям класса и функциям, относящимся к классу	31
8.3 Шаблон класса KalmanFilters::CKalmanEKF< SizeX, SizeY >	32
8.3.1 Подробное описание	33
8.3.2 Конструктор(ы)	34
8.3.3 Методы	34
8.3.4 Документация по друзьям класса и функциям, относящимся к классу	37
8.3.5 Данные класса	37
8.4 Шаблон класса KalmanFilters::CKalmanEUKF< SizeX, SizeY >	39
8.4.1 Подробное описание	41
8.4.2 Конструктор(ы)	41
8.4.3 Методы	42
8.4.4 Документация по друзьям класса и функциям, относящимся к классу	43

8.5	Шаблон класса <code>KalmanFilters::CKalmanLKF&lt; SizeX, SizeY &gt;</code> . . . . .	44
8.5.1	Подробное описание . . . . .	47
8.5.2	Конструктор(ы) . . . . .	47
8.5.3	Методы . . . . .	48
8.5.4	Документация по друзьям класса и функциям, относящимся к классу . . . . .	56
8.5.5	Данные класса . . . . .	56
8.6	Шаблон класса <code>KalmanFilters::CKalmanSRCKF&lt; SizeX, SizeY &gt;</code> . . . . .	59
8.6.1	Подробное описание . . . . .	61
8.6.2	Конструктор(ы) . . . . .	61
8.6.3	Методы . . . . .	62
8.6.4	Документация по друзьям класса и функциям, относящимся к классу . . . . .	64
8.7	Шаблон класса <code>KalmanFilters::CKalmanSRCKFB&lt; SizeX, SizeY &gt;</code> . . . . .	64
8.7.1	Подробное описание . . . . .	66
8.7.2	Конструктор(ы) . . . . .	67
8.7.3	Методы . . . . .	67
8.8	Шаблон класса <code>KalmanFilters::CKalmanSRECKF&lt; SizeX, SizeY &gt;</code> . . . . .	67
8.8.1	Подробное описание . . . . .	68
8.8.2	Конструктор(ы) . . . . .	69
8.8.3	Методы . . . . .	69
8.8.4	Документация по друзьям класса и функциям, относящимся к классу . . . . .	70
8.9	Шаблон класса <code>KalmanFilters::CKalmanSRECKFB&lt; SizeX, SizeY &gt;</code> . . . . .	71
8.9.1	Подробное описание . . . . .	72
8.9.2	Конструктор(ы) . . . . .	72
8.9.3	Методы . . . . .	73
8.9.4	Документация по друзьям класса и функциям, относящимся к классу . . . . .	74
8.10	Шаблон класса <code>KalmanFilters::CKalmanSREKF&lt; SizeX, SizeY &gt;</code> . . . . .	74
8.10.1	Подробное описание . . . . .	76
8.10.2	Конструктор(ы) . . . . .	76
8.10.3	Методы . . . . .	77
8.10.4	Документация по друзьям класса и функциям, относящимся к классу . . . . .	78
8.11	Шаблон класса <code>KalmanFilters::CKalmanSREUKF&lt; SizeX, SizeY &gt;</code> . . . . .	78
8.11.1	Подробное описание . . . . .	80
8.11.2	Конструктор(ы) . . . . .	80
8.11.3	Методы . . . . .	81
8.11.4	Документация по друзьям класса и функциям, относящимся к классу . . . . .	82
8.11.5	Данные класса . . . . .	82
8.12	Шаблон класса <code>KalmanFilters::CKalmanSREUKFB&lt; SizeX, SizeY &gt;</code> . . . . .	83
8.12.1	Подробное описание . . . . .	84
8.12.2	Конструктор(ы) . . . . .	84
8.12.3	Методы . . . . .	85
8.12.4	Документация по друзьям класса и функциям, относящимся к классу . . . . .	86
8.12.5	Данные класса . . . . .	86
8.13	Шаблон класса <code>KalmanFilters::CKalmanSRUKF&lt; SizeX, SizeY &gt;</code> . . . . .	87

---

8.13.1 Подробное описание . . . . .	88
8.13.2 Конструктор(ы) . . . . .	89
8.13.3 Методы . . . . .	90
8.13.4 Документация по друзьям класса и функциям, относящимся к классу . . . . .	92
8.13.5 Данные класса . . . . .	93
8.14 Шаблон класса KalmanFilters::CKalmanSRUKFB< SizeX, SizeY > . . . . .	93
8.14.1 Подробное описание . . . . .	96
8.14.2 Конструктор(ы) . . . . .	96
8.14.3 Методы . . . . .	97
8.14.4 Документация по друзьям класса и функциям, относящимся к классу . . . . .	98
8.14.5 Данные класса . . . . .	98
8.15 Шаблон класса KalmanFilters::CKalmanUKF< SizeX, SizeY > . . . . .	98
8.15.1 Подробное описание . . . . .	101
8.15.2 Конструктор(ы) . . . . .	101
8.15.3 Методы . . . . .	102
8.15.4 Документация по друзьям класса и функциям, относящимся к классу . . . . .	105
8.15.5 Данные класса . . . . .	106
Предметный указатель . . . . .	109



## Раздел 1

# Результаты моделирования фильтров Калмана

- 
- 1) ЛФК - Линейный Фильтр Калмана (LKF - Linear Kalman Filter);
  - 2) РФК - Расширенный Фильтр Калмана (EKF - Extended Kalman Filter);
  - 3) ККРФК - Квадратно-Корневой Расширенный Фильтр Калмана (SREKF - Square Root Extended Kalman Filter);
  - 4) СТФК - Сигма-точечный (ансцентный) Фильтр Калмана (UKF - Unscented Kalman Filter);
  - 5) КК-СТФК - Квадратно-Корневой Сигма-точечный Фильтр Калмана (SRUKF - Square Root Unscented Kalman Filter);
  - 6) КК-СТФКБ - Блочная реализация КК-СТФК (SRUKFB - Square Root Unscented Kalman Filter Block);
  - 7) КФК - Кубатурный Фильтр Калмана (CKF - Cubature Kalman Filter);
  - 8) КК-КФК - Квадратно-Корневой Кубатурный Фильтр Калмана (SRCKF - Square Root Cubature Kalman Filter);
  - 9) КК-КФКБ - Блочная реализация КК-КФК (SRCKFB - Square Root Cubature Kalman Filter Block);
  - 10) РСТФК - Расширенно-Сигма-точечный Фильтр Калмана (EUKF - Extended Unscented Kalman Filter);
  - 11) КК-РСТФК - Квадратно-Корневой Расширенно-Сигма-точечный Фильтр Калмана (SREUKF - Square Root Extended Unscented Kalman Filter);
  - 12) КК-РСТФКБ - Блочная реализация КК-РСТФК (SREUKFB - Square Root Extended Unscented Kalman Filter Block);
  - 13) РКФК - Расширенно-Кубатурный Фильтр Калмана (ECKF - Extended Cubature Kalman Filter);
  - 14) КК-РКФК - Квадратно-Корневой Расширенно-Кубатурный Фильтр Калмана (SRECKF - Square Root Extended Cubature Kalman Filter);
  - 15) КК-РКФКБ - Блочная реализация КК-РКФК (SRECKFB - Square Root Extended Cubature Kalman Filter Block).
- 

Выбранная система координат:

$$X = \{ X, Y, V, K, dK/dt \}, Y = \{ R, Az, Vr \},$$

где пространство состояния:

- $X, Y$  - плоскостные декартовы координаты, км;
- $V$  - полная скорость, м/с;
- $K$  - курс, град;
- $dK/dt$  - скорость изменения курса, град/с;

пространство измерений:

- $R$  - дальность, км;

- Az - азимут, град;
- Vr - радиальная скорость, м/с;

ЛФК не может быть применен при выбранной системе координат напрямую, поэтому не приводится далее.

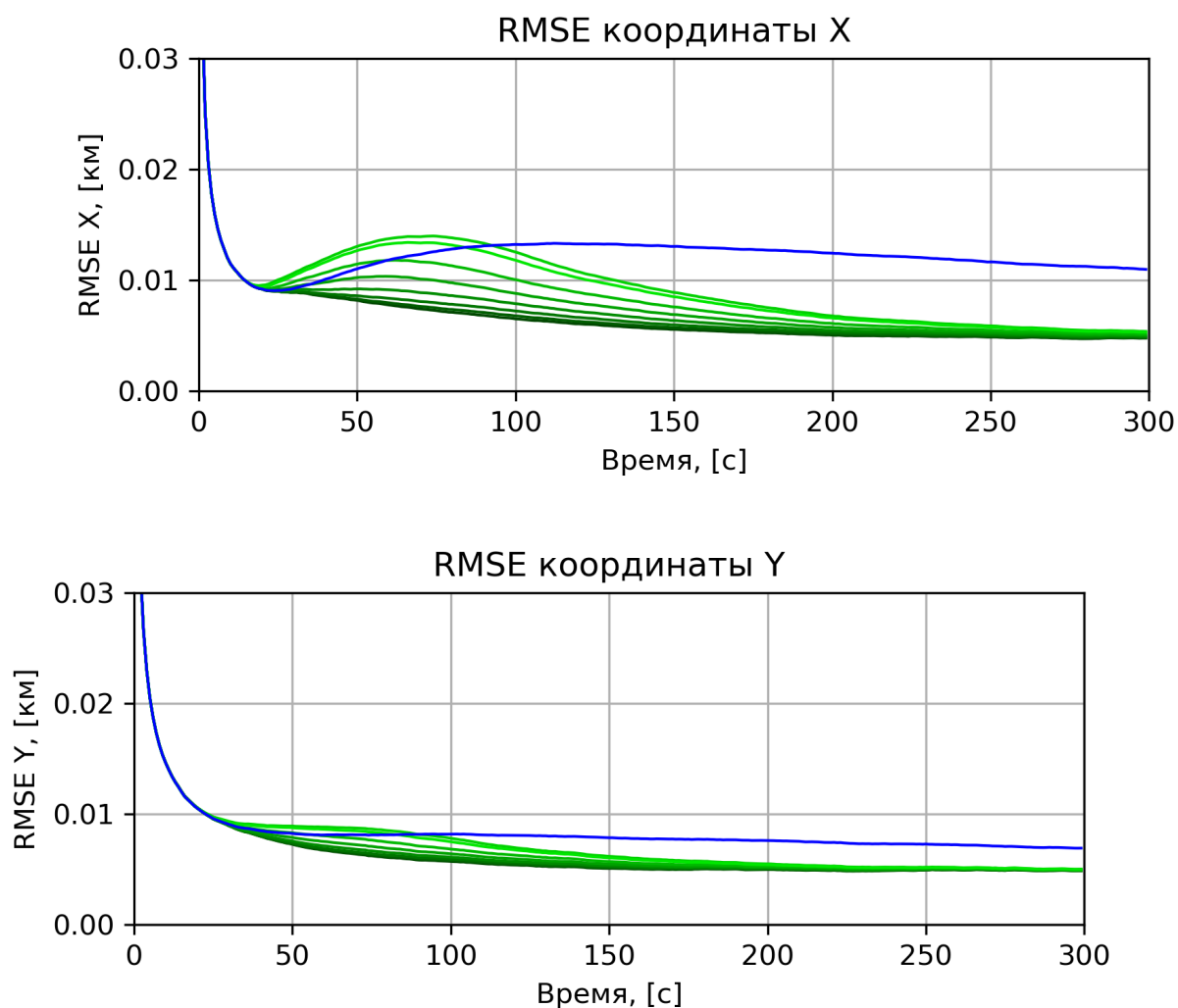
Начальное состояние объекта:

- X = 100 км;
- Y = 100 км;
- V = 100 м/с;
- K = 45 град;
- $dK/dt = 0$  град/с;

СКО измерений:

- RMS\_R = 0.048 км;
- RMS\_Az = 0.008 град;
- RMS\_Vr = 0.016 м/с;

Рис.1 - RMSE вектора состояния при изменении параметра разброса сигма-точек  $\alpha=1.0...0.35$





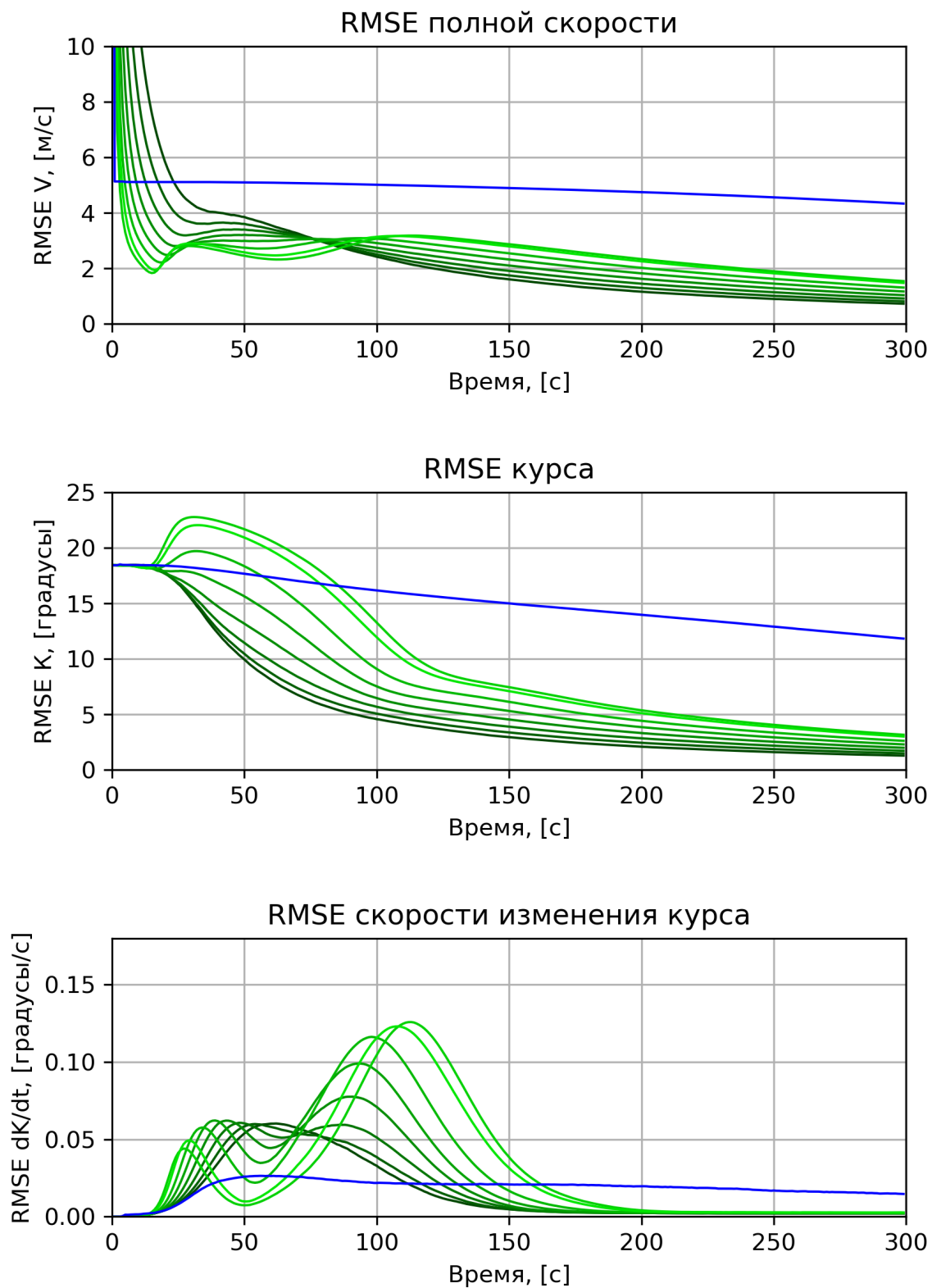
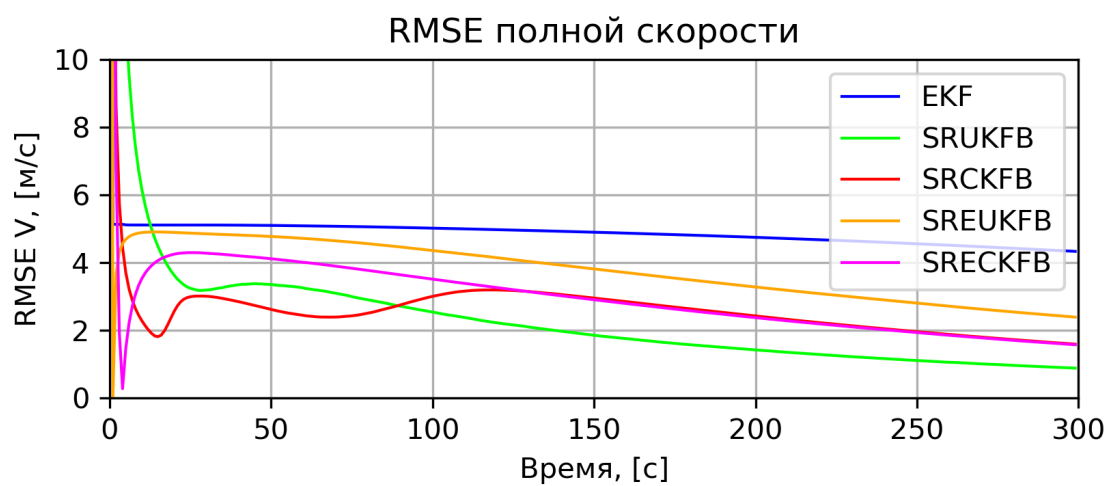
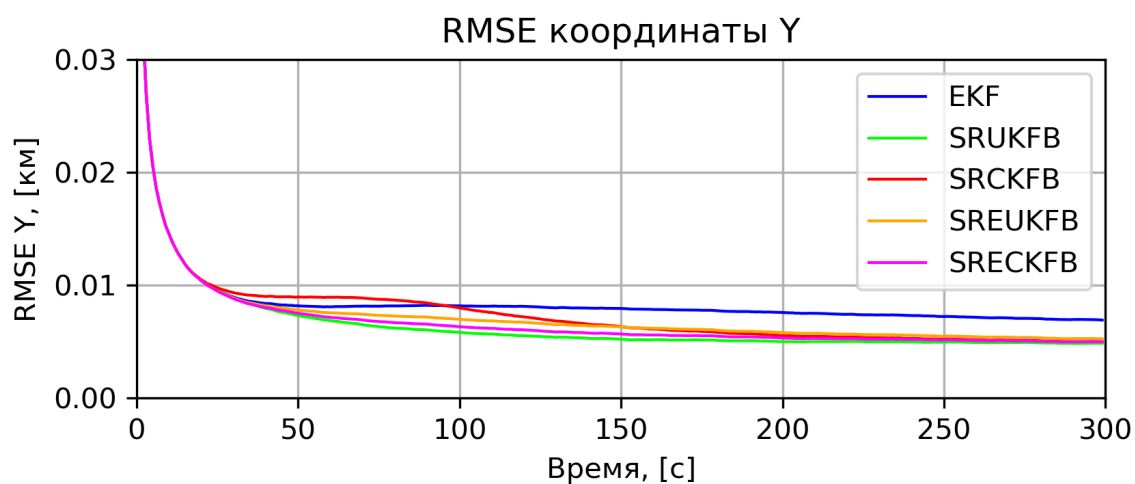
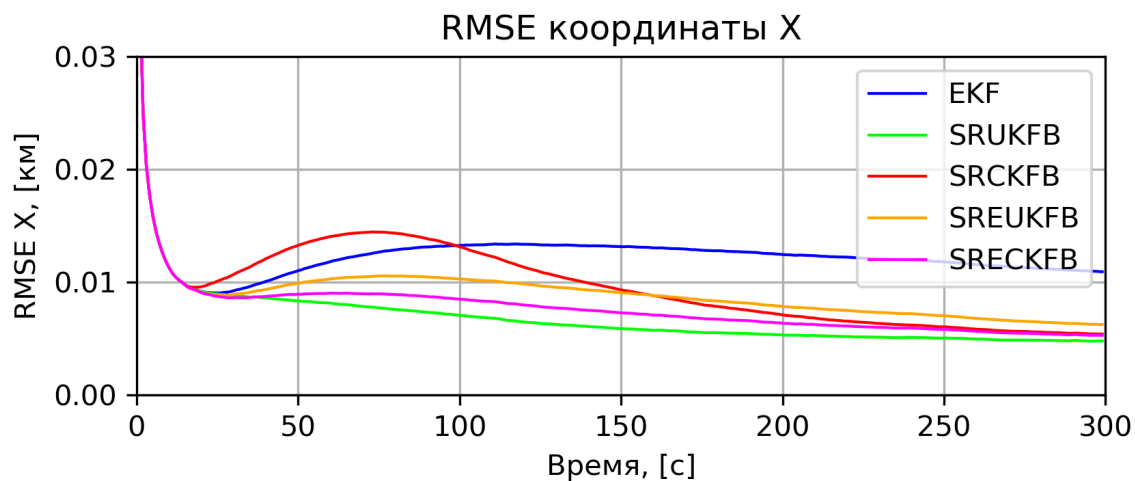


Рис.2 - Сравнение RMSE вектора состояния



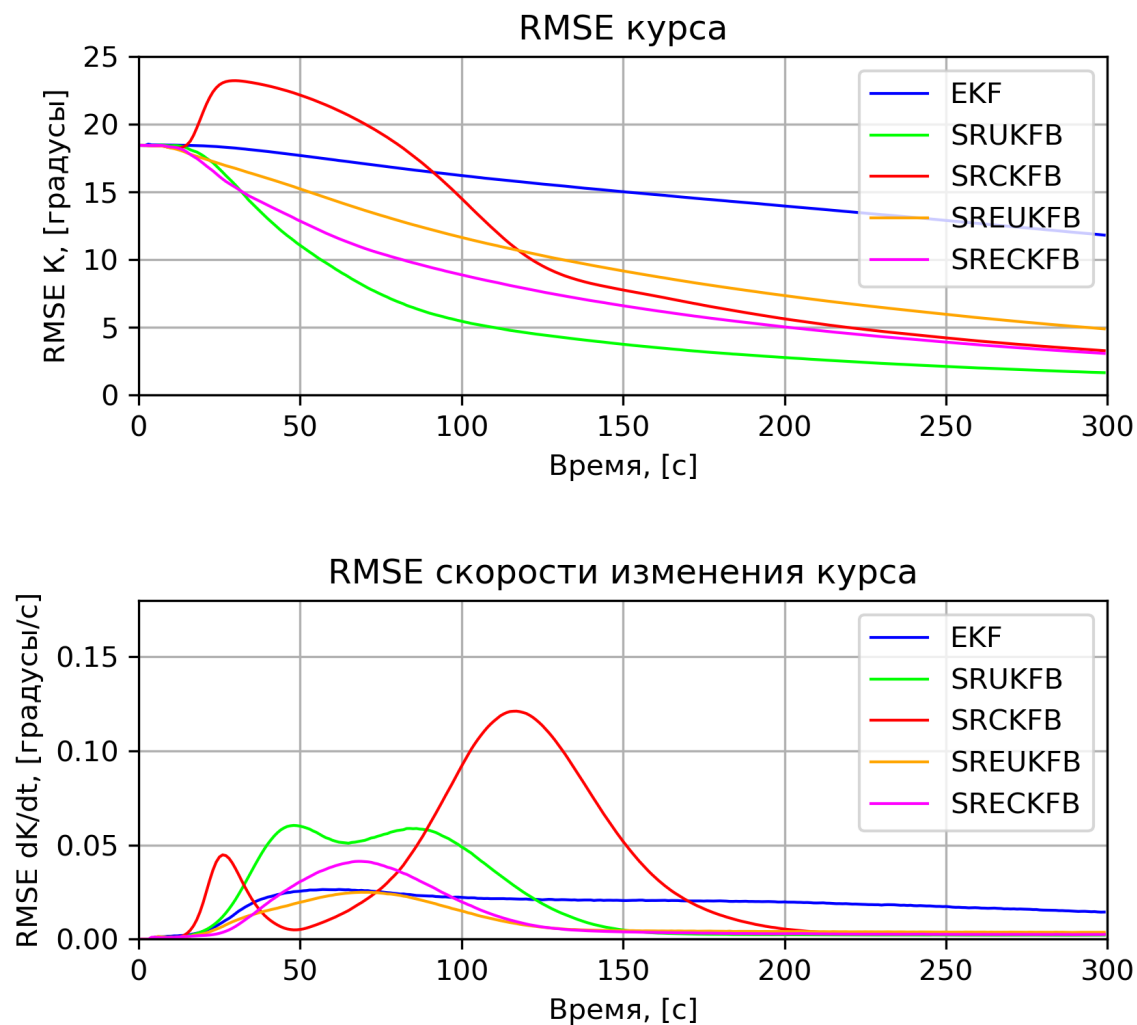


Рис.3 - Время выполнения относительно EKF при JQR разложении в SRUKF фильтрах

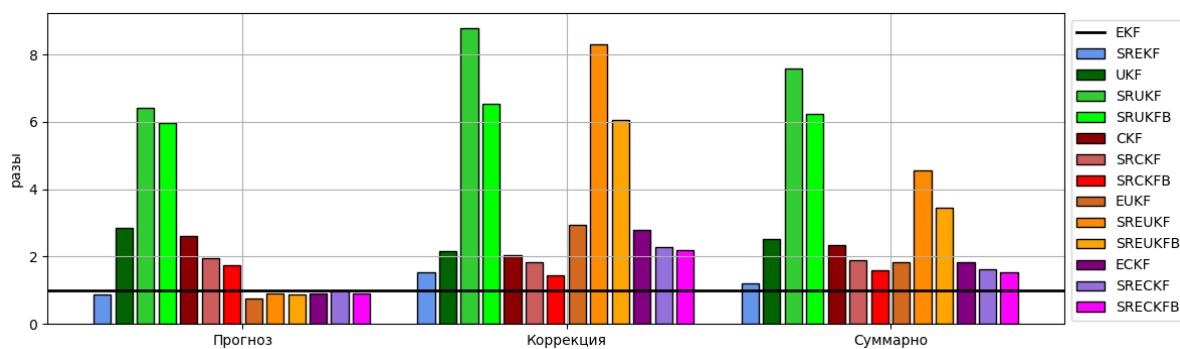
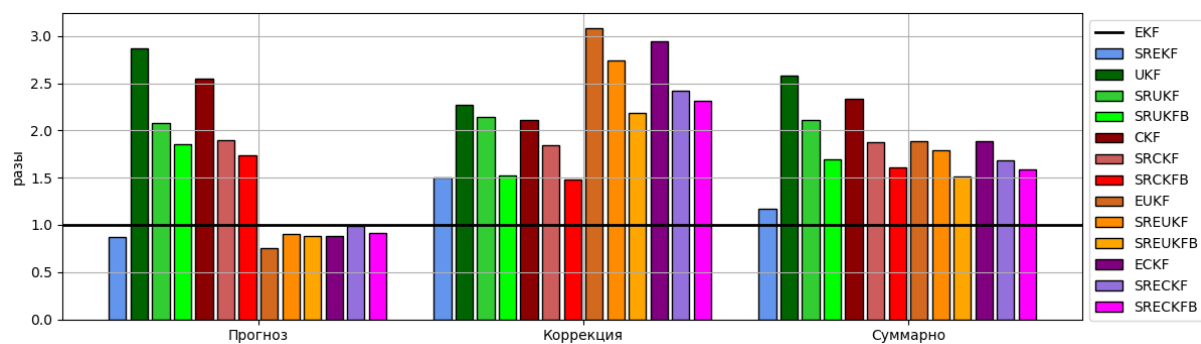


Рис.4 - Время выполнения относительно EKF при QR разложении в SRUKF фильтрах



## Раздел 2

# Алфавитный указатель групп

### 2.1 Группы

Полный список групп.

`Kalman_filters` . . . . . 15



## Раздел 3

# Алфавитный указатель пространств имен

### 3.1 Пространства имен

Полный список пространств имен.

[KalmanFilters](#)

Фильтры Калмана . . . . . 17





## Раздел 4

# Иерархический список классов

### 4.1 Иерархия классов

Иерархия классов.

KalmanFilters::CKalmanLKF< SizeX, SizeY > . . . . .	44
KalmanFilters::CKalmanEKF< SizeX, SizeY > . . . . .	32
KalmanFilters::CKalmanCKF< SizeX, SizeY > . . . . .	19
KalmanFilters::CKalmanECKF< SizeX, SizeY > . . . . .	26
KalmanFilters::CKalmanSRCKF< SizeX, SizeY > . . . . .	59
KalmanFilters::CKalmanSRCKFB< SizeX, SizeY > . . . . .	64
KalmanFilters::CKalmanSRECKF< SizeX, SizeY > . . . . .	67
KalmanFilters::CKalmanSRECKFB< SizeX, SizeY > . . . . .	71
KalmanFilters::CKalmanSREKF< SizeX, SizeY > . . . . .	74
KalmanFilters::CKalmanSRECKF< SizeX, SizeY > . . . . .	67
KalmanFilters::CKalmanSREUKF< SizeX, SizeY > . . . . .	78
KalmanFilters::CKalmanSREUKFB< SizeX, SizeY > . . . . .	83
KalmanFilters::CKalmanUKF< SizeX, SizeY > . . . . .	98
KalmanFilters::CKalmanEUKF< SizeX, SizeY > . . . . .	39
KalmanFilters::CKalmanSRUKF< SizeX, SizeY > . . . . .	87
KalmanFilters::CKalmanSREUKF< SizeX, SizeY > . . . . .	78
KalmanFilters::CKalmanSRUKFB< SizeX, SizeY > . . . . .	93



## Раздел 5

# Алфавитный указатель классов

### 5.1 Классы

Классы с их кратким описанием.

<a href="#">KalmanFilters::CKalmanCKF&lt; SizeX, SizeY &gt;</a>	Шаблонный класс кубатурного фильтра Калмана, КФК (Cubature Kalman Filter, CKF) . . . . .	19
<a href="#">KalmanFilters::CKalmanECKF&lt; SizeX, SizeY &gt;</a>	Шаблонный класс расширенного кубатурного фильтра Калмана, РКФК (Extended Cubature Kalman Filter, ECKF) . . . . .	26
<a href="#">KalmanFilters::CKalmanEKF&lt; SizeX, SizeY &gt;</a>	Класс расширенного фильтра Калмана, РФК (Extended Kalman Filter, EKF) . .	32
<a href="#">KalmanFilters::CKalmanEUKF&lt; SizeX, SizeY &gt;</a>	Шаблонный класс расширенного сигма-точечного (ансцентного) фильтра Калмана, РСТФК (Extended Unscented Kalman Filter, EUKF) . . . . .	39
<a href="#">KalmanFilters::CKalmanLKF&lt; SizeX, SizeY &gt;</a>	Шаблонный класс линейного фильтра Калмана, ЛФК (Linear Kalman Filter, LKF)	44
<a href="#">KalmanFilters::CKalmanSRCKF&lt; SizeX, SizeY &gt;</a>	Шаблонный класс квадратно-корневого кубатурного фильтра Калмана, КК-КФК (Square Root Cubature Kalman Filter, SR-CKF) . . . . .	59
<a href="#">KalmanFilters::CKalmanSRCKFB&lt; SizeX, SizeY &gt;</a>	Шаблонный класс квадратно-корневого кубатурного фильтра Калмана (блочная фильтрация), КК-КФКБ (Square Root Cubature Kalman Filter Block, SR-CKFB) .	64
<a href="#">KalmanFilters::CKalmanSRECKF&lt; SizeX, SizeY &gt;</a>	Шаблонный класс квадратно-корневого расширенного кубатурного фильтра Калмана, КК-РКФК (Square Root Extended Cubature Kalman Filter, SR-ECKF) . . .	67
<a href="#">KalmanFilters::CKalmanSRECKFB&lt; SizeX, SizeY &gt;</a>	Шаблонный класс квадратно-корневого расширенного кубатурного фильтра Калмана (блочная фильтрация), КК-РКФКБ (Square Root Extended Cubature Kalman Filter Block, SR-ECKFB) . . . . .	71
<a href="#">KalmanFilters::CKalmanSREKF&lt; SizeX, SizeY &gt;</a>	Класс квадратно-корневого расширенного фильтра Калмана, КК-РФК (Square Root Extended Kalman Filter, EKF) . . . . .	74
<a href="#">KalmanFilters::CKalmanSREUKF&lt; SizeX, SizeY &gt;</a>	Шаблонный класс квадратно-корневого расширенного сигма-точечного (ансцентного) фильтра Калмана, КК-РСТФК (Square Root Extended Unscented Kalman Filter, SR-EUKF) . . . . .	78
<a href="#">KalmanFilters::CKalmanSREUKFB&lt; SizeX, SizeY &gt;</a>	Шаблонный класс квадратно-корневого расширенного сигма-точечного (ансцентного) фильтра Калмана (блочная фильтрация), КК-РСТФК (Square Root Extended Unscented Kalman Filter Block, SR-EUKFB) . . . . .	83
<a href="#">KalmanFilters::CKalmanSRUKF&lt; SizeX, SizeY &gt;</a>	Шаблонный класс квадратно-корневого сигма-точечного (ансцентного) фильтра Калмана, КК-СТФК (КК-АФК) (Square Root Unscented Kalman Filter, SR-UKF)	87

[KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >](#)

Шаблонный класс квадратно-корневого сигма-точечного (ансцентного) фильтра Калмана (блочная фильтрация), КК-СТФКБ (КК-АФКБ) (Square Root Unscented Kalman Filter Block, SR-UKFB) . . . . . 93

[KalmanFilters::CKalmanUKF< SizeX, SizeY >](#)

Шаблонный класс сигма-точечного (ансцентного) фильтра Калмана, СТФК (АФК) (Unscented Kalman Filter, UKF) . . . . . 98

## Раздел 6

# Группы

### 6.1 Kalman\_filters

Пространства имен

- namespace [KalmanFilters](#)  
Фильтры Калмана

#### 6.1.1 Подробное описание



## Раздел 7

# Пространства имен

### 7.1 Пространство имен KalmanFilters

Фильтры Калмана

Классы

- class [CKalmanCKF](#)  
Шаблонный класс кубатурного фильтра Калмана, КФК (Cubature Kalman Filter, CKF)
- class [CKalmanECKF](#)  
Шаблонный класс расширенного кубатурного фильтра Калмана, РКФК (Extended Cubature Kalman Filter, ECKF)
- class [CKalmanEKF](#)  
Класс расширенного фильтра Калмана, РКФ (Extended Kalman Filter, EKF)
- class [CKalmanEUKF](#)  
Шаблонный класс расширенного сигма-точечного (ансцентного) фильтра Калмана, РСТФК (Extended Unscented Kalman Filter, EUKF)
- class [CKalmanLKF](#)  
Шаблонный класс линейного фильтра Калмана, ЛФК (Linear Kalman Filter, LKF)
- class [CKalmanSRCKF](#)  
Шаблонный класс квадратно-корневого кубатурного фильтра Калмана, КК-КФК (Square Root Cubature Kalman Filter, SR-CKF)
- class [CKalmanSRCKFB](#)  
Шаблонный класс квадратно-корневого кубатурного фильтра Калмана (блочная фильтрация), КК-КФКБ (Square Root Cubature Kalman Filter Block, SR-CKFB)
- class [CKalmanSRECKF](#)  
Шаблонный класс квадратно-корневого расширенного кубатурного фильтра Калмана, КК-РКФК (Square Root Extended Cubature Kalman Filter, SR-ECKF)
- class [CKalmanSRECKFB](#)  
Шаблонный класс квадратно-корневого расширенного кубатурного фильтра Калмана (блочная фильтрация), КК-РКФКБ (Square Root Extended Cubature Kalman Filter Block, SR-ECKFB)
- class [CKalmanSREKF](#)  
Класс квадратно-корневого расширенного фильтра Калмана, КК-РФК (Square Root Extended Kalman Filter, EKF)
- class [CKalmanSREUKF](#)  
Шаблонный класс квадратно-корневого расширенного сигма-точечного (ансцентного) фильтра Калмана, КК-РСТФК (Square Root Extended Unscented Kalman Filter, SR-EUKF)
- class [CKalmanSREUKFB](#)  
Шаблонный класс квадратно-корневого расширенного сигма-точечного (ансцентного) фильтра Калмана (блочная фильтрация), КК-РСТФК (Square Root Extended Unscented Kalman Filter Block, SR-EUKFB)

- class [CKalmanSRUKF](#)

Шаблонный класс квадратно-корневого сигма-точечного (ансцентного) фильтра Калмана, КК-СТФК (КК-АФК) (Square Root Unscented Kalman Filter, SR-UKF)

- class [CKalmanSRUKFB](#)

Шаблонный класс квадратно-корневого сигма-точечного (ансцентного) фильтра Калмана (блочная фильтрация), КК-СТФКБ (КК-АФКБ) (Square Root Unscented Kalman Filter Block, SR-UKFB)

- class [CKalmanUKF](#)

Шаблонный класс сигма-точечного (ансцентного) фильтра Калмана, СТФК (АФК) (Unscented Kalman Filter, UKF)

## Функции

- std::string [GetVersion](#) ()

Возвращает строку, содержащую информацию о версии

### 7.1.1 Подробное описание

#### Фильтры Калмана

### 7.1.2 Функции

#### 7.1.2.1 GetVersion()

std::string KalmanFilters::GetVersion ()

Возвращает строку, содержащую информацию о версии

Возвращает

Строка версии в формате DD-MM-YY-VV\_COMMENTS, где DD - день, MM - месяц, YY - год, VV - версия, COMMENTS - комментарий(опционально)



## Раздел 8

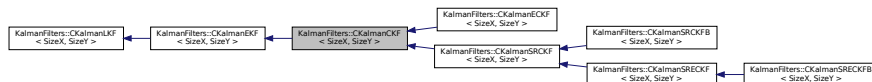
# Классы

### 8.1 Шаблон класса KalmanFilters::CKalmanCKF< SizeX, SizeY >

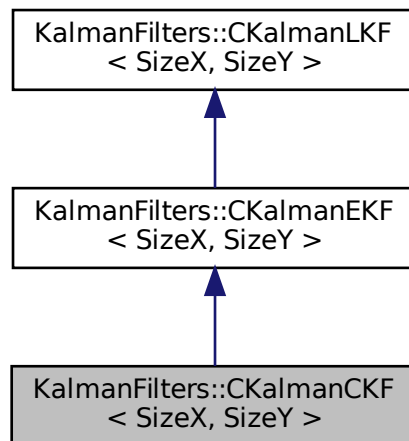
Шаблонный класс кубатурного фильтра Калмана, КФК (Cubature Kalman Filter, CKF)

#include <kalman\_filter\_cubature.h>

Граф наследования: KalmanFilters::CKalmanCKF< SizeX, SizeY >:



Граф связей класса KalmanFilters::CKalmanCKF< SizeX, SizeY >:



Открытые члены

- [CKalmanCKF](#) ()  
Конструктор по умолчанию
- [CKalmanCKF](#) (const [CKalmanCKF](#) &other)  
Конструктор копирования

- [CKalmanCKF](#) & [operator=](#) (const [CKalmanCKF](#) &other)  
Перегрузка оператора присвоения
- [CKalmanCKF](#) ([CKalmanCKF](#) &&other) noexcept  
Конструктор перемещения
- [CKalmanCKF](#) & [operator=](#) ([CKalmanCKF](#) &&other) noexcept  
Перегрузка оператора перемещения
- virtual [~CKalmanCKF](#) ()=default  
Деструктор
- void [SetWeightedSumStateSigmas](#) (std::function< arma::vec(const arma::vec &weights, const arma::mat &sigmaPoints)> weightedSumStateSigmas)  
Установка функции вычисления взвешенной суммы сигма-точек пространства X
- void [SetWeightedSumMeasurementSigmas](#) (std::function< arma::vec(const arma::vec &weights, const arma::mat &sigmaPoints)> weightedSumMeasurementSigmas)  
Установка функции вычисления взвешенной суммы сигма-точек пространства Y.
- virtual void [Prediction](#) (double dt)  
Прогноз
- virtual void [Correction](#) (const arma::vec &Y\_msd)  
Коррекция

### Защищенные члены

- void [PredictionCKF](#) (double dt)  
Прогноз CKF.
- void [CorrectionCKF](#) (const arma::vec &Y\_msd)  
Коррекция CKF.
- void [SetupDesignParametersCubatureBaseSet](#) ()  
Установка кубатурных весов (базовый вариант ансцентного преобразования)
- void [SetStateTransitionJacobianF](#) ()  
Запрет доступа
- void [SetObservationJacobianH](#) ()  
Запрет доступа

### Защищенные данные

- int [k\\_sigma\\_points\\_](#)  
Число сигма-точек
- arma::vec [weights\\_mean\\_](#)  
Веса среднего
- arma::vec [weights\\_covariance\\_](#)  
Веса ковариации
- arma::mat [x\\_est\\_sigma\\_points\\_](#)  
Матрица сигма-точек (сигма-точки - столбцы) в пространстве X на текущем такте, размерность [SizeX,k\_sigma\_points\_].
- arma::mat [x\\_pred\\_sigma\\_points\\_](#)  
Матрица сигма-точек (сигма-точки - столбцы) в пространстве X, экстраполированный на текущий такт, размерность [SizeX,k\_sigma\_points\_].
- arma::mat [y\\_pred\\_sigma\\_points\\_](#)  
Матрица сигма-точек (сигма-точки - столбцы) в пространстве Y, экстраполированный на текущий такт, размерность [SizeX,k\_sigma\_points\_].
- arma::mat [dXcal\\_](#)  
Матрица X-каллиграфическое (матрица сигма-точек - столбцов)
- arma::mat [dYcal\\_](#)  
Матрица Y-каллиграфическое (матрица сигма-точек - столбцов)

- arma::mat [P\\_xy\\_](#)  
Матрица кросс-коварации векторов X и Y, размерность [SizeX \* SizeY].
- arma::mat [sqrt\\_P\\_chol\\_](#)  
Корень из матрицы P.
- double [gamma\\_](#)  
Автоматически вычисляемый (в методах SetDesignParameters\*) параметр (множитель при корне из P при создании сигма-точек)
- std::function< arma::vec(const arma::vec &weights, const arma::mat &sigmaPoints)> [weightedSumStateSigmas\\_](#)  
Вычисление взвешенной суммы сигма-точек пространства X
- std::function< arma::vec(const arma::vec &weights, const arma::mat &sigmaPoints)> [weightedSumMeasurementSigmas\\_](#)  
Вычисление взвешенной суммы сигма-точек пространства Y.

## Друзья

- void [swap](#) (CKalmanCKF< SizeX, SizeY > &lhs, CKalmanCKF< SizeX, SizeY > &rhs) noexcept  
Метод свапа

### 8.1.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanCKF< SizeX, SizeY >
```

Шаблонный класс кубатурного фильтра Калмана, КФК (Cubature Kalman Filter, CKF)

Частный случай сигма-точечного фильтра Калмана при параметрах разброса сигма-точек выбранных по рекомендации Merwe alpha=1.0, beta=0.0, kappa=0.0.

Источники:

[1] Cubature Kalman Filters, Ienkar Arasaratnam and Simon Haykin, Life Fellow, IEEE

[2] Sebastian Bitzer, Technische Universität Dresden, <https://github.com/sbitzer/UKF-exposed/blob/master/UKF.pdf>

Внимание

Фильтр построен по классическому его варианту НИЖНЕтреугольного разложения Холецкого!

Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

### 8.1.2 Конструктор(ы)

#### 8.1.2.1 CKalmanCKF() [1/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanCKF< SizeX, SizeY >::CKalmanCKF ( ) [inline]
Конструктор по умолчанию
```

#### 8.1.2.2 CKalmanCKF() [2/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanCKF< SizeX, SizeY >::CKalmanCKF (
    const CKalmanCKF< SizeX, SizeY > & other ) [inline]
Конструктор копирования
```

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.1.2.3 CKalmanCKF() [3/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanCKF< SizeX, SizeY >::CKalmanCKF (
    CKalmanCKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Конструктор перемещения

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.1.2.4 ~CKalmanCKF()

```
template<size_t SizeX, size_t SizeY>
virtual KalmanFilters::CKalmanCKF< SizeX, SizeY >::~~CKalmanCKF ( ) [virtual], [default]
```

Деструктор

## 8.1.3 Методы

## 8.1.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanCKF< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd ) [inline], [virtual]
```

Коррекция

## Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanEKF< SizeX, SizeY >](#).

Переопределяется в [KalmanFilters::CKalmanSRCKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRCKFB< SizeX, SizeY >](#), [KalmanFilters::CKalmanECKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRECKF< SizeX, SizeY >](#) и [KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >](#).

## 8.1.3.2 CorrectionCKF()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanCKF< SizeX, SizeY >::CorrectionCKF (
    const arma::vec & Y_msd ) [inline], [protected]
```

Коррекция CKF.

## Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

## 8.1.3.3 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanCKF & KalmanFilters::CKalmanCKF< SizeX, SizeY >::operator= (
    CKalmanCKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Перегрузка оператора перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

## 8.1.3.4 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanCKF & KalmanFilters::CKalmanCKF< SizeX, SizeY >::operator= (
    const CKalmanCKF< SizeX, SizeY > & other ) [inline]
```

Перегрузка оператора присвоения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

## 8.1.3.5 Prediction()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual void KalmanFilters::CKalmanCKF< SizeX, SizeY >::Prediction (
    double dt ) [inline], [virtual]
```

Прогноз

Аргументы

dt	- Время прогноза, [с]
----	-----------------------

Переопределяет метод предка [KalmanFilters::CKalmanEKF< SizeX, SizeY >](#).

Переопределяется в [KalmanFilters::CKalmanSRCKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanECKF< SizeX, SizeY >](#) и [KalmanFilters::CKalmanSRECKF< SizeX, SizeY >](#).

## 8.1.3.6 PredictionCKF()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanCKF< SizeX, SizeY >::PredictionCKF (
    double dt ) [inline], [protected]
```

Прогноз CKF.

## Аргументы

dt	- Время прогноза, [с]
----	-----------------------

## 8.1.3.7 SetObservationJacobianH()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanCKF< SizeX, SizeY >::SetObservationJacobianH ( ) [inline], [protected]
```

Запрет доступа

## 8.1.3.8 SetStateTransitionJacobianF()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanCKF< SizeX, SizeY >::SetStateTransitionJacobianF ( ) [inline], [protected]
```

Запрет доступа

## 8.1.3.9 SetupDesignParametersCubatureBaseSet()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanCKF< SizeX, SizeY >::SetupDesignParametersCubatureBaseSet ( ) [inline], [protected]
```

Установка кубатурных весов (базовый вариант ансцентного преобразования)

Смотри [1] и [2]

## 8.1.3.10 SetWeightedSumMeasurementSigmas()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanCKF< SizeX, SizeY >::SetWeightedSumMeasurementSigmas (
    std::function< arma::vec(const arma::vec &weights, const arma::mat &sigmaPoints)> weightedSum←
    MeasurementSigmas ) [inline]
```

Установка функции вычисления взвешенной суммы сигма-точек пространства Y.

## Аргументы

weightedSumMeasurementSigmas	- Функция вычисления взвешенной суммы сигма-точек пространства Y
------------------------------	--

## 8.1.3.11 SetWeightedSumStateSigmas()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanCKF< SizeX, SizeY >::SetWeightedSumStateSigmas (
    std::function< arma::vec(const arma::vec &weights, const arma::mat &sigmaPoints)> weightedSumState←
    Sigmas ) [inline]
```

Установка функции вычисления взвешенной суммы сигма-точек пространства X

## Аргументы

weightedSumStateSigmas	- Функция вычисления взвешенной суммы сигма-точек пространства X
------------------------	--

## 8.1.4 Документация по друзьям класса и функциям, относящимся к классу

## 8.1.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    CKalmanCKF< SizeX, SizeY > & lhs,
    CKalmanCKF< SizeX, SizeY > & rhs ) [friend]
```

Метод свапа

Аргументы

lhs	- left hand side instance
rhs	- right hand side instance

## 8.1.5 Данные класса

## 8.1.5.1 dXcal\_

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanCKF< SizeX, SizeY >::dXcal_ [protected]
```

Матрица X-каллиграфическое (матрица сигма-точек - столбцов)

## 8.1.5.2 dYcal\_

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanCKF< SizeX, SizeY >::dYcal_ [protected]
```

Матрица Y-каллиграфическое (матрица сигма-точек - столбцов)

## 8.1.5.3 gamma\_

```
template<size_t SizeX, size_t SizeY>
double KalmanFilters::CKalmanCKF< SizeX, SizeY >::gamma_ [protected]
```

Автоматически вычисляемый (в методах SetDesignParameters\*) параметр (множитель при корне из P при создании сигма-точек)

## 8.1.5.4 k\_sigma\_points\_

```
template<size_t SizeX, size_t SizeY>
int KalmanFilters::CKalmanCKF< SizeX, SizeY >::k_sigma_points_ [protected]
```

Число сигма-точек

## 8.1.5.5 P\_xy\_

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanCKF< SizeX, SizeY >::P_xy_ [protected]
```

Матрица кросс-коварации векторов X и Y, размерность [SizeX \* SizeY].

## 8.1.5.6 sqrt\_P\_chol\_

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanCKF< SizeX, SizeY >::sqrt_P_chol_ [protected]
```

Корень из матрицы P.

### 8.1.5.7 weightedSumMeasurementSigmas\_

```
template<size_t SizeX, size_t SizeY>
std::function<arma::vec( const arma::vec &weights, const arma::mat &sigmaPoints )> KalmanFilters::CKalmanCKF<
SizeX, SizeY >::weightedSumMeasurementSigmas_ [protected]
```

Вычисление взвешенной суммы сигма-точек пространства Y.

### 8.1.5.8 weightedSumStateSigmas\_

```
template<size_t SizeX, size_t SizeY>
std::function<arma::vec( const arma::vec &weights, const arma::mat &sigmaPoints )> KalmanFilters::CKalmanCKF<
SizeX, SizeY >::weightedSumStateSigmas_ [protected]
```

Вычисление взвешенной суммы сигма-точек пространства X

### 8.1.5.9 weights\_covariance\_

```
template<size_t SizeX, size_t SizeY>
arma::vec KalmanFilters::CKalmanCKF< SizeX, SizeY >::weights_covariance_ [protected]
```

Веса ковариации

### 8.1.5.10 weights\_mean\_

```
template<size_t SizeX, size_t SizeY>
arma::vec KalmanFilters::CKalmanCKF< SizeX, SizeY >::weights_mean_ [protected]
```

Веса среднего

### 8.1.5.11 x\_est\_sigma\_points\_

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanCKF< SizeX, SizeY >::x_est_sigma_points_ [protected]
```

Матрица сигма-точек (сигма-точки - столбцы) в пространстве X на текущем такте, размерность [SizeX,k\_sigma\_points\_].

### 8.1.5.12 x\_pred\_sigma\_points\_

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanCKF< SizeX, SizeY >::x_pred_sigma_points_ [protected]
```

Матрица сигма-точек (сигма-точки - столбцы) в пространстве X, экстраполированный на текущий такт, размерность [SizeX,k\_sigma\_points\_].

### 8.1.5.13 y\_pred\_sigma\_points\_

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanCKF< SizeX, SizeY >::y_pred_sigma_points_ [protected]
```

Матрица сигма-точек (сигма-точки - столбцы) в пространстве Y, экстраполированный на текущий такт, размерность [SizeX,k\_sigma\_points\_].

Объявления и описания членов класса находятся в файле:

- kalman\_filter\_cubature.h

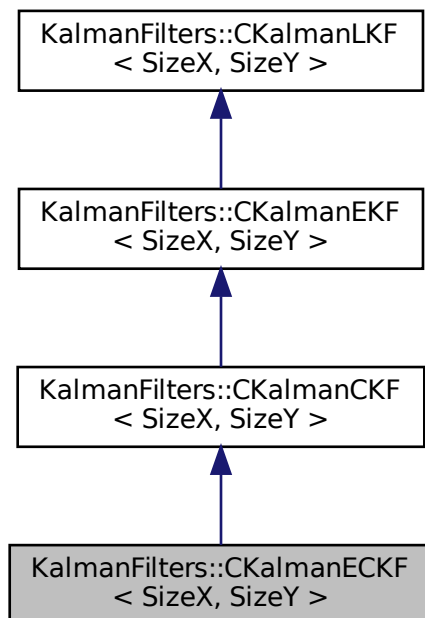
## 8.2 Шаблон класса KalmanFilters::CKalmanECKF< SizeX, SizeY >

Шаблонный класс расширенного кубатурного фильтра Калмана, РКФК (Extended Cubature Kalman Filter, ECKF)

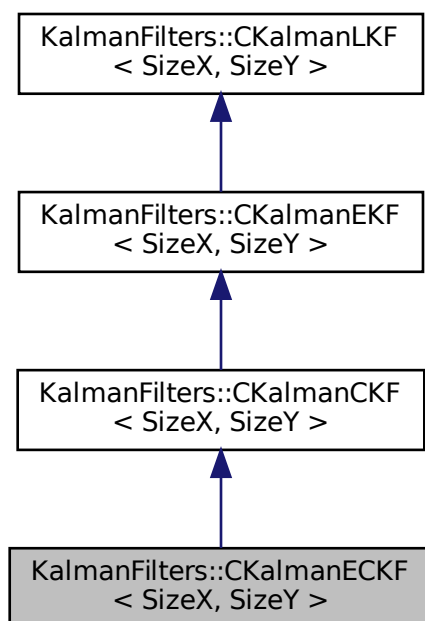


```
#include <kalman_filter_extended_cubature.h>
```

Граф наследования: KalmanFilters::CKalmanECKF< SizeX, SizeY >:



Граф связей класса `KalmanFilters::CKalmanECKF < SizeX, SizeY >`:



#### Открытые члены

- [CKalmanECKF](#) ()  
Конструктор по умолчанию
- [CKalmanECKF](#) (const [CKalmanECKF](#) &other)  
Конструктор копирования
- [CKalmanECKF](#) & [operator=](#) (const [CKalmanECKF](#) &other)  
Перегрузка оператора присвоения
- [CKalmanECKF](#) ([CKalmanECKF](#) &&other) noexcept  
Конструктор перемещения
- [CKalmanECKF](#) & [operator=](#) ([CKalmanECKF](#) &&other) noexcept  
Перегрузка оператора перемещения
- virtual [~CKalmanECKF](#) ()=default  
Деструктор
- void [SetStateTransitionJacobianF](#) (std::function< arma::mat(const arma::vec &X, double dt)> stateTransitionJacobianF)  
Установка функции вычисления матрицы перехода состояния F (makeMatrixF)
- void [SetObservationJacobianH](#) (std::function< arma::mat(const arma::vec &X)> observationJacobianH)  
Установка функции вычисления матрицы перехода измерений H (makeMatrixH)
- virtual void [Prediction](#) (double dt)  
Прогноз расширенного кубатурного фильтра Калмана (РКФК, ЕСКФ)
- virtual void [Correction](#) (const arma::vec &Y\_msd)  
Коррекция расширенного кубатурного фильтра Калмана (РКФК, ЕСКФ)

## Друзья

- void [swap](#) ([CKalmanECKF](#)< SizeX, SizeY > &lhs, [CKalmanECKF](#)< SizeX, SizeY > &rhs)  
noexcept  
Метод свапа

## Дополнительные унаследованные члены

## 8.2.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanECKF< SizeX, SizeY >
```

Шаблонный класс расширенного кубатурного фильтра Калмана, РКФК (Extended Cubature Kalman Filter, ECKF)

Источники:

[1] Cubature Kalman Filters, Ienkararan Arasaratnam and Simon Haykin, Life Fellow, IEEE

Внимание

Фильтр построен по классическому его варианту НИЖНЕтреугольного разложения Холецкого!

## Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

## 8.2.2 Конструктор(ы)

## 8.2.2.1 CKalmanECKF() [1/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanECKF< SizeX, SizeY >::CKalmanECKF ( ) [inline]
Конструктор по умолчанию
```

## 8.2.2.2 CKalmanECKF() [2/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanECKF< SizeX, SizeY >::CKalmanECKF (
    const CKalmanECKF< SizeX, SizeY > & other ) [inline]
Конструктор копирования
```

Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.2.2.3 CKalmanECKF() [3/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanECKF< SizeX, SizeY >::CKalmanECKF (
    CKalmanECKF< SizeX, SizeY > && other ) [inline], [noexcept]
Конструктор перемещения
```

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.2.2.4 ~CKalmanECKF()

```
template<size_t SizeX, size_t SizeY>
virtual KalmanFilters::CKalmanECKF< SizeX, SizeY >::~~CKalmanECKF ( ) [virtual], [default]
```

Деструктор

## 8.2.3 Методы

## 8.2.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanECKF< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd ) [inline], [virtual]
```

Коррекция расширенного кубатурного фильтра Калмана (РКФК, ЕСКФ)

## Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanCKF](#)< SizeX, SizeY >.

## 8.2.3.2 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
CKalmanECKF & KalmanFilters::CKalmanECKF< SizeX, SizeY >::operator= (
    CKalmanECKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Перегрузка оператора перемещения

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

## 8.2.3.3 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
CKalmanECKF & KalmanFilters::CKalmanECKF< SizeX, SizeY >::operator= (
    const CKalmanECKF< SizeX, SizeY > & other ) [inline]
```

Перегрузка оператора присвоения

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

#### 8.2.3.4 Prediction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanECKF< SizeX, SizeY >::Prediction (
    double dt ) [inline], [virtual]
```

Прогноз расширенного кубатурного фильтра Калмана (РКФК, ЕСКФ)

Аргументы

dt	- Время прогноза, [с]
----	-----------------------

Переопределяет метод предка [KalmanFilters::CKalmanCKF< SizeX, SizeY >](#).

#### 8.2.3.5 SetObservationJacobianH()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanECKF< SizeX, SizeY >::SetObservationJacobianH (
    std::function< arma::mat(const arma::vec &X)> observationJacobianH ) [inline]
Установка функции вычисления матрицы перехода измерений H (makeMatrixH)
```

См. также

[observationJacobianH\\_](#)

#### 8.2.3.6 SetStateTransitionJacobianF()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanECKF< SizeX, SizeY >::SetStateTransitionJacobianF (
    std::function< arma::mat(const arma::vec &X, double dt)> stateTransitionJacobianF ) [inline]
Установка функции вычисления матрицы перехода состояния F (makeMatrixF)
```

См. также

[stateTransitionJacobianF\\_](#)

### 8.2.4 Документация по друзьям класса и функциям, относящимся к классу

#### 8.2.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    CKalmanECKF< SizeX, SizeY > & lhs,
    CKalmanECKF< SizeX, SizeY > & rhs ) [friend]
```

Метод свапа

Аргументы

lhs	- left hand side instance
rhs	- right hand side instance

Объявления и описания членов класса находятся в файле:

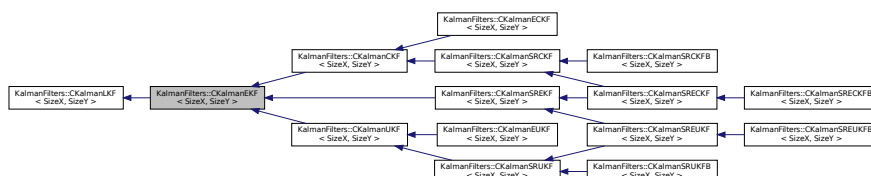
- `kalman_filter_extended_cubature.h`

### 8.3 Шаблон класса `KalmanFilters::CKalmanEKF< SizeX, SizeY >`

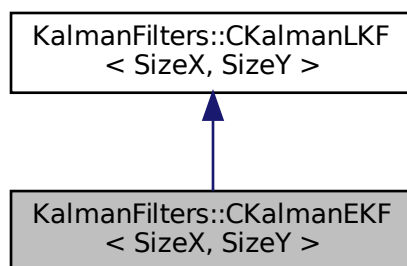
Класс расширенного фильтра Калмана, РФК (Extended Kalman Filter, EKF)

`#include <kalman_filter_extended.h>`

Граф наследования: `KalmanFilters::CKalmanEKF< SizeX, SizeY >`:



Граф связей класса `KalmanFilters::CKalmanEKF< SizeX, SizeY >`:



Открытые члены

- `CKalmanEKF ()`  
Конструктор по умолчанию
- `CKalmanEKF (const CKalmanEKF &other)`  
Конструктор копирования
- `CKalmanEKF & operator= (const CKalmanEKF &other)`  
Перегрузка оператора присвоения
- `CKalmanEKF (CKalmanEKF &&other) noexcept`  
Конструктор перемещения
- `CKalmanEKF & operator= (CKalmanEKF &&other) noexcept`  
Перегрузка оператора перемещения
- `virtual ~CKalmanEKF ()=default`  
Деструктор
- `void SetStateTransitionModel (std::function< arma::vec(const arma::vec &X, double dt)> state←TransitionModel)`  
Установка функции прогноза состояния (`predictState`)
- `void SetObservationModel (std::function< arma::vec(const arma::vec &X)> observationModel)`

- Установка функции перевода состояния в измерение (XtoY)
- void [SetStateTransitionJacobianF](#) (std::function< arma::mat(const arma::vec &X, double dt)> stateTransitionJacobianF)
- Установка функции вычисления матрицы перехода состояния F (makeMatrixF)
- void [SetObservationJacobianH](#) (std::function< arma::mat(const arma::vec &X)> observationJacobianH)
- Установка функции вычисления матрицы перехода измерений H (makeMatrixH)
- virtual void [Prediction](#) (double dt)
- Прогноз
- virtual void [Correction](#) (const arma::vec &Y\_msd)
- Коррекция

#### Защищенные члены

- void [PredictionEKF](#) (double dt)
- Прогноз EKF.

#### Защищенные данные

- std::function< arma::vec(const arma::vec &X, double dt)> [stateTransitionModel\\_](#)
- Функция прогноза состояния (predictState)
- std::function< arma::vec(const arma::vec &X)> [observationModel\\_](#)
- Функция перевода состояния в измерения (XtoY)
- std::function< arma::mat(const arma::vec &X, double dt)> [stateTransitionJacobianF\\_](#)
- Функция вычисления матрицы перехода состояния F (makeMatrixF)
- std::function< arma::mat(const arma::vec &X)> [observationJacobianH\\_](#)
- Функция вычисления матрицы перехода измерений H (makeMatrixH)

#### Закрытые члены

- void [SetStateTransitionJacobianLinearF](#) ()
- void [SetStateTransitionMatrixF](#) ()
- Запрет доступа
- void [SetObservationMatrixH](#) ()
- Запрет доступа

#### Друзья

- void [swap](#) (CKalmanEKF< SizeX, SizeY > &lhs, CKalmanEKF< SizeX, SizeY > &rhs) noexcept
- Метод свапа

#### 8.3.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanEKF< SizeX, SizeY >
```

Класс расширенного фильтра Калмана, РФК (Extended Kalman Filter, EKF)

Источник: NASA Technical report R-135, Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle, Gerald L. Smith, Stanley F. Schmidt and Leonard A. McGee, 1962

Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

### 8.3.2 Конструктор(ы)

#### 8.3.2.1 CKalmanEKF() [1/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanEKF< SizeX, SizeY >::CKalmanEKF ( ) [inline]
```

Конструктор по умолчанию

#### 8.3.2.2 CKalmanEKF() [2/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanEKF< SizeX, SizeY >::CKalmanEKF (
    const CKalmanEKF< SizeX, SizeY > & other ) [inline]
```

Конструктор копирования

Аргументы

other	- экземпляр, с которого делается копия
-------	--

#### 8.3.2.3 CKalmanEKF() [3/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanEKF< SizeX, SizeY >::CKalmanEKF (
    CKalmanEKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Конструктор перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

#### 8.3.2.4 ~CKalmanEKF()

```
template<size_t SizeX, size_t SizeY>
virtual KalmanFilters::CKalmanEKF< SizeX, SizeY >::~~CKalmanEKF ( ) [virtual], [default]
```

Деструктор

### 8.3.3 Методы

#### 8.3.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanEKF< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd ) [inline], [virtual]
```

Коррекция

Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanLKF< SizeX, SizeY >](#).



Переопределяется в [KalmanFilters::CKalmanCKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRCKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRCKFB< SizeX, SizeY >](#), [KalmanFilters::CKalmanECKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRECKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >](#), [KalmanFilters::CKalmanSREKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanEUKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSREUKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >](#), [KalmanFilters::CKalmanUKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRUKF< SizeX, SizeY >](#) и [KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >](#).

#### 8.3.3.2 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanEKF & KalmanFilters::CKalmanEKF< SizeX, SizeY >::operator= (
    CKalmanEKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Перегрузка оператора перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

#### 8.3.3.3 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanEKF & KalmanFilters::CKalmanEKF< SizeX, SizeY >::operator= (
    const CKalmanEKF< SizeX, SizeY > & other ) [inline]
```

Перегрузка оператора присвоения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

#### 8.3.3.4 Prediction()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual void KalmanFilters::CKalmanEKF< SizeX, SizeY >::Prediction (
    double dt ) [inline], [virtual]
```

Прогноз

Аргументы

dt	- Время прогноза, [с]
----	-----------------------

Переопределяет метод предка [KalmanFilters::CKalmanLKF< SizeX, SizeY >](#).

Переопределяется в [KalmanFilters::CKalmanCKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRCKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanECKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRECKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSREKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanEUKF< SizeX, SizeY >](#),

[KalmanFilters::CKalmanSREUKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanUKF< SizeX, SizeY >](#) и [KalmanFilters::CKalmanSRUKF< SizeX, SizeY >](#).

#### 8.3.3.5 PredictionEKF()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanEKF< SizeX, SizeY >::PredictionEKF (
    double dt ) [inline], [protected]
```

Прогноз EKF.

Аргументы

dt	- Время прогноза, [с]
----	-----------------------

#### 8.3.3.6 SetObservationJacobianH()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanEKF< SizeX, SizeY >::SetObservationJacobianH (
    std::function< arma::mat(const arma::vec &X)> observationJacobianH ) [inline]
```

Установка функции вычисления матрицы перехода измерений H (makeMatrixH)

См. также

[observationJacobianH\\_](#)

#### 8.3.3.7 SetObservationMatrixH()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanEKF< SizeX, SizeY >::SetObservationMatrixH ( ) [inline], [private]
```

Запрет доступа

#### 8.3.3.8 SetObservationModel()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanEKF< SizeX, SizeY >::SetObservationModel (
    std::function< arma::vec(const arma::vec &X)> observationModel ) [inline]
```

Установка функции перевода состояния в измерение (XtoY)

См. также

[observationModel\\_](#)

#### 8.3.3.9 SetStateTransitionJacobianF()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanEKF< SizeX, SizeY >::SetStateTransitionJacobianF (
    std::function< arma::mat(const arma::vec &X, double dt)> stateTransitionJacobianF ) [inline]
```

Установка функции вычисления матрицы перехода состояния F (makeMatrixF)

См. также

[stateTransitionJacobianF\\_](#)

## 8.3.3.10 SetStateTransitionJacobianLinearF()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanEKF< SizeX, SizeY >::SetStateTransitionJacobianLinearF ( ) [inline], [private]
```

## 8.3.3.11 SetStateTransitionMatrixF()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanEKF< SizeX, SizeY >::SetStateTransitionMatrixF ( ) [inline], [private]
Запрет доступа
```

## 8.3.3.12 SetStateTransitionModel()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanEKF< SizeX, SizeY >::SetStateTransitionModel (
    std::function< arma::vec(const arma::vec &X, double dt)> stateTransitionModel ) [inline]
Установка функции прогноза состояния (predictState)
```

См. также

[stateTransitionModel\\_](#)

## 8.3.4 Документация по друзьям класса и функциям, относящимся к классу

## 8.3.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    CKalmanEKF< SizeX, SizeY > & lhs,
    CKalmanEKF< SizeX, SizeY > & rhs ) [friend]
```

Метод свапа

Аргументы

lhs	- left hand side instance
rhs	- right hand side instance

## 8.3.5 Данные класса

## 8.3.5.1 observationJacobianH\_

```
template<size_t SizeX, size_t SizeY>
std::function<arma::mat( const arma::vec &X )> KalmanFilters::CKalmanEKF< SizeX, SizeY >::observationJacobian↔
H_ [protected]
```

Функция вычисления матрицы перехода измерений H (makeMatrixH)

Аргументы

X	- вектор состояния текущего момента времени
---	---

См. также

[SetObservationJacobianH](#)

#### 8.3.5.2 observationModel\_

```
template<size_t SizeX, size_t SizeY>
std::function<arma::vec( const arma::vec &X )> KalmanFilters::CKalmanEKF< SizeX, SizeY >::observationModel_ ←
[protected]
```

Функция перевода состояния в измерения (XtoY)

Аргументы

X	- вектор состояния текущего момента времени
---	---

См. также

[SetObservationModel](#)

#### 8.3.5.3 stateTransitionJacobianF\_

```
template<size_t SizeX, size_t SizeY>
std::function<arma::mat( const arma::vec &X, double dt )> KalmanFilters::CKalmanEKF< SizeX, SizeY >::state←
TransitionJacobianF_ [protected]
```

Функция вычисления матрицы перехода состояния F (makeMatrixF)

Аргументы

X	- вектор состояния с прошлого момента времени
---	---

См. также

[SetStateTransitionJacobianF](#)

#### 8.3.5.4 stateTransitionModel\_

```
template<size_t SizeX, size_t SizeY>
std::function<arma::vec( const arma::vec &X, double dt )> KalmanFilters::CKalmanEKF< SizeX, SizeY >::state←
TransitionModel_ [protected]
```

Функция прогноза состояния (predictState)

Аргументы

X	- вектор состояния с прошлого момента времени
---	---

См. также

[SetStateTransitionModel](#)

Объявления и описания членов класса находятся в файле:

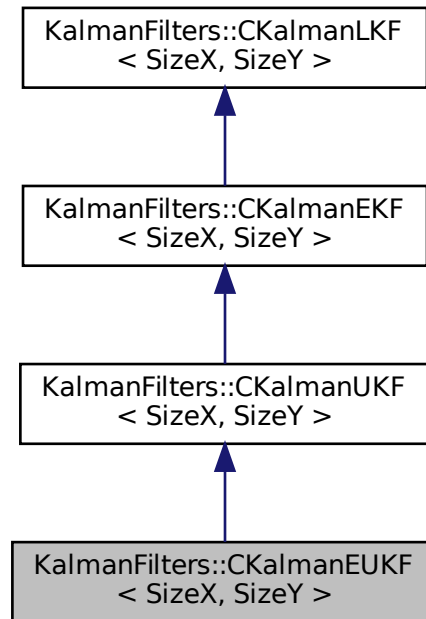
- kalman\_filter\_extended.h

## 8.4 Шаблон класса KalmanFilters::CKalmanEUKF< SizeX, SizeY >

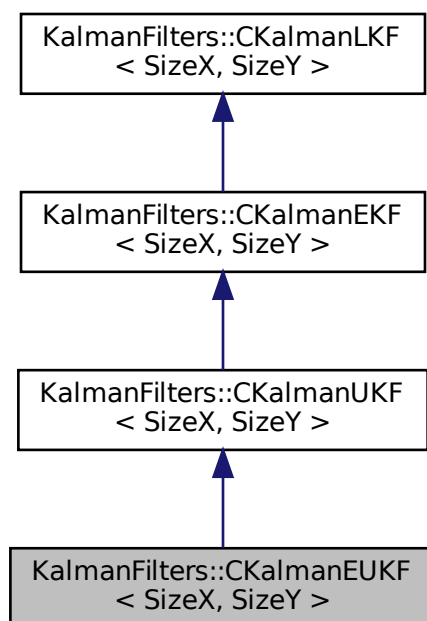
Шаблонный класс расширенного сигма-точечного (ансцентного) фильтра Калмана, РСТФК (Extended Unscented Kalman Filter, EUKF)

#include <kalman\_filter\_extended\_unscented.h>

Граф наследования: KalmanFilters::CKalmanEUKF< SizeX, SizeY >:



Граф связей класса `KalmanFilters::CKalmanEUKF < SizeX, SizeY >`:



#### Открытые члены

- [CKalmanEUKF](#) ()  
Конструктор по умолчанию
- [CKalmanEUKF](#) (const [CKalmanEUKF](#) &other)  
Конструктор копирования
- [CKalmanEUKF](#) & [operator=](#) (const [CKalmanEUKF](#) &other)  
Перегрузка оператора присвоения
- [CKalmanEUKF](#) ([CKalmanEUKF](#) &&other) noexcept  
Конструктор перемещения
- [CKalmanEUKF](#) & [operator=](#) ([CKalmanEUKF](#) &&other) noexcept  
Перегрузка оператора перемещения
- virtual [~CKalmanEUKF](#) ()=default  
Деструктор
- void [SetStateTransitionJacobianF](#) (std::function< arma::mat(const arma::vec &X, double dt)> stateTransitionJacobianF)  
Установка функции вычисления матрицы перехода состояния F (makeMatrixF)
- void [SetObservationJacobianH](#) (std::function< arma::mat(const arma::vec &X)> observationJacobianH)  
Установка функции вычисления матрицы перехода измерений H (makeMatrixH)
- virtual void [Prediction](#) (double dt)  
Прогноз расширенного сигма-точечного фильтра Калмана (РСТФК, EUKF)
- virtual void [Correction](#) (const arma::vec &Y\_msd)  
Коррекция расширенного сигма-точечного фильтра Калмана (РСТФК, EUKF)

## Друзья

- void [swap](#) ([CKalmanEUKF](#)< SizeX, SizeY > &lhs, [CKalmanEUKF](#)< SizeX, SizeY > &rhs)  
noexcept  
Метод свапа

## Дополнительные унаследованные члены

## 8.4.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanEUKF< SizeX, SizeY >
```

Шаблонный класс расширенного сигма-точечного (ансцентного) фильтра Калмана, РСТФК (Extended Unscented Kalman Filter, EUKF)

Источники:

- [1] A New Extension of the Kalman Filter to Nonlinear Systems, Simon J. Julier, Jeffrey K. Uhlmann, The Robotics Research Group, Department of Engineering Science, The University of Oxford, 1997
- [2] The Unscented Kalman Filter for Nonlinear Estimation, Eric A. Wan and Rudolph van der Merwe Oregon Graduate Institute of Science & Technology 20000 NW Walker Rd, Beaverton, Oregon 97006, 2000 ericwan[at]ece.ogi.edu, rvdmerwe[at]ece.ogi.edu
- [3] Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models, Rudolph van der Merwe & Eric Wan, OGI School of Science & Engineering Oregon Health & Science University Beaverton, Oregon, 97006, USA, 2003 {rvdmerwe,ericwan}[at]ece.ogi.edu
- [4] THE SQUARE-ROOT UNSCENTED KALMAN FILTER FOR STATE AND PARAMETER-ESTIMATION, Rudolph van der Merwe and Eric A. Wan, Oregon Graduate Institute of Science and Technology 20000 NW Walker Road, Beaverton, Oregon 97006, USA rvdmerwe,ericwan[at]ece.ogi.edu

## Внимание

Внимание! В реализации UKF вес нулевой сигма-точки Wcov не может быть отрицательным! (Wmean - может)

Фильтр построен по классическому его варианту НИЖНЕтреугольного разложения Холецкого!

## Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

## 8.4.2 Конструктор(ы)

## 8.4.2.1 CKalmanEUKF() [1/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanEUKF< SizeX, SizeY >::CKalmanEUKF ( ) [inline]
Конструктор по умолчанию
```

## 8.4.2.2 CKalmanEUKF() [2/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanEUKF< SizeX, SizeY >::CKalmanEUKF (
    const CKalmanEUKF< SizeX, SizeY > & other ) [inline]
Конструктор копирования
```

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.4.2.3 CKalmanEUKF() [3/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanEUKF< SizeX, SizeY >::CKalmanEUKF (
    CKalmanEUKF< SizeX, SizeY > && other )    [inline], [noexcept]
```

Конструктор перемещения

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.4.2.4 ~CKalmanEUKF()

```
template<size_t SizeX, size_t SizeY>
virtual KalmanFilters::CKalmanEUKF< SizeX, SizeY >::~CKalmanEUKF ( )    [virtual], [default]
```

Деструктор

## 8.4.3 Методы

## 8.4.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanEUKF< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd )    [inline], [virtual]
```

Коррекция расширенного сигма-точечного фильтра Калмана (РСТФК, EUKF)

## Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanUKF< SizeX, SizeY >](#).

## 8.4.3.2 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
CKalmanEUKF & KalmanFilters::CKalmanEUKF< SizeX, SizeY >::operator= (
    CKalmanEUKF< SizeX, SizeY > && other )    [inline], [noexcept]
```

Перегрузка оператора перемещения

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this



## 8.4.3.3 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanEUKF & KalmanFilters::CKalmanEUKF< SizeX, SizeY >::operator= (
    const CKalmanEUKF< SizeX, SizeY > & other ) [inline]
```

Перегрузка оператора присвоения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

## 8.4.3.4 Prediction()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual void KalmanFilters::CKalmanEUKF< SizeX, SizeY >::Prediction (
    double dt ) [inline], [virtual]
```

Прогноз расширенного сигма-точечного фильтра Калмана (РСТФК, EUKF)

Аргументы

dt	- Время прогноза, [с]
----	-----------------------

Переопределяет метод предка [KalmanFilters::CKalmanUKF< SizeX, SizeY >](#).

## 8.4.3.5 SetObservationJacobianH()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanEUKF< SizeX, SizeY >::SetObservationJacobianH (
    std::function< arma::mat(const arma::vec &X)> observationJacobianH ) [inline]
```

Установка функции вычисления матрицы перехода измерений H (makeMatrixH)

См. также

[observationJacobianH\\_](#)

## 8.4.3.6 SetStateTransitionJacobianF()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanEUKF< SizeX, SizeY >::SetStateTransitionJacobianF (
    std::function< arma::mat(const arma::vec &X, double dt)> stateTransitionJacobianF ) [inline]
```

Установка функции вычисления матрицы перехода состояния F (makeMatrixF)

См. также

[stateTransitionJacobianF\\_](#)

## 8.4.4 Документация по друзьям класса и функциям, относящимся к классу



- Установка ковариационной матрицы Q шумов состояния X.
- void [SetProcessCovarianceMatrixQdiag](#) (const arma::vec &Qdiag)
- Установка диагонали ковариационной матрицы Q шумов состояния X.
- void [SetObservationCovarianceMatrixR](#) (const arma::mat &R)
- Установка ковариационной матрицы R шумов измерений Y.
- void [SetObservationCovarianceMatrixRdiag](#) (const arma::vec &Rdiag)
- Установка ковариационной матрицы R шумов измерений Y.
- void [SetEstimatedVectorX](#) (const arma::vec &X\_est)
- Установка оценки вектора состояния X.
- void [SetEstimatedVectorY](#) (const arma::vec &Y\_est)
- Установка оценки вектора состояния Y.
- void [SetMeasuredVectorY](#) (const arma::vec &Y\_msd)
- Установка измеренного вектора измерений Y.
- void [SetDeltaY](#) (const arma::vec &DeltaY)
- Установка вектора невязки измерений DeltaY.
- void [SetCheckBordersStateAfterPrediction](#) (std::function< arma::vec(const arma::vec &X)> checkBordersStateAfterPrediction)
- Установка функции проверки вектора состояния X после прогноза
- void [SetCheckBordersStateAfterCorrection](#) (std::function< arma::vec(const arma::vec &X)> checkBordersStateAfterCorrection)
- Установка функции проверки вектора состояния X после коррекции
- void [SetCheckBordersMeasurement](#) (std::function< arma::vec(const arma::vec &Y)> check← BordersMeasurement)
- Установка функции проверки вектора измерений Y.
- void [SetCheckDeltaState](#) (std::function< arma::vec(const arma::vec &DeltaX)> checkDeltaState)
- Установка функции проверки разности векторов состояний X.
- void [SetCheckDeltaMeasurement](#) (std::function< arma::vec(const arma::vec &DeltaY)> check← DeltaMeasurement)
- Установка функции проверки разности векторов измерений Y.
- void [SetStateTransitionJacobianLinearF](#) (std::function< arma::mat(double dt)> stateTransition← JacobianLinearF)
- Установка функции вычисления матрицы перехода состояния F в случае LKF (makeMatrixF)
- const int [GetSizeX](#) () const
- Получить размерность вектора состояния X.
- const int [GetSizeY](#) () const
- Получить размерность вектора состояния Y.
- const arma::mat & [GetEstimatedCovarianceMatrixP](#) () const
- Получить ковариационную матрицу P состояния X.
- const arma::mat & [GetInnovationCovarianceMatrixS](#) () const
- Получить ковариационную матрицу S вектора невязки DeltaY.
- const arma::mat & [GetDeltaY](#) () const
- Получить вектор невязки измерений DeltaY.
- const arma::mat & [GetKalmanGainMatrixK](#) () const
- Получить матрицу коэффициентов усиления фильтра K.
- const arma::mat & [GetEstimatedVectorX](#) () const
- Получить уточненный вектор состояния X.
- const arma::mat & [GetEstimatedVectorY](#) () const
- Получить уточненный вектор состояния Y.
- virtual void [Prediction](#) (double dt)
- Прогноз
- virtual void [Correction](#) (const arma::vec &Y\_msd)
- Коррекция
- virtual void [CalculateInnovationCovarianceS](#) (const arma::vec &PdiagAdd, const arma::vec Rdiag)
- Отдельное вычисление ковариационной матрицы S невязки измерений

## Защищенные члены

- void [fixMatrixMainDiagonalSymmetry](#) (arma::mat &A)  
Исправление симметричности матрицы относительно главной диагонали
- void [checkMatrixDiagPositive](#) (const arma::mat &A) const  
Проверка что в диагонали матрицы лежат только положительные элементы

## Защищенные данные

- size\_t [SizeX\\_](#)  
Размерность вектора состояния X (state)
- size\_t [SizeY\\_](#)  
Размерность вектора измерений Y (measurement)
- arma::mat [F\\_](#)  
Матрица эволюции системы (перехода состояния) (state-transition model), размерность [SizeX \* SizeX].
- arma::mat [H\\_](#)  
Матрица измерений (перехода измерений) (observation model), размерность [SizeY \* SizeX].
- arma::mat [K\\_](#)  
Коэффициент усиления фильтра Калмана (Kalman gain), размерность [SizeX \* SizeY].
- arma::mat [I\\_](#)  
Единичная матрица, размерность [SizeX \* SizeX].
- arma::vec [DeltaY\\_](#)  
Вектор невязки измерений, размерность [SizeY].
- arma::mat [P\\_](#)  
Ковариационная матрица вектора состояния X (estimate covariance matrix), размерность [SizeX \* SizeX].
- arma::mat [S\\_](#)  
Ковариационная матрица вектора невязки (innovation covariance), размерность [SizeY \* SizeY].
- arma::mat [Q\\_](#)  
Ковариационная матрица (обычно диагональная) шумов вектора состояния X НА 1 СЕКУНДЕ (covariance of the process noise), размерность [SizeX \* SizeX].
- arma::mat [R\\_](#)  
Ковариационная матрица (обычно диагональная) шумов вектора измерений Y (covariance of the observation noise), размерность [SizeY \* SizeY].
- arma::vec [X\\_pred\\_](#)  
Экстраполированный (predicted) вектор состояния X, размерность [SizeX].
- arma::vec [Y\\_pred\\_](#)  
Экстраполированный (predicted) вектор измерений Y, размерность [SizeY].
- arma::vec [X\\_est\\_](#)  
Скорректированный (estimated) вектор состояния X, размерность [SizeX].
- arma::vec [Y\\_est\\_](#)  
Скорректированный (estimated) вектор измерений Y, размерность [SizeY].
- arma::vec [Y\\_msd\\_](#)  
Измеренный (measured) вектор Y (отметка), размерность [SizeY].
- bool [deltaY\\_isSet](#)  
Признак установки вектора невязки DeltaY (нужно в методе Correction)
- bool [Y\\_msd\\_isSet](#)  
Признак установки вектора измерений Y\_msd;.
- bool [prediction\\_isDone](#)  
Признак выполненного прогноза (без этого нельзя делать фильтрацию)
- std::function< arma::vec(const arma::vec &X)> [checkBordersStateAfterPrediction\\_](#)  
Проверка границ вектора состояния X после прогноза

- `std::function< arma::vec(const arma::vec &X)>` [checkBordersStateAfterCorrection\\_](#)  
Проверка границ вектора состояния X после фильтрации
- `std::function< arma::vec(const arma::vec &Y)>` [checkBordersMeasurement\\_](#)  
Проверка границ вектора измерения Y.
- `std::function< arma::vec(const arma::vec &DeltaX)>` [checkDeltaState\\_](#)  
Проверка разности векторов состояния X.
- `std::function< arma::vec(const arma::vec &DeltaY)>` [checkDeltaMeasurement\\_](#)  
Проверка разности векторов измерения Y.

## Закрытые данные

- `std::function< arma::mat(double dt)>` [stateTransitionJacobianLinearF\\_](#)  
Функция вычисления матрицы перехода состояния F (makeMatrixF)

## Друзья

- `void swap (CKalmanLKF< SizeX, SizeY > &lhs, CKalmanLKF< SizeX, SizeY > &rhs)` `noexcept`  
Метод свапа

### 8.5.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanLKF< SizeX, SizeY >
```

Шаблонный класс линейного фильтра Калмана, ЛФК (Linear Kalman Filter, LKF)

Источник: A New Approach to Linear Filtering and Prediction Problems, R.E.KALMAN, Research Institute for Advanced Study, Baltimore, Md., 1960

Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

### 8.5.2 Конструктор(ы)

#### 8.5.2.1 CKalmanLKF() [1/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanLKF< SizeX, SizeY >::CKalmanLKF ( ) [inline]
Конструктор по умолчанию
```

#### 8.5.2.2 CKalmanLKF() [2/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanLKF< SizeX, SizeY >::CKalmanLKF (
    const CKalmanLKF< SizeX, SizeY > & other ) [inline]
Конструктор копирования
```

Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.5.2.3 CKalmanLKF() [3/3]

```
template<size_t SizeX, size_t SizeY>
```

```
KalmanFilters::CKalmanLKF< SizeX, SizeY >::CKalmanLKF (
    CKalmanLKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Конструктор перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.5.2.4 ~CKalmanLKF()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual KalmanFilters::CKalmanLKF< SizeX, SizeY >::~~CKalmanLKF ( ) [virtual], [default]
```

Деструктор

## 8.5.3 Методы

## 8.5.3.1 CalculateInnovationCovarianceS()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual void KalmanFilters::CKalmanLKF< SizeX, SizeY >::CalculateInnovationCovarianceS (
    const arma::vec & PdiagAdd,
    const arma::vec Rdiag ) [inline], [virtual]
```

Отдельное вычисление ковариационной матрицы S невязки измерений

Отдельное вычисление S применяется при стробировании

Внимание

Необходимо учесть 2 обстоятельства: 1) метод должен выполняться после вызова метода Prediction, где должна быть вычислена матрица H; 2) матрица R при вычислении S будет использована та, которая уже имеется в фильтре и если необходима иная матрица R, то следует её установить, вызвав метод SetObservationCovarianceMatrixR

Аргументы

PdiagAdd	- добавка в диагональ матрицы P, размерность [SizeX]
Rdiag	- диагональная матрица R априорных шумов измерений, размерность [SizeY]

## 8.5.3.2 checkMatrixDiagPositive()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::checkMatrixDiagPositive (
    const arma::mat & A ) const [inline], [protected]
```

Проверка что в диагонали матрицы лежат только положительные элементы

Аргументы

A	- проверяемая матрица
---	-----------------------

Внимание

Метод вызывает assert!

### 8.5.3.3 Correction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanLKF< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd ) [inline], [virtual]
```

Коррекция

Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяется в [KalmanFilters::CKalmanCKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRCKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRCKFB< SizeX, SizeY >](#), [KalmanFilters::CKalmanEKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanECKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRECKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >](#), [KalmanFilters::CKalmanSREKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanEUKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSREUKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >](#), [KalmanFilters::CKalmanUKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRUKF< SizeX, SizeY >](#) и [KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >](#).

### 8.5.3.4 fixMatrixMainDiagonalSymmetry()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::fixMatrixMainDiagonalSymmetry (
    arma::mat & A ) [inline], [protected]
```

Исправление симметричности матрицы относительно главной диагонали

Элементам вне главной диагонали присваивается полусумма между соответствующими элементами

Аргументы

A	- проверяемая матрица
---	-----------------------

### 8.5.3.5 GetDeltaY()

```
template<size_t SizeX, size_t SizeY>
const arma::mat & KalmanFilters::CKalmanLKF< SizeX, SizeY >::GetDeltaY ( ) const [inline]
```

Получить вектор невязки измерений DeltaY.

Возвращает

Вектор невязки измерений DeltaY

### 8.5.3.6 GetEstimatedCovarianceMatrixP()

```
template<size_t SizeX, size_t SizeY>
const arma::mat & KalmanFilters::CKalmanLKF< SizeX, SizeY >::GetEstimatedCovarianceMatrixP ( ) const [inline]
```

Получить ковариационную матрицу P состояния X.

Возвращает

Текущая ковариационная матрица P

#### 8.5.3.7 GetEstimatedVectorX()

```
template<size_t SizeX, size_t SizeY>
```

```
const arma::mat & KalmanFilters::CKalmanLKF< SizeX, SizeY >::GetEstimatedVectorX ( ) const [inline]
```

Получить уточненный вектор состояния X.

Возвращает

Текущий уточненный вектор состояния X

#### 8.5.3.8 GetEstimatedVectorY()

```
template<size_t SizeX, size_t SizeY>
```

```
const arma::mat & KalmanFilters::CKalmanLKF< SizeX, SizeY >::GetEstimatedVectorY ( ) const [inline]
```

Получить уточненный вектор состояния Y.

Возвращает

Текущий уточненный вектор состояния Y

#### 8.5.3.9 GetInnovationCovarianceMatrixS()

```
template<size_t SizeX, size_t SizeY>
```

```
const arma::mat & KalmanFilters::CKalmanLKF< SizeX, SizeY >::GetInnovationCovarianceMatrixS ( ) const [inline]
```

Получить ковариационную матрицу S вектора невязки DeltaY.

Возвращает

Текущая ковариационная матрица S

#### 8.5.3.10 GetKalmanGainMatrixK()

```
template<size_t SizeX, size_t SizeY>
```

```
const arma::mat & KalmanFilters::CKalmanLKF< SizeX, SizeY >::GetKalmanGainMatrixK ( ) const [inline]
```

Получить матрицу коэффициентов усиления фильтра K.

Возвращает

Текущая матрица коэффициентов усиления фильтра K

#### 8.5.3.11 GetSizeX()

```
template<size_t SizeX, size_t SizeY>
```

```
const int KalmanFilters::CKalmanLKF< SizeX, SizeY >::GetSizeX ( ) const [inline]
```

Получить размерность вектора состояния X.

Возвращает

Размерность вектора состояния X

#### 8.5.3.12 GetSizeY()

```
template<size_t SizeX, size_t SizeY>
```

```
const int KalmanFilters::CKalmanLKF< SizeX, SizeY >::GetSizeY ( ) const [inline]
```

Получить размерность вектора состояния Y.

Возвращает

Размерность вектора состояния Y



## 8.5.3.13 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanLKF & KalmanFilters::CKalmanLKF< SizeX, SizeY >::operator= (
    CKalmanLKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Перегрузка оператора перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

## 8.5.3.14 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanLKF & KalmanFilters::CKalmanLKF< SizeX, SizeY >::operator= (
    const CKalmanLKF< SizeX, SizeY > & other ) [inline]
```

Перегрузка оператора присвоения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

## 8.5.3.15 Prediction()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual void KalmanFilters::CKalmanLKF< SizeX, SizeY >::Prediction (
    double dt ) [inline], [virtual]
```

Прогноз

Аргументы

dt	- Время прогноза, [с]
----	-----------------------

Переопределяется в [KalmanFilters::CKalmanCKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRCKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanEKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanECKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRECKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSREKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanEUKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSREUKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanUKF< SizeX, SizeY >](#) и [KalmanFilters::CKalmanSRUKF< SizeX, SizeY >](#).

## 8.5.3.16 SetCheckBordersMeasurement()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetCheckBordersMeasurement (
    std::function< arma::vec(const arma::vec &Y)> checkBordersMeasurement ) [inline]
```

Установка функции проверки вектора измерений Y.

## Аргументы

checkBordersMeasurement	- Функция проверки вектора измерений
-------------------------	--------------------------------------

## 8.5.3.17 SetCheckBordersStateAfterCorrection()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetCheckBordersStateAfterCorrection (
    std::function< arma::vec(const arma::vec &X)> checkBordersStateAfterCorrection ) [inline]
```

Установка функции проверки вектора состояния X после коррекции

## Аргументы

checkBordersStateAfterCorrection	- Функция проверки вектора состояния после коррекции
----------------------------------	--

## 8.5.3.18 SetCheckBordersStateAfterPrediction()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetCheckBordersStateAfterPrediction (
    std::function< arma::vec(const arma::vec &X)> checkBordersStateAfterPrediction ) [inline]
```

Установка функции проверки вектора состояния X после прогноза

## Аргументы

checkBordersStateAfterPrediction	- Функция проверки вектора состояния после прогноза
----------------------------------	---

## 8.5.3.19 SetCheckDeltaMeasurement()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetCheckDeltaMeasurement (
    std::function< arma::vec(const arma::vec &DeltaY)> checkDeltaMeasurement ) [inline]
```

Установка функции проверки разности векторов измерений Y.

## Аргументы

checkDeltaMeasurement	- Функция проверки разности векторов измерений
-----------------------	--

## 8.5.3.20 SetCheckDeltaState()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetCheckDeltaState (
    std::function< arma::vec(const arma::vec &DeltaX)> checkDeltaState ) [inline]
```

Установка функции проверки разности векторов состояний X.

## Аргументы

checkDeltaState	- Функция проверки разности векторов состояний
-----------------	--

## 8.5.3.21 SetDeltaY()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetDeltaY (
    const arma::vec & DeltaY ) [inline]
```

Установка вектора невязки измерений DeltaY.

Аргументы

DeltaY	- вектор невязки измерений, размерность [SizeY]
--------	---

## 8.5.3.22 SetEstimateCovarianceMatrixP()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetEstimateCovarianceMatrixP (
    const arma::mat & P ) [inline]
```

Установка ковариационной матрицы P состояния X.

Аргументы

P	- матрица шумов состояния X, размерность [SizeX * SizeX]
---	--

## 8.5.3.23 SetEstimateCovarianceMatrixPdiag()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetEstimateCovarianceMatrixPdiag (
    const arma::vec & Pdiag ) [inline]
```

Установка диагонали ковариационной матрицы P состояния X.

Аргументы

Pdiag	- вектор главной диагонали шумов состояния X (остальные элементы полагаются равными нулю), размерность [SizeX]
-------	--

## 8.5.3.24 SetEstimatedVectorX()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetEstimatedVectorX (
    const arma::vec & X_est ) [inline]
```

Установка оценки вектора состояния X.

Требуется, например, при начальной установке фильтра

Аргументы

X_est	- вектор состояния, размерность [SizeX]
-------	---

## 8.5.3.25 SetEstimatedVectorY()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetEstimatedVectorY (
```

```
const arma::vec & Y_est ) [inline]
```

Установка оценки вектора состояния  $Y$ .

Требуется, например, при начальной установке фильтра

Аргументы

$Y\_est$	- вектор состояния, размерность [SizeY]
----------	---

#### 8.5.3.26 SetMeasuredVectorY()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetMeasuredVectorY (
```

```
const arma::vec & Y_msd ) [inline]
```

Установка измеренного вектора измерений  $Y$ .

Требуется после получения новых измерений

Аргументы

$Y\_msd$	- вектор измерений, размерность [SizeY]
----------	---

#### 8.5.3.27 SetObservationCovarianceMatrixR()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetObservationCovarianceMatrixR (
```

```
const arma::mat & R ) [inline]
```

Установка ковариационной матрицы  $R$  шумов измерений  $Y$ .

Аргументы

$R$	- матрица $R$ , размерность [SizeY, SizeY]
-----	--

#### 8.5.3.28 SetObservationCovarianceMatrixRdiag()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetObservationCovarianceMatrixRdiag (
```

```
const arma::vec & Rdiag ) [inline]
```

Установка ковариационной матрицы  $R$  шумов измерений  $Y$ .

Аргументы

$Rdiag$	- вектор элементов главной диагонали матрицы $R$ (остальные элементы полагаются равными нулю), размерность [SizeY]
---------	--

#### 8.5.3.29 SetObservationMatrixH()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetObservationMatrixH (
```

```
const arma::mat & H ) [inline]
```

Установка матрицы перехода измерений  $H$ .

## Аргументы

Н	- матрица перехода измерений, размерность [SizeY * SizeX]
---	---

## 8.5.3.30 SetProcessCovarianceMatrixQ()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetProcessCovarianceMatrixQ (
    const arma::mat & Q ) [inline]
```

Установка ковариационной матрицы Q шумов состояния X.

## Аргументы

Q	- матрица Q, размерность [SizeX, SizeX]
---	---

## 8.5.3.31 SetProcessCovarianceMatrixQdiag()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetProcessCovarianceMatrixQdiag (
    const arma::vec & Qdiag ) [inline]
```

Установка диагонали ковариационной матрицы Q шумов состояния X.

## Аргументы

Qdiag	- вектор элементов главной диагонали матрицы Q (остальные элементы полагаются равными нулю), размерность [SizeX]
-------	--

## 8.5.3.32 SetStateTransitionJacobianLinearF()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetStateTransitionJacobianLinearF (
    std::function< arma::mat(double dt)> stateTransitionJacobianLinearF ) [inline]
```

Установка функции вычисления матрицы перехода состояния F в случае LKF (makeMatrixF)

## Аргументы

stateTransitionJacobianLinearF	- Функция вычисления матрицы перехода состояния F в случае LKF
--------------------------------	--

## 8.5.3.33 SetStateTransitionMatrixF()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanLKF< SizeX, SizeY >::SetStateTransitionMatrixF (
    const arma::mat & F ) [inline]
```

Установка матрицы перехода состояния F.

## Аргументы

F	- матрица перехода состояния, размерность [SizeX * SizeX]
---	---

## 8.5.4 Документация по друзьям класса и функциям, относящимся к классу

### 8.5.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    CKalmanLKF< SizeX, SizeY > & lhs,
    CKalmanLKF< SizeX, SizeY > & rhs ) [friend]
```

Метод свапа

Аргументы

lhs	- left hand side instance
rhs	- right hand side instance

## 8.5.5 Данные класса

### 8.5.5.1 checkBordersMeasurement\_

```
template<size_t SizeX, size_t SizeY>
std::function<arma::vec( const arma::vec &Y )> KalmanFilters::CKalmanLKF< SizeX, SizeY >::checkBorders←
Measurement_ [protected]
```

Проверка границ вектора измерения Y.

### 8.5.5.2 checkBordersStateAfterCorrection\_

```
template<size_t SizeX, size_t SizeY>
std::function<arma::vec( const arma::vec &X )> KalmanFilters::CKalmanLKF< SizeX, SizeY >::checkBordersStateAfter←
Correction_ [protected]
```

Проверка границ вектора состояния X после фильтрации

### 8.5.5.3 checkBordersStateAfterPrediction\_

```
template<size_t SizeX, size_t SizeY>
std::function<arma::vec( const arma::vec &X )> KalmanFilters::CKalmanLKF< SizeX, SizeY >::checkBordersStateAfter←
Prediction_ [protected]
```

Проверка границ вектора состояния X после прогноза

### 8.5.5.4 checkDeltaMeasurement\_

```
template<size_t SizeX, size_t SizeY>
std::function<arma::vec( const arma::vec &DeltaY )> KalmanFilters::CKalmanLKF< SizeX, SizeY >::checkDelta←
Measurement_ [protected]
```

Проверка разности векторов измерения Y.

### 8.5.5.5 checkDeltaState\_

```
template<size_t SizeX, size_t SizeY>
std::function<arma::vec( const arma::vec &DeltaX )> KalmanFilters::CKalmanLKF< SizeX, SizeY >::checkDeltaState_
[protected]
```

Проверка разности векторов состояния X.

## 8.5.5.6 DeltaY\_

```
template<size_t SizeX, size_t SizeY>
arma::vec KalmanFilters::CKalmanLKF< SizeX, SizeY >::DeltaY_ [protected]
Вектор невязки измерений, размерность [SizeY].
```

## 8.5.5.7 deltaY\_isSet

```
template<size_t SizeX, size_t SizeY>
bool KalmanFilters::CKalmanLKF< SizeX, SizeY >::deltaY_isSet [protected]
Признак установки вектора невязки DeltaY (нужно в методе Correction)
```

## 8.5.5.8 F\_

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanLKF< SizeX, SizeY >::F_ [protected]
Матрица эволюции системы (перехода состояния) (state-transition model), размерность [SizeX * SizeX].
```

## 8.5.5.9 H\_

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanLKF< SizeX, SizeY >::H_ [protected]
Матрица измерений (перехода измерений) (observation model), размерность [SizeY * SizeX].
```

## 8.5.5.10 I\_

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanLKF< SizeX, SizeY >::I_ [protected]
Единичная матрица, размерность [SizeX * SizeX].
```

## 8.5.5.11 K\_

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanLKF< SizeX, SizeY >::K_ [protected]
Коэффициент усиления фильтра Калмана (Kalman gain), размерность [SizeX * SizeY].
```

## 8.5.5.12 P\_

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanLKF< SizeX, SizeY >::P_ [protected]
Ковариационная матрица вектора состояния X (estimate covariance matrix), размерность [SizeX * SizeX].
```

## 8.5.5.13 prediction\_isDone

```
template<size_t SizeX, size_t SizeY>
bool KalmanFilters::CKalmanLKF< SizeX, SizeY >::prediction_isDone [protected]
Признак выполненного прогноза (без этого нельзя делать фильтрацию)
```

## 8.5.5.14 Q\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::mat KalmanFilters::CKalmanLKF< SizeX, SizeY >::Q_ [protected]
```

Ковариационная матрица (обычно диагональная) шумов вектора состояния X НА 1 СЕКУНДЕ (covariance of the process noise), размерность [SizeX \* SizeX].

## 8.5.5.15 R\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::mat KalmanFilters::CKalmanLKF< SizeX, SizeY >::R_ [protected]
```

Ковариационная матрица (обычно диагональная) шумов вектора измерений Y (covariance of the observation noise), размерность [SizeY \* SizeY].

## 8.5.5.16 S\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::mat KalmanFilters::CKalmanLKF< SizeX, SizeY >::S_ [protected]
```

Ковариационная матрица вектора невязки (innovation covariance), размерность [SizeY \* SizeY].

## 8.5.5.17 SizeX\_

```
template<size_t SizeX, size_t SizeY>
```

```
size_t KalmanFilters::CKalmanLKF< SizeX, SizeY >::SizeX_ [protected]
```

Размерность вектора состояния X (state)

## 8.5.5.18 SizeY\_

```
template<size_t SizeX, size_t SizeY>
```

```
size_t KalmanFilters::CKalmanLKF< SizeX, SizeY >::SizeY_ [protected]
```

Размерность вектора измерений Y (measurement)

## 8.5.5.19 stateTransitionJacobianLinearF\_

```
template<size_t SizeX, size_t SizeY>
```

```
std::function<arma::mat( double dt )> KalmanFilters::CKalmanLKF< SizeX, SizeY >::stateTransitionJacobianLinearF_ [private]
```

Функция вычисления матрицы перехода состояния F (makeMatrixF)

Внимание

Используется только в линейном фильтре Калмана

Аргументы

X	- вектор состояния с прошлого момента времени
---	---

## 8.5.5.20 X\_est\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::vec KalmanFilters::CKalmanLKF< SizeX, SizeY >::X_est_ [protected]
```

Скорректированный (estimated) вектор состояния X, размерность [SizeX].



## 8.5.5.21 X\_pred\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::vec KalmanFilters::CKalmanLKF< SizeX, SizeY >::X_pred_ [protected]
```

Экстраполированный (predicted) вектор состояния X, размерность [SizeX].

## 8.5.5.22 Y\_est\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::vec KalmanFilters::CKalmanLKF< SizeX, SizeY >::Y_est_ [protected]
```

Скорректированный (estimated) вектор измерений Y, размерность [SizeY].

## 8.5.5.23 Y\_msd\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::vec KalmanFilters::CKalmanLKF< SizeX, SizeY >::Y_msd_ [protected]
```

Измеренный (measured) вектор Y (отметка), размерность [SizeY].

## 8.5.5.24 Y\_msd\_isSet

```
template<size_t SizeX, size_t SizeY>
```

```
bool KalmanFilters::CKalmanLKF< SizeX, SizeY >::Y_msd_isSet [protected]
```

Признак установки вектора измерений Y\_msd;.

## 8.5.5.25 Y\_pred\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::vec KalmanFilters::CKalmanLKF< SizeX, SizeY >::Y_pred_ [protected]
```

Экстраполированный (predicted) вектор измерений Y, размерность [SizeY].

Объявления и описания членов класса находятся в файле:

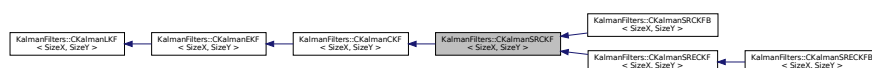
- kalman\_filter\_linear.h

## 8.6 Шаблон класса KalmanFilters::CKalmanSRCKF&lt; SizeX, SizeY &gt;

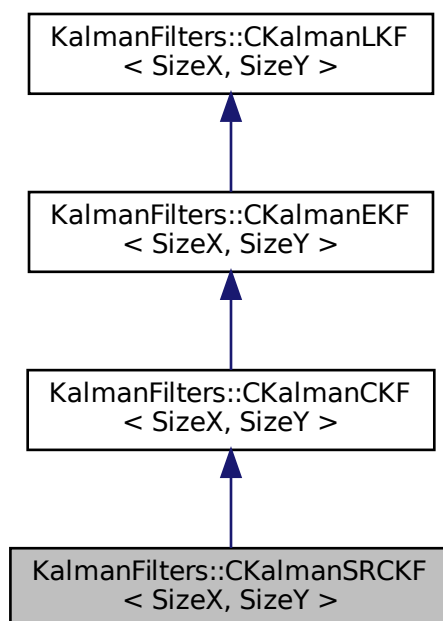
Шаблонный класс квадратно-корневого кубатурного фильтра Калмана, КК-КФК (Square Root Cubature Kalman Filter, SR-CKF)

```
#include <kalman_filter_cubature_square_root.h>
```

Граф наследования: KalmanFilters::CKalmanSRCKF< SizeX, SizeY >:



Граф связей класса `KalmanFilters::CKalmanSRCKF< SizeX, SizeY >`:



#### Открытые члены

- [CKalmanSRCKF](#) ()  
Конструктор по умолчанию
- [CKalmanSRCKF](#) (const [CKalmanSRCKF](#) &other)  
Конструктор копирования
- [CKalmanSRCKF](#) & [operator=](#) (const [CKalmanSRCKF](#) &other)  
Перегрузка оператора присвоения
- [CKalmanSRCKF](#) ([CKalmanSRCKF](#) &&other) noexcept  
Конструктор перемещения
- [CKalmanSRCKF](#) & [operator=](#) ([CKalmanSRCKF](#) &&other) noexcept  
Перегрузка оператора перемещения
- virtual [~CKalmanSRCKF](#) ()=default  
Деструктор
- void [SetStateTransitionJacobianF](#) (std::function< arma::mat(const arma::vec &X, double dt)> stateTransitionJacobianF)  
Установка функции вычисления матрицы перехода состояния F (makeMatrixF)
- void [SetObservationJacobianH](#) (std::function< arma::mat(const arma::vec &X)> observationJacobianH)  
Установка функции вычисления матрицы перехода измерений H (makeMatrixH)
- virtual void [Prediction](#) (double dt)  
Прогноз
- virtual void [Correction](#) (const arma::vec &Y\_msd)  
Коррекция

## Защищенные члены

- void [CorrectionSRCKF](#) (const arma::vec &Y\_msd)  
Коррекция SRCKF.

## Друзья

- void [swap](#) ([CKalmanSRCKF](#)< SizeX, SizeY > &lhs, [CKalmanSRCKF](#)< SizeX, SizeY > &rhs)  
поехсепт  
Метод свапа

## Дополнительные унаследованные члены

## 8.6.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanSRCKF< SizeX, SizeY >
```

Шаблонный класс квадратно-корневого кубатурного фильтра Калмана, КК-КФК (Square Root Cubature Kalman Filter, SR-CKF)

Источники:

[1] Cubature Kalman Filters, Ienkaran Arasaratnam and Simon Haykin, Life Fellow, IEEE

## Внимание

Фильтр построен по классическому его варианту НИЖНЕтреугольного разложения Холецкого!

## Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

## 8.6.2 Конструктор(ы)

## 8.6.2.1 CKalmanSRCKF() [1/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSRCKF< SizeX, SizeY >::CKalmanSRCKF ( ) [inline]
Конструктор по умолчанию
```

## 8.6.2.2 CKalmanSRCKF() [2/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSRCKF< SizeX, SizeY >::CKalmanSRCKF (
    const CKalmanSRCKF< SizeX, SizeY > & other ) [inline]
```

Конструктор копирования

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.6.2.3 CKalmanSRCKF() [3/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSRCKF< SizeX, SizeY >::CKalmanSRCKF (
    CKalmanSRCKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Конструктор перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.6.2.4 ~CKalmanSRCKF()

```
template<size_t SizeX, size_t SizeY>
virtual KalmanFilters::CKalmanSRCKF< SizeX, SizeY >::~~CKalmanSRCKF ( ) [virtual], [default]
```

Деструктор

## 8.6.3 Методы

## 8.6.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanSRCKF< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd ) [inline], [virtual]
```

Коррекция

Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanCKF< SizeX, SizeY >](#).

Переопределяется в [KalmanFilters::CKalmanSRCKFB< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRECKF< SizeX, SizeY >](#) и [KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >](#).

## 8.6.3.2 CorrectionSRCKF()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanSRCKF< SizeX, SizeY >::CorrectionSRCKF (
    const arma::vec & Y_msd ) [inline], [protected]
```

Коррекция SRCKF.

Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

## 8.6.3.3 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
CKalmanSRCKF & KalmanFilters::CKalmanSRCKF< SizeX, SizeY >::operator= (
    CKalmanSRCKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Перегрузка оператора перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

#### 8.6.3.4 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanSRCKF & KalmanFilters::CKalmanSRCKF< SizeX, SizeY >::operator= (
    const CKalmanSRCKF< SizeX, SizeY > & other ) [inline]
```

Перегрузка оператора присвоения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

#### 8.6.3.5 Prediction()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual void KalmanFilters::CKalmanSRCKF< SizeX, SizeY >::Prediction (
    double dt ) [inline], [virtual]
```

Прогноз

Аргументы

dt	- Время прогноза, [с]
----	-----------------------

Переопределяет метод предка [KalmanFilters::CKalmanCKF< SizeX, SizeY >](#).  
Переопределяется в [KalmanFilters::CKalmanSRECKF< SizeX, SizeY >](#).

#### 8.6.3.6 SetObservationJacobianH()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanSRCKF< SizeX, SizeY >::SetObservationJacobianH (
    std::function< arma::mat(const arma::vec &X)> observationJacobianH ) [inline]
```

Установка функции вычисления матрицы перехода измерений H (makeMatrixH)

См. также

[observationJacobianH\\_](#)

#### 8.6.3.7 SetStateTransitionJacobianF()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanSRCKF< SizeX, SizeY >::SetStateTransitionJacobianF (
    std::function< arma::mat(const arma::vec &X, double dt)> stateTransitionJacobianF ) [inline]
```

Установка функции вычисления матрицы перехода состояния F (makeMatrixF)

См. также

[stateTransitionJacobianF\\_](#)

## 8.6.4 Документация по друзьям класса и функциям, относящимся к классу

### 8.6.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    CKalmanSRCKF< SizeX, SizeY > & lhs,
    CKalmanSRCKF< SizeX, SizeY > & rhs ) [friend]
```

Метод свапа

Аргументы

lhs	- left hand side instance
rhs	- right hand side instance

Объявления и описания членов класса находятся в файле:

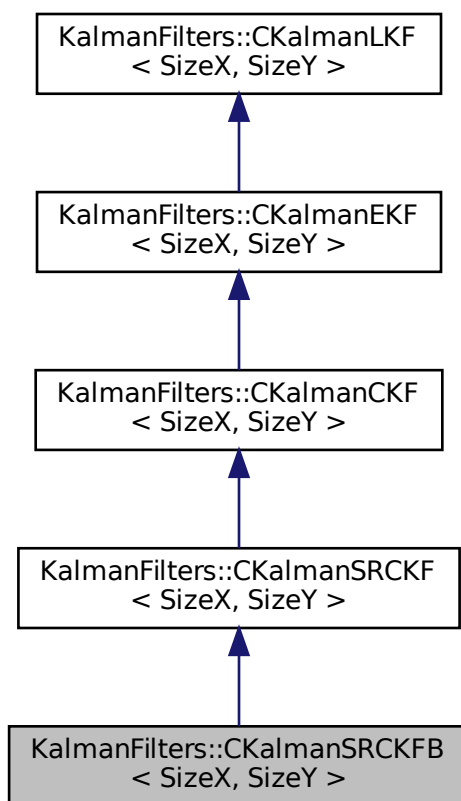
- kalman\_filter\_cubature\_square\_root.h

## 8.7 Шаблон класса KalmanFilters::CKalmanSRCKFB< SizeX, SizeY >

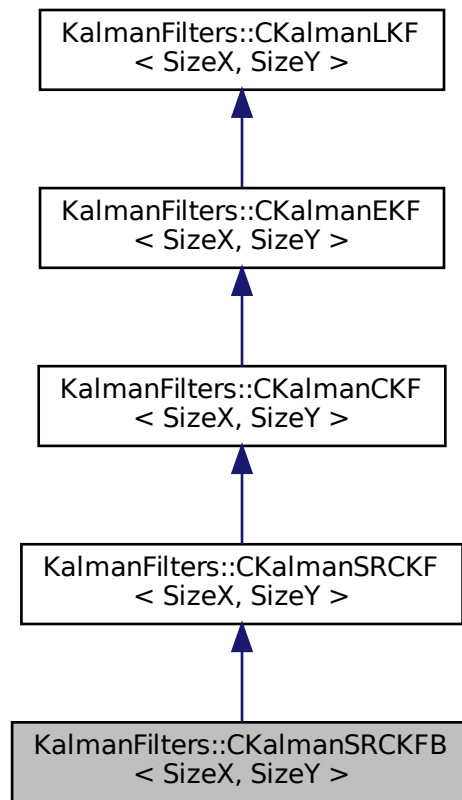
Шаблонный класс квадратно-корневого кубатурного фильтра Калмана (блочная фильтрация), КК-КФКБ (Square Root Cubature Kalman Filter Block, SR-CKFB)

#include <kalman\_filter\_cubature\_square\_root.h>

Граф наследования: KalmanFilters::CKalmanSRCKFB< SizeX, SizeY >:



Граф связей класса `KalmanFilters::CKalmanSRCKFB< SizeX, SizeY >`:



#### Открытые члены

- [CKalmanSRCKFB](#) ()  
Конструктор по умолчанию
- virtual void [Correction](#) (const arma::vec &Y\_msd)  
Коррекция

#### Дополнительные унаследованные члены

##### 8.7.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanSRCKFB< SizeX, SizeY >
```

Шаблонный класс квадратно-корневого кубатурного фильтра Калмана (блочная фильтрация), КК-КФКБ (Square Root Cubature Kalman Filter Block, SR-CKFB)

См. [CKalmanSRCKF](#) и The J-Orthogonal Square-Root Euler-Maruyama-Based Unscented Kalman Filter for Nonlinear Stochastic Systems, Gennady Yu. Kulikov, Maria V. Kulikova, CEMAT, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 LISBOA, Portugal (emails: gennady.kulikov[at]tecnico.ulisboa.pt, maria.kulikova[at]ist.utl.pt)



Внимание

Фильтр построен по классическому его варианту НИЖНЕтреугольного разложения Холецкого!

Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

## 8.7.2 Конструктор(ы)

### 8.7.2.1 CKalmanSRCKFB()

```
template<size_t SizeX, size_t SizeY>
```

```
KalmanFilters::CKalmanSRCKFB< SizeX, SizeY >::CKalmanSRCKFB ( ) [inline]
```

Конструктор по умолчанию

## 8.7.3 Методы

### 8.7.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual void KalmanFilters::CKalmanSRCKFB< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd ) [inline], [virtual]
```

Коррекция

Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanSRCKF< SizeX, SizeY >](#).

Объявления и описания членов класса находятся в файле:

- kalman\_filter\_cubature\_square\_root.h

## 8.8 Шаблон класса KalmanFilters::CKalmanSRECKF< SizeX, SizeY >

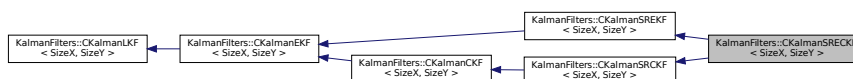
Шаблонный класс квадратно-корневого расширенного кубатурного фильтра Калмана, КК-РКФК (Square Root Extended Cubature Kalman Filter, SR-ECKF)

```
#include <kalman_filter_extended_cubature_square_root.h>
```

Граф наследования: KalmanFilters::CKalmanSRECKF< SizeX, SizeY >:



Граф связей класса KalmanFilters::CKalmanSRECKF< SizeX, SizeY >:



## Открытые члены

- [CKalmanSRECKF](#) ()  
Конструктор по умолчанию
- [CKalmanSRECKF](#) (const [CKalmanSRECKF](#) &other)  
Конструктор копирования
- [CKalmanSRECKF](#) & [operator=](#) (const [CKalmanSRECKF](#) &other)  
Перегрузка оператора присвоения
- [CKalmanSRECKF](#) ([CKalmanSRECKF](#) &&other) noexcept  
Конструктор перемещения
- [CKalmanSRECKF](#) & [operator=](#) ([CKalmanSRECKF](#) &&other) noexcept  
Перегрузка оператора перемещения
- virtual [~CKalmanSRECKF](#) ()=default  
Деструктор
- virtual void [Prediction](#) (double dt)  
Прогноз квадратно-корневого расширенного кубатурного фильтра Калмана (КК-РКФК, SR-↔ECKF)
- virtual void [Correction](#) (const arma::vec &Y\_msd)  
Коррекция квадратно-корневого расширенного кубатурного фильтра Калмана (КК-РКФК, SR-↔ECKF)

## Друзья

- void [swap](#) ([CKalmanSRECKF](#)< SizeX, SizeY > &lhs, [CKalmanSRECKF](#)< SizeX, SizeY > &rhs)  
noexcept  
Метод свапа

## Дополнительные унаследованные члены

### 8.8.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanSRECKF< SizeX, SizeY >
```

Шаблонный класс квадратно-корневого расширенного кубатурного фильтра Калмана, КК-РКФК (Square Root Extended Cubature Kalman Filter, SR-ECKF)

Источники:

[1] Cubature Kalman Filters, Ienkaran Arasaratnam and Simon Haykin, Life Fellow, IEEE

## Внимание

Фильтр построен по классическому его варианту НИЖНЕтреугольного разложения Холецкого!

## Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

## 8.8.2 Конструктор(ы)

### 8.8.2.1 CKalmanSRECKF() [1/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSRECKF< SizeX, SizeY >::CKalmanSRECKF ( ) [inline]
Конструктор по умолчанию
```

### 8.8.2.2 CKalmanSRECKF() [2/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSRECKF< SizeX, SizeY >::CKalmanSRECKF (
    const CKalmanSRECKF< SizeX, SizeY > & other ) [inline]
Конструктор копирования
```

Аргументы

other	- экземпляр, с которого делается копия
-------	--

### 8.8.2.3 CKalmanSRECKF() [3/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSRECKF< SizeX, SizeY >::CKalmanSRECKF (
    CKalmanSRECKF< SizeX, SizeY > && other ) [inline], [noexcept]
Конструктор перемещения
```

Аргументы

other	- экземпляр, с которого делается копия
-------	--

### 8.8.2.4 ~CKalmanSRECKF()

```
template<size_t SizeX, size_t SizeY>
virtual KalmanFilters::CKalmanSRECKF< SizeX, SizeY >::~~CKalmanSRECKF ( ) [virtual], [default]
Деструктор
```

## 8.8.3 Методы

### 8.8.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanSRECKF< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd ) [inline], [virtual]
```

Коррекция квадратно-корневого расширенного кубатурного фильтра Калмана (КК-РКФК, SR- $\leftrightarrow$  ECKF)

Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanSRCKF< SizeX, SizeY >](#).  
 Переопределяется в [KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >](#).

#### 8.8.3.2 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
CKalmanSRECKF & KalmanFilters::CKalmanSRECKF< SizeX, SizeY >::operator= (
    CKalmanSRECKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Перегрузка оператора перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

#### 8.8.3.3 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
CKalmanSRECKF & KalmanFilters::CKalmanSRECKF< SizeX, SizeY >::operator= (
    const CKalmanSRECKF< SizeX, SizeY > & other ) [inline]
```

Перегрузка оператора присвоения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

#### 8.8.3.4 Prediction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanSRECKF< SizeX, SizeY >::Prediction (
    double dt ) [inline], [virtual]
```

Прогноз квадратно-корневого расширенного кубатурного фильтра Калмана (КК-РКФК, SR-ECKF)

Аргументы

dt	- Время прогноза, [с]
----	-----------------------

Переопределяет метод предка [KalmanFilters::CKalmanSRCKF< SizeX, SizeY >](#).

### 8.8.4 Документация по друзьям класса и функциям, относящимся к классу

## 8.8.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    CKalmanSRECKFB< SizeX, SizeY > & lhs,
    CKalmanSRECKFB< SizeX, SizeY > & rhs ) [friend]
```

Метод свапа

Аргументы

lhs	- left hand side instance
rhs	- right hand side instance

Объявления и описания членов класса находятся в файле:

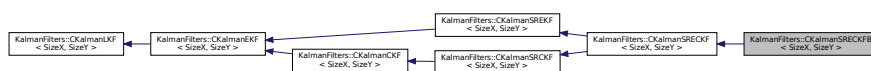
- kalman\_filter\_extended\_cubature\_square\_root.h

## 8.9 Шаблон класса KalmanFilters::CKalmanSRECKFB&lt; SizeX, SizeY &gt;

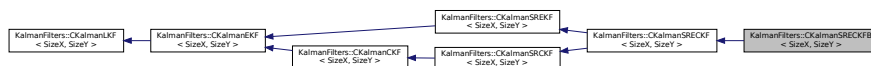
Шаблонный класс квадратно-корневого расширенного кубатурного фильтра Калмана (блочная фильтрация), КК-РКФКБ (Square Root Extended Cubature Kalman Filter Block, SR-ECKFB)

```
#include <kalman_filter_extended_cubature_square_root.h>
```

Граф наследования:KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >:



Граф связей класса KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >:



Открытые члены

- [CKalmanSRECKFB](#) ()  
Конструктор по умолчанию
- [CKalmanSRECKFB](#) (const [CKalmanSRECKFB](#) &other)  
Конструктор копирования
- [CKalmanSRECKFB](#) & operator= (const [CKalmanSRECKFB](#) &other)  
Перегрузка оператора присвоения
- [CKalmanSRECKFB](#) ([CKalmanSRECKFB](#) &&other) noexcept  
Конструктор перемещения
- [CKalmanSRECKFB](#) & operator= ([CKalmanSRECKFB](#) &&other) noexcept  
Перегрузка оператора перемещения
- virtual [~CKalmanSRECKFB](#) ()=default  
Деструктор
- virtual void [Correction](#) (const arma::vec &Y\_msd)  
Коррекция

## Друзья

- void [swap](#) ([CKalmanSRECKFB](#)< SizeX, SizeY > &lhs, [CKalmanSRECKFB](#)< SizeX, SizeY > &rhs) noexcept  
Метод свапа

## Дополнительные унаследованные члены

### 8.9.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >
```

Шаблонный класс квадратно-корневого расширенного кубатурного фильтра Калмана (блочная фильтрация), КК-РКФКБ (Square Root Extended Cubature Kalman Filter Block, SR-ECKFB)  
См. [CKalmanSRECKF](#) и The J-Orthogonal Square-Root Euler-Maruyama-Based Unscented Kalman Filter for Nonlinear Stochastic Systems, Gennady Yu. Kulikov, Maria V. Kulikova, CEMAT, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 LISBOA, Portugal (emails↔: gennady.kulikov[at]tecnico.ulisboa.pt, maria.kulikova[at]ist.utl.pt)

#### Внимание

Фильтр построен по классическому его варианту НИЖНЕтреугольного разложения Холецкого!

#### Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

### 8.9.2 Конструктор(ы)

#### 8.9.2.1 CKalmanSRECKFB() [1/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >::CKalmanSRECKFB ( ) [inline]
Конструктор по умолчанию
```

#### 8.9.2.2 CKalmanSRECKFB() [2/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >::CKalmanSRECKFB (
    const CKalmanSRECKFB< SizeX, SizeY > & other ) [inline]
Конструктор копирования
```

#### Аргументы

other	- экземпляр, с которого делается копия
-------	--

#### 8.9.2.3 CKalmanSRECKFB() [3/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >::CKalmanSRECKFB (
    CKalmanSRECKFB< SizeX, SizeY > && other ) [inline], [noexcept]
```

## Конструктор перемещения

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.9.2.4 ~CKalmanSRECKFB()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >::~CKalmanSRECKFB ( ) [virtual], [default]
```

Деструктор

## 8.9.3 Методы

## 8.9.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual void KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd ) [inline], [virtual]
```

Коррекция

## Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanSRECKF< SizeX, SizeY >](#).

## 8.9.3.2 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanSRECKFB & KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >::operator= (
    CKalmanSRECKFB< SizeX, SizeY > && other ) [inline], [noexcept]
```

Перегрузка оператора перемещения

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

## 8.9.3.3 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanSRECKFB & KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >::operator= (
    const CKalmanSRECKFB< SizeX, SizeY > & other ) [inline]
```

Перегрузка оператора присвоения

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

## 8.9.4 Документация по друзьям класса и функциям, относящимся к классу

### 8.9.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    CKalmanSRECKFB< SizeX, SizeY > & lhs,
    CKalmanSRECKFB< SizeX, SizeY > & rhs ) [friend]
```

Метод свапа

Аргументы

lhs	- left hand side instance
rhs	- right hand side instance

Объявления и описания членов класса находятся в файле:

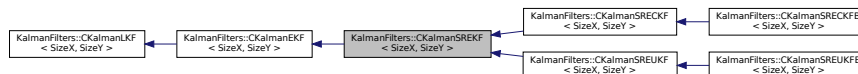
- kalman\_filter\_extended\_cubature\_square\_root.h

## 8.10 Шаблон класса KalmanFilters::CKalmanSREKF< SizeX, SizeY >

Класс квадратно-корневого расширенного фильтра Калмана, КК-РФК (Square Root Extended Kalman Filter, EKF)

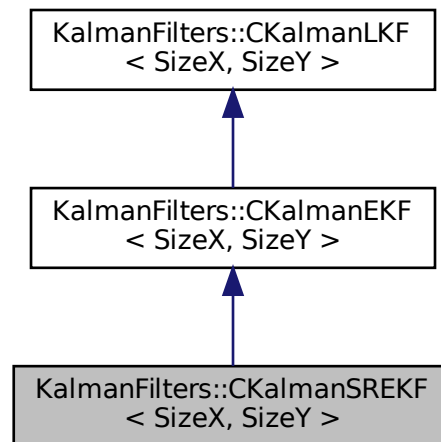
#include <kalman\_filter\_extended\_square\_root.h>

Граф наследования: KalmanFilters::CKalmanSREKF< SizeX, SizeY >:





Граф связей класса KalmanFilters::CKalmanSREKF < SizeX, SizeY >:



#### Открытые члены

- `CKalmanSREKF` ()  
Конструктор по умолчанию
- `CKalmanSREKF` (const `CKalmanSREKF` &other)  
Конструктор копирования
- `CKalmanSREKF` & `operator=` (const `CKalmanSREKF` &other)  
Перегрузка оператора присвоения
- `CKalmanSREKF` (`CKalmanSREKF` &&other) noexcept  
Конструктор перемещения
- `CKalmanSREKF` & `operator=` (`CKalmanSREKF` &&other) noexcept  
Перегрузка оператора перемещения
- virtual `~CKalmanSREKF` ()=default  
Деструктор
- virtual void `Prediction` (double dt)  
Прогноз
- virtual void `Correction` (const arma::vec &Y\_msd)  
Коррекция

#### Защищенные члены

- void `PredictionSREKF` (double dt)  
Прогноз SREKF.

#### Друзья

- void `swap` (`CKalmanSREKF` &lhs, `CKalmanSREKF` &rhs) noexcept  
Метод свапа

## Дополнительные унаследованные члены

### 8.10.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanSREKF< SizeX, SizeY >
```

Класс квадратно-корневого расширенного фильтра Калмана, КК-РФК (Square Root Extended Kalman Filter, EKF)

Источник: NASA Technical report R-135, Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle, Gerald L. Smith, Stanley F. Schmidt and Leonard A. McGee, 1962

Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

### 8.10.2 Конструктор(ы)

#### 8.10.2.1 CKalmanSREKF() [1/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSREKF< SizeX, SizeY >::CKalmanSREKF ( ) [inline]
Конструктор по умолчанию
```

#### 8.10.2.2 CKalmanSREKF() [2/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSREKF< SizeX, SizeY >::CKalmanSREKF (
    const CKalmanSREKF< SizeX, SizeY > & other ) [inline]
Конструктор копирования
```

Аргументы

other	- экземпляр, с которого делается копия
-------	--

#### 8.10.2.3 CKalmanSREKF() [3/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSREKF< SizeX, SizeY >::CKalmanSREKF (
    CKalmanSREKF< SizeX, SizeY > && other ) [inline], [noexcept]
Конструктор перемещения
```

Аргументы

other	- экземпляр, с которого делается копия
-------	--

#### 8.10.2.4 ~CKalmanSREKF()

```
template<size_t SizeX, size_t SizeY>
```

virtual [KalmanFilters::CKalmanSREKF< SizeX, SizeY >::~CKalmanSREKF](#) ( ) [virtual], [default]  
 Деструктор

### 8.10.3 Методы

#### 8.10.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanSREKF< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd ) [inline], [virtual]
```

Коррекция

Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanEKF< SizeX, SizeY >](#).

Переопределяется в [KalmanFilters::CKalmanSRECKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >](#), [KalmanFilters::CKalmanSREUKF< SizeX, SizeY >](#) и [KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >](#).

#### 8.10.3.2 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
CKalmanSREKF & KalmanFilters::CKalmanSREKF< SizeX, SizeY >::operator= (
    CKalmanSREKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Перегрузка оператора перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

#### 8.10.3.3 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
CKalmanSREKF & KalmanFilters::CKalmanSREKF< SizeX, SizeY >::operator= (
    const CKalmanSREKF< SizeX, SizeY > & other ) [inline]
```

Перегрузка оператора присвоения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

`*this`

#### 8.10.3.4 Prediction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanSREKF< SizeX, SizeY >::Prediction (
    double dt ) [inline], [virtual]
```

Прогноз

Аргументы

dt	- Время прогноза, [с]
----	-----------------------

Переопределяет метод предка [KalmanFilters::CKalmanEKF](#)< SizeX, SizeY >.

Переопределяется в [KalmanFilters::CKalmanSRECKF](#)< SizeX, SizeY > и [KalmanFilters::CKalmanSREUKF](#)< SizeX, SizeY >.

#### 8.10.3.5 PredictionSREKF()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanSREKF< SizeX, SizeY >::PredictionSREKF (
    double dt ) [inline], [protected]
```

Прогноз SREKF.

Аргументы

dt	- Время прогноза, [с]
----	-----------------------

### 8.10.4 Документация по друзьям класса и функциям, относящимся к классу

#### 8.10.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    CKalmanSREKF< SizeX, SizeY > & lhs,
    CKalmanSREKF< SizeX, SizeY > & rhs ) [friend]
```

Метод свапа

Аргументы

lhs	- left hand side instance
rhs	- right hand side instance

Объявления и описания членов класса находятся в файле:

- `kalman_filter_extended_square_root.h`

## 8.11 Шаблон класса [KalmanFilters::CKalmanSREUKF](#)< SizeX, SizeY >

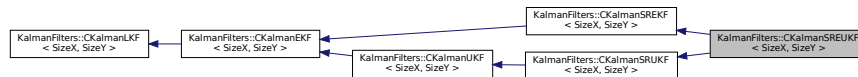
Шаблонный класс квадратно-корневого расширенного сигма-точечного (ансцентного) фильтра Калмана, КК-РСТФК (Square Root Extended Unscented Kalman Filter, SR-EUKF)

```
#include <kalman_filter_extended_unscented_square_root.h>
```

Граф наследования: KalmanFilters::CKalmanSREUKF< SizeX, SizeY >:



Граф связей класса KalmanFilters::CKalmanSREUKF< SizeX, SizeY >:



### Открытые члены

- [CKalmanSREUKF](#) ()  
Конструктор по умолчанию
- [CKalmanSREUKF](#) (const [CKalmanSREUKF](#) &other)  
Конструктор копирования
- [CKalmanSREUKF](#) & [operator=](#) (const [CKalmanSREUKF](#) &other)  
Перегрузка оператора присвоения
- [CKalmanSREUKF](#) ([CKalmanSREUKF](#) &&other) noexcept  
Конструктор перемещения
- [CKalmanSREUKF](#) & [operator=](#) ([CKalmanSREUKF](#) &&other) noexcept  
Перегрузка оператора перемещения
- virtual [~CKalmanSREUKF](#) ()=default  
Деструктор
- virtual void [Prediction](#) (double dt)  
Прогноз квадратно-корневого расширенного сигма-точечного фильтра Калмана (КК-РСТФК, SR-EUKF)
- virtual void [Correction](#) (const arma::vec &Y\_msd)  
Коррекция квадратно-корневого расширенного сигма-точечного фильтра Калмана (КК-РСТФК, SR-EUKF)

### Защищенные члены

- void [createSignMatrices](#) ()  
Создание матриц знаков

### Защищенные данные

- arma::mat [J\\_](#)  
Матрица знаков при Pху.
- arma::mat [Jcorrect\\_](#)  
Матрица знаков при коррекции

### Друзья

- void [swap](#) ([CKalmanSREUKF](#)< SizeX, SizeY > &lhs, [CKalmanSREUKF](#)< SizeX, SizeY > &rhs)  
noexcept  
Метод свапа

### 8.11.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanSREUKF< SizeX, SizeY >
```

Шаблонный класс квадратно-корневого расширенного сигма-точечного (ансцентного) фильтра Калмана, КК-РСТФК (Square Root Extended Unscented Kalman Filter, SR-EUKF)

Источники:

- [1] A New Extension of the Kalman Filter to Nonlinear Systems, Simon J. Julier, Jeffrey K. Uhlmann, The Robotics Research Group, Department of Engineering Science, The University of Oxford, 1997
- [2] The Unscented Kalman Filter for Nonlinear Estimation, Eric A. Wan and Rudolph van der Merwe Oregon Graduate Institute of Science & Technology 20000 NW Walker Rd, Beaverton, Oregon 97006, 2000 ericwan[at]ece.ogi.edu, rvdmerwe[at]ece.ogi.edu
- [3] Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models, Rudolph van der Merwe & Eric Wan, OGI School of Science & Engineering Oregon Health & Science University Beaverton, Oregon, 97006, USA, 2003 {rvdmerwe,ericwan}[at]ece.ogi.edu
- [4] THE SQUARE-ROOT UNSCENTED KALMAN FILTER FOR STATE AND PARAMETER-ESTIMATION, Rudolph van der Merwe and Eric A. Wan, Oregon Graduate Institute of Science and Technology 20000 NW Walker Road, Beaverton, Oregon 97006, USA rvdmerwe,ericwan[at]ece.ogi.edu

Внимание

Внимание! В реализации UKF вес нулевой сигма-точки  $W_{cov}$  не может быть отрицательным! ( $W_{mean}$  - может)

Фильтр построен по классическому его варианту НИЖНЕтреугольного разложения Холецкого!

Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

### 8.11.2 Конструктор(ы)

#### 8.11.2.1 CKalmanSREUKF() [1/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSREUKF< SizeX, SizeY >::CKalmanSREUKF ( ) [inline]
Конструктор по умолчанию
```

#### 8.11.2.2 CKalmanSREUKF() [2/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSREUKF< SizeX, SizeY >::CKalmanSREUKF (
    const CKalmanSREUKF< SizeX, SizeY > & other ) [inline]
Конструктор копирования
```

Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.11.2.3 CKalmanSREUKF() [3/3]

```
template<size_t SizeX, size_t SizeY>
```

```
KalmanFilters::CKalmanSREUKF< SizeX, SizeY >::CKalmanSREUKF (
    CKalmanSREUKF< SizeX, SizeY > && other )    [inline], [noexcept]
```

Конструктор перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.11.2.4 ~CKalmanSREUKF()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual KalmanFilters::CKalmanSREUKF< SizeX, SizeY >::~~CKalmanSREUKF ( )    [virtual], [default]
```

Деструктор

## 8.11.3 Методы

## 8.11.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual void KalmanFilters::CKalmanSREUKF< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd )    [inline], [virtual]
```

Коррекция квадратно-корневого расширенного сигма-точечного фильтра Калмана (КК-РСТФК, SR-EUKF)

Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanSREKF< SizeX, SizeY >](#).

Переопределяется в [KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >](#).

## 8.11.3.2 createSignMatrices()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanSREUKF< SizeX, SizeY >::createSignMatrices ( )    [inline], [protected]
```

Создание матриц знаков

## 8.11.3.3 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanSREUKF & KalmanFilters::CKalmanSREUKF< SizeX, SizeY >::operator= (
    CKalmanSREUKF< SizeX, SizeY > && other )    [inline], [noexcept]
```

Перегрузка оператора перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

`*this`

#### 8.11.3.4 `operator=()` [2/2]

```
template<size_t SizeX, size_t SizeY>
CKalmanSREUKF & KalmanFilters::CKalmanSREUKF< SizeX, SizeY >::operator= (
    const CKalmanSREUKF< SizeX, SizeY > & other ) [inline]
```

Перегрузка оператора присвоения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

`*this`

#### 8.11.3.5 `Prediction()`

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanSREUKF< SizeX, SizeY >::Prediction (
    double dt ) [inline], [virtual]
```

Прогноз квадратно-корневого расширенного сигма-точечного фильтра Калмана (КК-РСТФК, SR-EUKF)

Аргументы

dt	- Время прогноза, [с]
----	-----------------------

Переопределяет метод предка [KalmanFilters::CKalmanSREKF< SizeX, SizeY >](#).

### 8.11.4 Документация по друзьям класса и функциям, относящимся к классу

#### 8.11.4.1 `swap`

```
template<size_t SizeX, size_t SizeY>
void swap (
    CKalmanSREUKF< SizeX, SizeY > & lhs,
    CKalmanSREUKF< SizeX, SizeY > & rhs ) [friend]
```

Метод свапа

Аргументы

lhs	- left hand side instance
rhs	- right hand side instance

#### 8.11.5 Данные класса



## 8.11.5.1 J\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::mat KalmanFilters::CKalmanSREUKF< SizeX, SizeY >::J_ [protected]
```

Матрица знаков при Pxy.

## 8.11.5.2 Jcorrect\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::mat KalmanFilters::CKalmanSREUKF< SizeX, SizeY >::Jcorrect_ [protected]
```

Матрица знаков при коррекции

Объявления и описания членов класса находятся в файле:

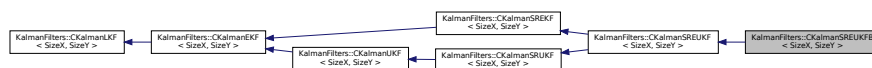
- kalman\_filter\_extended\_unscented\_square\_root.h

## 8.12 Шаблон класса KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >

Шаблонный класс квадратно-корневого расширенного сигма-точечного (ансцентного) фильтра Калмана (блочная фильтрация), КК-РСТФК (Square Root Extended Unscented Kalman Filter Block, SR-EUKFB)

```
#include <kalman_filter_extended_unscented_square_root.h>
```

Граф наследования: KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >:



Граф связей класса KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >:



## Открытые члены

- [CKalmanSREUKFB](#) ()  
Конструктор по умолчанию
- [CKalmanSREUKFB](#) (const [CKalmanSREUKFB](#) &other)  
Конструктор копирования
- [CKalmanSREUKFB](#) & operator= (const [CKalmanSREUKFB](#) &other)  
Перегрузка оператора присвоения
- [CKalmanSREUKFB](#) ([CKalmanSREUKFB](#) &&other) noexcept  
Конструктор перемещения
- [CKalmanSREUKFB](#) & operator= ([CKalmanSREUKFB](#) &&other) noexcept  
Перегрузка оператора перемещения
- virtual ~[CKalmanSREUKFB](#) ()=default  
Деструктор
- virtual void [Correction](#) (const arma::vec &Y\_msd)  
Коррекция

## Защищенные члены

- void [createSignMatricesBlock](#) ()  
Создание матриц знаков

## Защищенные данные

- arma::mat [JcorrectBlock](#) \_  
Матрица знаков для фильтра в блочном виде

## Друзья

- void [swap](#) (CKalmanSREUKFB< SizeX, SizeY > &lhs, CKalmanSREUKFB< SizeX, SizeY > &rhs) noexcept  
Метод свапа

### 8.12.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >
```

Шаблонный класс квадратно-корневого расширенного сигма-точечного (ансцентного) фильтра Калмана (блочная фильтрация), КК-РСТФК (Square Root Extended Unscented Kalman Filter Block, SR-EUKFB)

Источники:

- [1] A New Extension of the Kalman Filter to Nonlinear Systems, Simon J. Julier, Jeffrey K. Uhlmann, The Robotics Research Group, Department of Engineering Science, The University of Oxford, 1997
- [2] The Unscented Kalman Filter for Nonlinear Estimation, Eric A. Wan and Rudolph van der Merwe Oregon Graduate Institute of Science & Technology 20000 NW Walker Rd, Beaverton, Oregon 97006, 2000 ericwan[at]ece.ogi.edu, rvdmerwe[at]ece.ogi.edu
- [3] Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models, Rudolph van der Merwe & Eric Wan, OGI School of Science & Engineering Oregon Health & Science University Beaverton, Oregon, 97006, USA, 2003 {rvdmerwe,ericwan}[at]ece.ogi.edu
- [4] THE SQUARE-ROOT UNSCENTED KALMAN FILTER FOR STATE AND PARAMETER-ESTIMATION, Rudolph van der Merwe and Eric A. Wan, Oregon Graduate Institute of Science and Technology 20000 NW Walker Road, Beaverton, Oregon 97006, USA rvdmerwe,ericwan[at]ece.ogi.edu

Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

### 8.12.2 Конструктор(ы)

#### 8.12.2.1 CKalmanSREUKFB() [1/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >::CKalmanSREUKFB ( ) [inline]
Конструктор по умолчанию
```

#### 8.12.2.2 CKalmanSREUKFB() [2/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >::CKalmanSREUKFB (
    const CKalmanSREUKFB< SizeX, SizeY > & other ) [inline]
```

## Конструктор копирования

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.12.2.3 CKalmanSREUKFB() [3/3]

```
template<size_t SizeX, size_t SizeY>
```

```
KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >::CKalmanSREUKFB (
    CKalmanSREUKFB< SizeX, SizeY > && other ) [inline], [noexcept]
```

## Конструктор перемещения

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.12.2.4 ~CKalmanSREUKFB()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >::~~CKalmanSREUKFB ( ) [virtual], [default]
```

## Деструктор

## 8.12.3 Методы

## 8.12.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual void KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd ) [inline], [virtual]
```

## Коррекция

## Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanSREUKF< SizeX, SizeY >](#).

## 8.12.3.2 createSignMatricesBlock()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >::createSignMatricesBlock ( ) [inline], [protected]
```

## Создание матриц знаков

## 8.12.3.3 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanSREUKFB & KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >::operator= (
    CKalmanSREUKFB< SizeX, SizeY > && other ) [inline], [noexcept]
```

## Перегрузка оператора перемещения

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

## Возвращает

\*this

## 8.12.3.4 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanSREUKFB & KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >::operator= (
    const CKalmanSREUKFB< SizeX, SizeY > & other ) [inline]
```

## Перегрузка оператора присвоения

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

## Возвращает

\*this

## 8.12.4 Документация по друзьям класса и функциям, относящимся к классу

## 8.12.4.1 swap

```
template<size_t SizeX, size_t SizeY>
```

```
void swap (
    CKalmanSREUKFB< SizeX, SizeY > & lhs,
    CKalmanSREUKFB< SizeX, SizeY > & rhs ) [friend]
```

## Метод свапа

## Аргументы

lhs	- left hand side instance
rhs	- right hand side instance

## 8.12.5 Данные класса

## 8.12.5.1 JcorrectBlock\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::mat KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >::JcorrectBlock_ [protected]
```

Матрица знаков для фильтра в блочном виде

Объявления и описания членов класса находятся в файле:

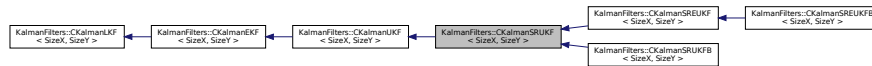
- kalman\_filter\_extended\_unscented\_square\_root.h

### 8.13 Шаблон класса KalmanFilters::CKalmanSRUKF< SizeX, SizeY >

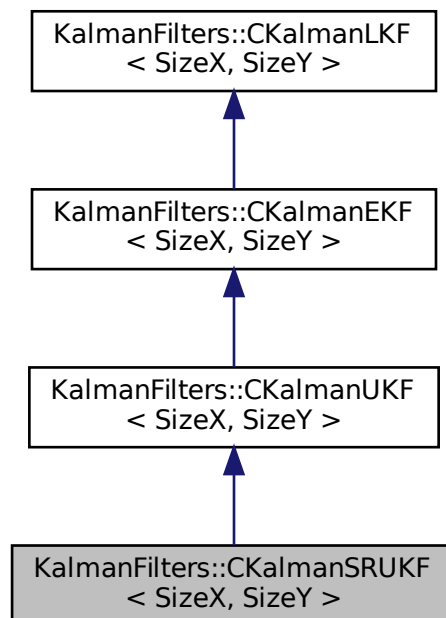
Шаблонный класс квадратно-корневого сигма-точечного (ансцентного) фильтра Калмана, КК-СТФК (КК-АФК) (Square Root Unscented Kalman Filter, SR-UKF)

#include <kalman\_filter\_unscented\_square\_root.h>

Граф наследования: KalmanFilters::CKalmanSRUKF< SizeX, SizeY >:



Граф связей класса KalmanFilters::CKalmanSRUKF< SizeX, SizeY >:



#### Открытые члены

- [CKalmanSRUKF](#) ()  
Конструктор по умолчанию
- [CKalmanSRUKF](#) (const [CKalmanSRUKF](#) &other)  
Конструктор копирования
- [CKalmanSRUKF](#) & operator= (const [CKalmanSRUKF](#) &other)  
Перегрузка оператора присвоения
- [CKalmanSRUKF](#) ([CKalmanSRUKF](#) &&other) noexcept  
Конструктор перемещения
- [CKalmanSRUKF](#) & operator= ([CKalmanSRUKF](#) &&other) noexcept  
Перегрузка оператора перемещения
- virtual [~CKalmanSRUKF](#) ()=default  
Деструктор

- virtual void [SetupDesignParametersMeanSet](#) (double w0)  
Установка параметра w0 ансцентного фильтра (MeanSet)
- virtual void [SetupDesignParametersScaledSet](#) (double alpha, double beta, double kappa)  
Установка параметров масштабируемого ансцентного преобразования (Scaled UT)
- virtual void [SetupDesignParametersCDKF](#) (double h2)  
Установка параметра фильтра по рекомендации Central Difference Kalman Filter (CDKF)
- void [SetStateTransitionJacobianF](#) (std::function< arma::mat(const arma::vec &X, double dt)> stateTransitionJacobianF)  
Установка функции вычисления матрицы перехода состояния F (makeMatrixF)
- void [SetObservationJacobianH](#) (std::function< arma::mat(const arma::vec &X)> observationJacobianH)  
Установка функции вычисления матрицы перехода измерений H (makeMatrixH)
- virtual void [Prediction](#) (double dt)  
Прогноз
- virtual void [Correction](#) (const arma::vec &Y\_msd)  
Коррекция

### Защищенные члены

- void [CorrectionSRUKF](#) (const arma::vec &Y\_msd)  
Коррекция SRUKF.
- void [createSignMatrices](#) ()  
Создание матриц знаков

### Защищенные данные

- arma::mat [J\\_](#)  
Матрица знаков при Pxy.
- arma::mat [Jpredict\\_](#)  
Матрица знаков при прогнозе
- arma::mat [Jcorrect\\_](#)  
Матрица знаков при коррекции
- bool [negativeZeroCovWeight\\_](#)  
Признак отрицательного веса "нулевой" сигма-точки Wcov.

### Друзья

- void [swap](#) (CKalmanSRUKF< SizeX, SizeY > &lhs, CKalmanSRUKF< SizeX, SizeY > &rhs)  
noexcept  
Метод свапа

#### 8.13.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanSRUKF< SizeX, SizeY >
```

Шаблонный класс квадратно-корневого сигма-точечного (ансцентного) фильтра Калмана, КК-СТФК (КК-АФК) (Square Root Unscented Kalman Filter, SR-UKF)

Источники:

- [1] A New Extension of the Kalman Filter to Nonlinear Systems, Simon J. Julier, Jeffrey K. Uhlmann, The Robotics Research Group, Department of Engineering Science, The University of Oxford, 1997
- [2] The Unscented Kalman Filter for Nonlinear Estimation, Eric A. Wan and Rudolph van der Merwe Oregon Graduate Institute of Science & Technology 20000 NW Walker Rd, Beaverton, Oregon 97006, 2000 [ericwan@ece.ogi.edu](mailto:ericwan@ece.ogi.edu), [rvdmerwe@ece.ogi.edu](mailto:rvdmerwe@ece.ogi.edu)
- [3] Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models, Rudolph

van der Merwe & Eric Wan, OGI School of Science & Engineering Oregon Health & Science University Beaverton, Oregon, 97006, USA, 2003 {rvdmerwe,ericwan}@ece.ogi.edu

[4] THE SQUARE-ROOT UNSCENTED KALMAN FILTER FOR STATE AND PARAMETER-ESTIMATION, Rudolph van der Merwe and Eric A. Wan, Oregon Graduate Institute of Science and Technology 20000 NW Walker Road, Beaverton, Oregon 97006, USA rvdmerwe,ericwan@ece.ogi.edu

[5] The J-Orthogonal Square-Root Euler-Maruyama-Based Unscented Kalman Filter for Nonlinear Stochastic Systems, Gennady Yu. Kulikov, Maria V. Kulikova, CEMAT, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 LISBOA, Portugal (emails: gennady.kulikov@tecnico.ulisboa.pt, maria.kulikova@ist.utl.pt)

Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

## 8.13.2 Конструктор(ы)

### 8.13.2.1 CKalmanSRUKF() [1/3]

```
template<size_t SizeX, size_t SizeY>
```

```
KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::CKalmanSRUKF ( ) [inline]
```

Конструктор по умолчанию

### 8.13.2.2 CKalmanSRUKF() [2/3]

```
template<size_t SizeX, size_t SizeY>
```

```
KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::CKalmanSRUKF (
    const CKalmanSRUKF< SizeX, SizeY > & other ) [inline]
```

Конструктор копирования

Аргументы

other	- экземпляр, с которого делается копия
-------	--

### 8.13.2.3 CKalmanSRUKF() [3/3]

```
template<size_t SizeX, size_t SizeY>
```

```
KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::CKalmanSRUKF (
    CKalmanSRUKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Конструктор перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

### 8.13.2.4 ~CKalmanSRUKF()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::~~CKalmanSRUKF ( ) [virtual], [default]
```

Деструктор

### 8.13.3 Методы

#### 8.13.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd ) [inline], [virtual]
```

Коррекция

Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanUKF](#)< SizeX, SizeY >.

Переопределяется в [KalmanFilters::CKalmanSREUKF](#)< SizeX, SizeY >, [KalmanFilters::CKalmanSREUKFB](#)< SizeX, SizeY > и [KalmanFilters::CKalmanSRUKFB](#)< SizeX, SizeY >.

#### 8.13.3.2 CorrectionSRUKF()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::CorrectionSRUKF (
    const arma::vec & Y_msd ) [inline], [protected]
```

Коррекция SRUKF.

Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

#### 8.13.3.3 createSignMatrices()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::createSignMatrices ( ) [inline], [protected]
```

Создание матриц знаков

#### 8.13.3.4 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
CKalmanSRUKF & KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::operator= (
    CKalmanSRUKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Перегрузка оператора перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

#### 8.13.3.5 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
```



```
CKalmanSRUKF & KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::operator= (
    const CKalmanSRUKF< SizeX, SizeY > & other ) [inline]
```

Перегрузка оператора присвоения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

#### 8.13.3.6 Prediction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::Prediction (
    double dt ) [inline], [virtual]
```

Прогноз

Аргументы

dt	- Время прогноза, [с]
----	-----------------------

Переопределяет метод предка [KalmanFilters::CKalmanUKF< SizeX, SizeY >](#).

Переопределяется в [KalmanFilters::CKalmanSREUKF< SizeX, SizeY >](#).

#### 8.13.3.7 SetObservationJacobianH()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::SetObservationJacobianH (
    std::function< arma::mat(const arma::vec &X)> observationJacobianH ) [inline]
```

Установка функции вычисления матрицы перехода измерений H (makeMatrixH)

См. также

[observationJacobianH\\_](#)

#### 8.13.3.8 SetStateTransitionJacobianF()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::SetStateTransitionJacobianF (
    std::function< arma::mat(const arma::vec &X, double dt)> stateTransitionJacobianF ) [inline]
```

Установка функции вычисления матрицы перехода состояния F (makeMatrixF)

См. также

[stateTransitionJacobianF\\_](#)

#### 8.13.3.9 SetupDesignParametersCDKF()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::SetupDesignParametersCDKF (
    double h2 ) [inline], [virtual]
```

Установка параметра фильтра по рекомендации Central Difference Kalman Filter (CDKF)

Смотри [3]

## Аргументы

h2	- параметр разброса сигма-точек ( $h^2 = 3$ типичная рекомендация для гауссовых шумов, [3])
----	---

Переопределяет метод предка [KalmanFilters::CKalmanUKF< SizeX, SizeY >](#).

## 8.13.3.10 SetupDesignParametersMeanSet()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::SetupDesignParametersMeanSet (
    double w0 ) [inline], [virtual]
```

Установка параметра  $w_0$  апсцентного фильтра (MeanSet)

При выборе  $w_0 = [0...1]$  обеспечиваются положительные веса. Смотри [1], [2]

## Внимание

Нельзя выбирать  $w_0$  так, чтобы нулевой вес был  $> 0$ , а остальные меньше нуля. Наоборот - МОЖНО, т.е. нулевой вес может быть отрицательным.

## Аргументы

w0	- параметр разброса сигма точек ( $w_0 = [0...1]$ типичная рекомендация для положительных весов)
----	--

Переопределяет метод предка [KalmanFilters::CKalmanUKF< SizeX, SizeY >](#).

## 8.13.3.11 SetupDesignParametersScaledSet()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::SetupDesignParametersScaledSet (
    double alpha,
    double beta,
    double kappa ) [inline], [virtual]
```

Установка параметров масштабируемого апсцентного преобразования (Scaled UT)

Смотри [2], [5], [6]

## Аргументы

alpha	- параметр разброса сигма-точек ( $\alpha = 10^{-3}$ - типичная рекомендация по van der Merwe)
beta	- параметр, отвечающий за характер распределения ( $\beta = 2$ - нормальное)
kappa	- параметр, отвечающий за разброс сигма-точек ( $\kappa = 0$ или $(3 - \text{SizeX})$ - типичная рекомендация по van der Merwe)

Переопределяет метод предка [KalmanFilters::CKalmanUKF< SizeX, SizeY >](#).

## 8.13.4 Документация по друзьям класса и функциям, относящимся к классу

## 8.13.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
```

```

    CKalmanSRUKF< SizeX, SizeY > & lhs,
    CKalmanSRUKF< SizeX, SizeY > & rhs ) [friend]

```

Метод свапа

Аргументы

lhs	- left hand side instance
rhs	- right hand side instance

### 8.13.5 Данные класса

#### 8.13.5.1 J\_

```

template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::J_ [protected]

```

Матрица знаков при Pxy.

#### 8.13.5.2 Jcorrect\_

```

template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::Jcorrect_ [protected]

```

Матрица знаков при коррекции

#### 8.13.5.3 Jpredict\_

```

template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::Jpredict_ [protected]

```

Матрица знаков при прогнозе

#### 8.13.5.4 negativeZeroCovWeight\_

```

template<size_t SizeX, size_t SizeY>
bool KalmanFilters::CKalmanSRUKF< SizeX, SizeY >::negativeZeroCovWeight_ [protected]

```

Признак отрицательного веса "нулевой" сигма-точки Wcov.

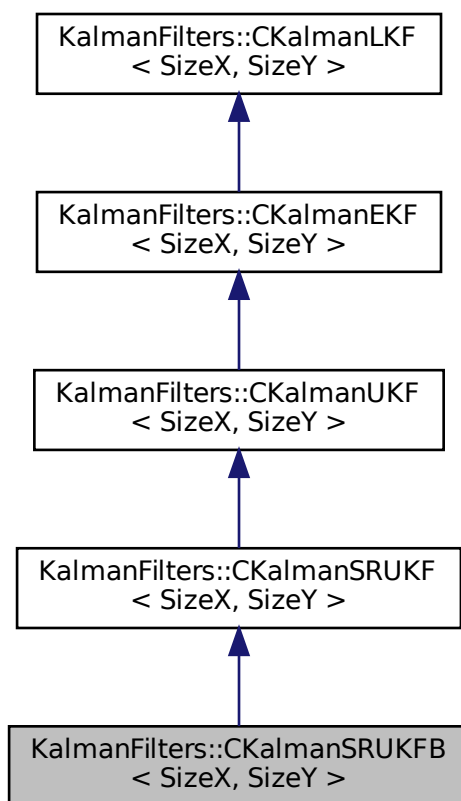
Объявления и описания членов класса находятся в файле:

- kalman\_filter\_unscented\_square\_root.h

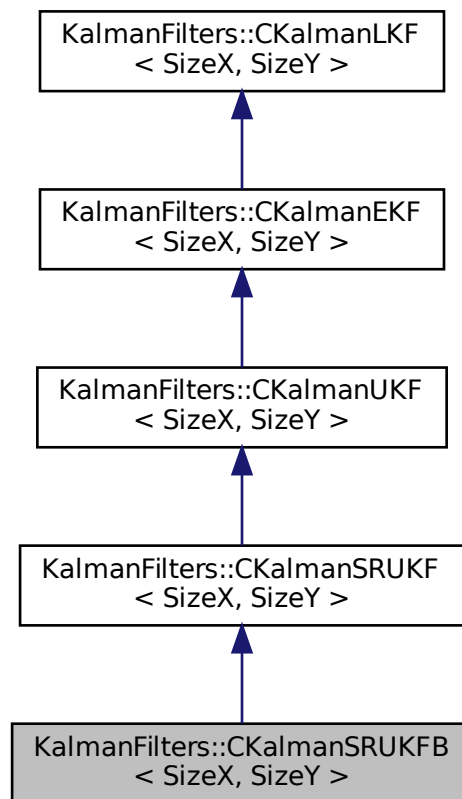
## 8.14 Шаблон класса KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >

Шаблонный класс квадратно-корневого сигма-точечного (ансцентного) фильтра Калмана (блочная фильтрация), КК-СТФКБ (КК-АФКБ) (Square Root Unscented Kalman Filter Block, SR-UKFB)  
#include <kalman\_filter\_unscented\_square\_root.h>

Граф наследования: KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >:



Граф связей класса KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >:



#### Открытые члены

- `CKalmanSRUKFB ()`  
Конструктор по умолчанию
- `CKalmanSRUKFB (const CKalmanSRUKFB &other)`  
Конструктор копирования
- `CKalmanSRUKFB & operator= (const CKalmanSRUKFB &other)`  
Перегрузка оператора присвоения
- `CKalmanSRUKFB (CKalmanSRUKFB &&other) noexcept`  
Конструктор перемещения
- `CKalmanSRUKFB & operator= (CKalmanSRUKFB &&other) noexcept`  
Перегрузка оператора перемещения
- `virtual ~CKalmanSRUKFB ()=default`  
Деструктор
- `virtual void Correction (const arma::vec &Y_msd)`  
Коррекция квадратно-корневого сигма-точечного фильтра Калмана (КК-СТФК, SR-UKF)

#### Защищенные члены

- `void createSignMatricesBlock ()`  
Создание матриц знаков

## Защищенные данные

- `arma::mat JcorrectBlock_`  
Матрица знаков для фильтра в блочном виде

## Друзья

- `void swap (CKalmanSRUKFB &lhs, CKalmanSRUKFB &rhs)` `noexcept`  
Метод свапа

### 8.14.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >
```

Шаблонный класс квадратно-корневого сигма-точечного (ансцентного) фильтра Калмана (блочная фильтрация), КК-СТФКБ (КК-АФКБ) (Square Root Unscented Kalman Filter Block, SR-UKFB) см. [CKalmanSRUKF](#)

Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

### 8.14.2 Конструктор(ы)

#### 8.14.2.1 CKalmanSRUKFB() [1/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >::CKalmanSRUKFB ( ) [inline]
Конструктор по умолчанию
```

#### 8.14.2.2 CKalmanSRUKFB() [2/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >::CKalmanSRUKFB (
    const CKalmanSRUKFB< SizeX, SizeY > & other ) [inline]
Конструктор копирования
```

Аргументы

other	- экземпляр, с которого делается копия
-------	--

#### 8.14.2.3 CKalmanSRUKFB() [3/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >::CKalmanSRUKFB (
    CKalmanSRUKFB< SizeX, SizeY > && other ) [inline], [noexcept]
Конструктор перемещения
```

Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.14.2.4 ~CKalmanSRUKFB()

```
template<size_t SizeX, size_t SizeY>
virtual KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >::~~CKalmanSRUKFB ( ) [virtual], [default]
Деструктор
```

## 8.14.3 Методы

## 8.14.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd ) [inline], [virtual]
```

Коррекция квадратно-корневого сигма-точечного фильтра Калмана (КК-СТФК, SR-UKF)

Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanSRUKF< SizeX, SizeY >](#).

## 8.14.3.2 createSignMatricesBlock()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >::createSignMatricesBlock ( ) [inline], [protected]
Создание матриц знаков
```

## 8.14.3.3 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
CKalmanSRUKFB & KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >::operator= (
    CKalmanSRUKFB< SizeX, SizeY > && other ) [inline], [noexcept]
```

Перегрузка оператора перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

## 8.14.3.4 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
CKalmanSRUKFB & KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >::operator= (
    const CKalmanSRUKFB< SizeX, SizeY > & other ) [inline]
```

Перегрузка оператора присвоения

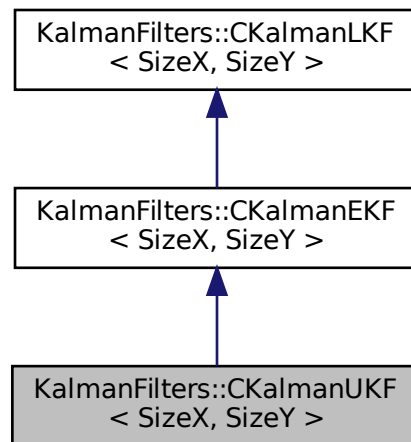
Аргументы

other	- экземпляр, с которого делается копия
-------	--





Граф связей класса KalmanFilters::CKalmanUKF< SizeX, SizeY >:



#### Открытые члены

- [CKalmanUKF](#) ()  
Конструктор по умолчанию
- [CKalmanUKF](#) (const [CKalmanUKF](#) &other)  
Конструктор копирования
- [CKalmanUKF](#) & [operator=](#) (const [CKalmanUKF](#) &other)  
Перегрузка оператора присвоения
- [CKalmanUKF](#) ([CKalmanUKF](#) &&other) noexcept  
Конструктор перемещения
- [CKalmanUKF](#) & [operator=](#) ([CKalmanUKF](#) &&other) noexcept  
Перегрузка оператора перемещения
- virtual [~CKalmanUKF](#) ()=default  
Деструктор
- virtual void [SetupDesignParametersMeanSet](#) (double w0)  
Установка параметра w0 апостериорного фильтра (MeanSet)
- virtual void [SetupDesignParametersScaledSet](#) (double alpha, double beta, double kappa)  
Установка параметров масштабируемого апостериорного преобразования (Scaled UT)
- virtual void [SetupDesignParametersCDKF](#) (double h2)  
Установка параметра фильтра по рекомендации Central Difference Kalman Filter (CDKF)
- void [SetWeightedSumStateSigmas](#) (std::function< arma::vec(const arma::vec &weights, const arma::mat &sigmaPoints)> weightedSumStateSigmas)  
Установка функции вычисления взвешенной суммы сигма-точек пространства X
- void [SetWeightedSumMeasurementSigmas](#) (std::function< arma::vec(const arma::vec &weights, const arma::mat &sigmaPoints)> weightedSumMeasurementSigmas)  
Установка функции вычисления взвешенной суммы сигма-точек пространства Y.
- virtual void [Prediction](#) (double dt)  
Прогноз
- virtual void [Correction](#) (const arma::vec &Y\_msd)  
Коррекция

## Защищенные члены

- void [PredictionUKF](#) (double dt)  
Прогноз UKF.
- void [CorrectionUKF](#) (const arma::vec &Y\_msd)  
Коррекция UKF.
- void [SetStateTransitionJacobianF](#) ()  
Запрет доступа
- void [SetObservationJacobianH](#) ()  
Запрет доступа

## Защищенные данные

- int [k\\_sigma\\_points\\_](#)  
Число сигма-точек
- arma::vec [weights\\_mean\\_](#)  
Веса среднего
- arma::vec [weights\\_covariance\\_](#)  
Веса ковариации
- arma::mat [x\\_est\\_sigma\\_points\\_](#)  
Матрица сигма-точек (сигма-точки - столбцы) в пространстве X на текущем такте, размерность [SizeX,k\_sigma\_points\_].
- arma::mat [x\\_pred\\_sigma\\_points\\_](#)  
Матрица сигма-точек (сигма-точки - столбцы) в пространстве X, экстраполированный на текущий такт, размерность [SizeX,k\_sigma\_points\_].
- arma::mat [y\\_pred\\_sigma\\_points\\_](#)  
Матрица сигма-точек (сигма-точки - столбцы) в пространстве Y, экстраполированный на текущий такт, размерность [SizeX,k\_sigma\_points\_].
- arma::mat [dXcal\\_](#)  
Матрица X-каллиграфическое (матрица сигма-точек - столбцов)
- arma::mat [dYcal\\_](#)  
Матрица Y-каллиграфическое (матрица сигма-точек - столбцов)
- arma::mat [P\\_xy\\_](#)  
Матрица кросс-ковариации векторов X и Y, размерность [SizeX \* SizeY].
- arma::mat [sqrt\\_P\\_chol\\_](#)  
Корень из матрицы P.
- double [alpha\\_](#)  
Параметр разброса сигма-точек ( $\alpha = 10^{-3}$  типичная рекомендация)
- double [kappa\\_](#)  
Параметр разброса сигма-точек ( $\kappa = 3 - \text{SizeX}$  типичная рекомендация)
- double [beta\\_](#)  
Параметр разброса сигма-точек ( $\beta = 2$  - нормальное, 0 - нет сведений о распределении)
- double [lambda\\_](#)  
Автоматически вычисляемый параметр, равный  $(\alpha * \alpha) * (\text{SizeX} + \kappa) - \text{SizeX}$ .
- double [w0\\_](#)  
Параметр разброса сигма-точек (0..1)
- double [gamma\\_](#)  
Автоматически вычисляемый (в методах SetDesignParameters\*) параметр (множитель при корне из P при создании сигма-точек)
- std::function< arma::vec(const arma::vec &weights, const arma::mat &sigmaPoints)> [weightedSumStateSigmas\\_](#)  
Вычисление взвешенной суммы сигма-точек пространства X
- std::function< arma::vec(const arma::vec &weights, const arma::mat &sigmaPoints)> [weightedSumMeasurementSigmas\\_](#)  
Вычисление взвешенной суммы сигма-точек пространства Y.

## Друзья

- void **swap** (CKalmanUKF< SizeX, SizeY > &lhs, CKalmanUKF< SizeX, SizeY > &rhs) noexcept  
Метод свапа

## 8.15.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class KalmanFilters::CKalmanUKF< SizeX, SizeY >
```

Шаблонный класс сигма-точечного (ансцентного) фильтра Калмана, СТФК (АФК) (Unscented Kalman Filter, UKF)

Источники:

- [1] A New Extension of the Kalman Filter to Nonlinear Systems, Simon J. Julier, Jeffrey K. Uhlmann, The Robotics Research Group, Department of Engineering Science, The University of Oxford, 1997
- [2] Julier S.J., Uhlmann J.K. Unscented filtering and nonlinear estimation // Proc. of the IEEE, 2004, №3.P.401-422
- [3] The Unscented Kalman Filter for Nonlinear Estimation, Eric A. Wan and Rudolph van der Merwe Oregon Graduate Institute of Science & Technology 20000 NW Walker Rd, Beaverton, Oregon 97006, 2000 ericwan[at]ece.ogi.edu, rvdmerwe[at]ece.ogi.edu
- [4] Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models, Rudolph van der Merwe & Eric Wan, OGI School of Science & Engineering Oregon Health & Science University Beaverton, Oregon, 97006, USA, 2003 {rvdmerwe,ericwan}[at]ece.ogi.edu
- [5] THE SQUARE-ROOT UNSCENTED KALMAN FILTER FOR STATE AND PARAMETER-ESTIMATION, Rudolph van der Merwe and Eric A. Wan, Oregon Graduate Institute of Science and Technology 20000 NW Walker Road, Beaverton, Oregon 97006, USA rvdmerwe,ericwan[at]ece.ogi.edu
- [6] Sebastian Bitzer, Technische Universität Dresden, <https://github.com/sbitzer/UKF-exposed/blob/master/UKF.pdf>

Внимание

Внимание! В реализации UKF вес нулевой сигма-точки  $W_{cov}$  не может быть отрицательным! ( $W_{mean}$  - может)

Фильтр построен по классическому его варианту НИЖНЕтреугольного разложения Холецкого!

Параметры шаблона

SizeX	- размерность пространства состояния X
SizeY	- размерность пространства измерений Y

## 8.15.2 Конструктор(ы)

## 8.15.2.1 CKalmanUKF() [1/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanUKF< SizeX, SizeY >::CKalmanUKF ( ) [inline]
Конструктор по умолчанию
```

## 8.15.2.2 CKalmanUKF() [2/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanUKF< SizeX, SizeY >::CKalmanUKF (
    const CKalmanUKF< SizeX, SizeY > & other ) [inline]
Конструктор копирования
```

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.15.2.3 CKalmanUKF() [3/3]

```
template<size_t SizeX, size_t SizeY>
KalmanFilters::CKalmanUKF< SizeX, SizeY >::CKalmanUKF (
    CKalmanUKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Конструктор перемещения

## Аргументы

other	- экземпляр, с которого делается копия
-------	--

## 8.15.2.4 ~CKalmanUKF()

```
template<size_t SizeX, size_t SizeY>
virtual KalmanFilters::CKalmanUKF< SizeX, SizeY >::~~CKalmanUKF ( ) [virtual], [default]
```

Деструктор

## 8.15.3 Методы

## 8.15.3.1 Correction()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanUKF< SizeX, SizeY >::Correction (
    const arma::vec & Y_msd ) [inline], [virtual]
```

Коррекция

## Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

Переопределяет метод предка [KalmanFilters::CKalmanEKF< SizeX, SizeY >](#).

Переопределяется в [KalmanFilters::CKalmanEUKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSREUKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >](#), [KalmanFilters::CKalmanSRUKF< SizeX, SizeY >](#) и [KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >](#).

## 8.15.3.2 CorrectionUKF()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanUKF< SizeX, SizeY >::CorrectionUKF (
    const arma::vec & Y_msd ) [inline], [protected]
```

Коррекция UKF.

## Аргументы

Y_msd	- вектор измерений, по которым производится коррекция
-------	---

## 8.15.3.3 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanUKF & KalmanFilters::CKalmanUKF< SizeX, SizeY >::operator= (
    CKalmanUKF< SizeX, SizeY > && other ) [inline], [noexcept]
```

Перегрузка оператора перемещения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

## 8.15.3.4 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
```

```
CKalmanUKF & KalmanFilters::CKalmanUKF< SizeX, SizeY >::operator= (
    const CKalmanUKF< SizeX, SizeY > & other ) [inline]
```

Перегрузка оператора присвоения

Аргументы

other	- экземпляр, с которого делается копия
-------	--

Возвращает

\*this

## 8.15.3.5 Prediction()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual void KalmanFilters::CKalmanUKF< SizeX, SizeY >::Prediction (
    double dt ) [inline], [virtual]
```

Прогноз

Аргументы

dt	- Время прогноза, [с]
----	-----------------------

Переопределяет метод предка [KalmanFilters::CKalmanEKF< SizeX, SizeY >](#).

Переопределяется в [KalmanFilters::CKalmanEUKF< SizeX, SizeY >](#), [KalmanFilters::CKalmanSREUKF< SizeX, SizeY >](#) и [KalmanFilters::CKalmanSRUKF< SizeX, SizeY >](#).

## 8.15.3.6 PredictionUKF()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanUKF< SizeX, SizeY >::PredictionUKF (
    double dt ) [inline], [protected]
```

Прогноз UKF.

## Аргументы

dt	- Время прогноза, [с]
----	-----------------------

## 8.15.3.7 SetObservationJacobianH()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanUKF< SizeX, SizeY >::SetObservationJacobianH ( ) [inline], [protected]
```

Запрет доступа

## 8.15.3.8 SetStateTransitionJacobianF()

```
template<size_t SizeX, size_t SizeY>
```

```
void KalmanFilters::CKalmanUKF< SizeX, SizeY >::SetStateTransitionJacobianF ( ) [inline], [protected]
```

Запрет доступа

## 8.15.3.9 SetupDesignParametersCDKF()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual void KalmanFilters::CKalmanUKF< SizeX, SizeY >::SetupDesignParametersCDKF (
    double h2 ) [inline], [virtual]
```

Установка параметра фильтра по рекомендации Central Difference Kalman Filter (CDKF)

Смотри [4]

## Аргументы

h2	- параметр разброса сигма-точек ( $h^2 = 3$ типичная рекомендация для гауссовых шумов, [3])
----	---

Переопределяется в [KalmanFilters::CKalmanSRUKF< SizeX, SizeY >](#).

## 8.15.3.10 SetupDesignParametersMeanSet()

```
template<size_t SizeX, size_t SizeY>
```

```
virtual void KalmanFilters::CKalmanUKF< SizeX, SizeY >::SetupDesignParametersMeanSet (
    double w0 ) [inline], [virtual]
```

Установка параметра  $w_0$  апостериорного фильтра (MeanSet)

При выборе  $w_0 = [0...1]$  обеспечиваются положительные веса. Смотри [1], [2]

## Внимание

Нельзя выбирать  $w_0$  так, чтобы нулевой вес был  $> 0$ , а остальные меньше нуля. Наоборот - МОЖНО, т.е. нулевой вес может быть отрицательным.

## Аргументы

w0	- параметр разброса сигма точек ( $w_0 = [0...1]$ типичная рекомендация для положительных весов)
----	--

Переопределяется в [KalmanFilters::CKalmanSRUKF< SizeX, SizeY >](#).

## 8.15.3.11 SetupDesignParametersScaledSet()

```
template<size_t SizeX, size_t SizeY>
virtual void KalmanFilters::CKalmanUKF< SizeX, SizeY >::SetupDesignParametersScaledSet (
    double alpha,
    double beta,
    double kappa ) [inline], [virtual]
```

Установка параметров масштабируемого апсцентного преобразования (Scaled UT)

Смотри [2], [5], [6]

Аргументы

alpha	- параметр разброса сигма-точек ( $\alpha = 10^{-3}$ - типичная рекомендация по van der Merwe)
beta	- параметр, отвечающий за характер распределения ( $\beta = 2$ - нормальное)
каппа	- параметр, отвечающий за разброс сигма-точек ( $\kappa = 0$ или $(3 - \text{SizeX})$ - типичная рекомендация по van der Merwe)

Переопределяется в [KalmanFilters::CKalmanSRUKF< SizeX, SizeY >](#).

## 8.15.3.12 SetWeightedSumMeasurementSigmas()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanUKF< SizeX, SizeY >::SetWeightedSumMeasurementSigmas (
    std::function< arma::vec(const arma::vec &weights, const arma::mat &sigmaPoints)> weightedSum←
    MeasurementSigmas ) [inline]
```

Установка функции вычисления взвешенной суммы сигма-точек пространства Y.

Аргументы

weightedSumMeasurementSigmas	- Функция вычисления взвешенной суммы сигма-точек пространства Y
------------------------------	--

## 8.15.3.13 SetWeightedSumStateSigmas()

```
template<size_t SizeX, size_t SizeY>
void KalmanFilters::CKalmanUKF< SizeX, SizeY >::SetWeightedSumStateSigmas (
    std::function< arma::vec(const arma::vec &weights, const arma::mat &sigmaPoints)> weightedSumState←
    Sigmas ) [inline]
```

Установка функции вычисления взвешенной суммы сигма-точек пространства X

Аргументы

weightedSumStateSigmas	- Функция вычисления взвешенной суммы сигма-точек пространства X
------------------------	--

## 8.15.4 Документация по друзьям класса и функциям, относящимся к классу

## 8.15.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
```

```

    CKalmanUKF< SizeX, SizeY > & lhs,
    CKalmanUKF< SizeX, SizeY > & rhs )    [friend]

```

Метод свапа

Аргументы

lhs	- left hand side instance
rhs	- right hand side instance

### 8.15.5 Данные класса

#### 8.15.5.1 alpha\_

```
template<size_t SizeX, size_t SizeY>
```

```
double KalmanFilters::CKalmanUKF< SizeX, SizeY >::alpha_    [protected]
```

Параметр разброса сигма-точек ( $\alpha = 10^{-3}$  типичная рекомендация)

#### 8.15.5.2 beta\_

```
template<size_t SizeX, size_t SizeY>
```

```
double KalmanFilters::CKalmanUKF< SizeX, SizeY >::beta_    [protected]
```

Параметр разброса сигма-точек ( $\beta = 2$  - нормальное, 0 - нет сведений о распределении)

#### 8.15.5.3 dXcal\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::mat KalmanFilters::CKalmanUKF< SizeX, SizeY >::dXcal_    [protected]
```

Матрица X-калиграфическое (матрица сигма-точек - столбцов)

#### 8.15.5.4 dYcal\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::mat KalmanFilters::CKalmanUKF< SizeX, SizeY >::dYcal_    [protected]
```

Матрица Y-калиграфическое (матрица сигма-точек - столбцов)

#### 8.15.5.5 gamma\_

```
template<size_t SizeX, size_t SizeY>
```

```
double KalmanFilters::CKalmanUKF< SizeX, SizeY >::gamma_    [protected]
```

Автоматически вычисляемый (в методах SetDesignParameters\*) параметр (множитель при корне из Р при создании сигма-точек)

#### 8.15.5.6 k\_sigma\_points\_

```
template<size_t SizeX, size_t SizeY>
```

```
int KalmanFilters::CKalmanUKF< SizeX, SizeY >::k_sigma_points_    [protected]
```

Число сигма-точек



## 8.15.5.7 kappa\_

```
template<size_t SizeX, size_t SizeY>
```

```
double KalmanFilters::CKalmanUKF< SizeX, SizeY >::kappa_ [protected]
```

Параметр разброса сигма-точек (kappa = 3 - SizeX типичная рекомендация)

## 8.15.5.8 lambda\_

```
template<size_t SizeX, size_t SizeY>
```

```
double KalmanFilters::CKalmanUKF< SizeX, SizeY >::lambda_ [protected]
```

Автоматически вычисляемый параметр, равный  $(\alpha * \alpha) * (SizeX + kappa) - SizeX$ ;

## 8.15.5.9 P\_xy\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::mat KalmanFilters::CKalmanUKF< SizeX, SizeY >::P_xy_ [protected]
```

Матрица кросс-ковариации векторов X и Y, размерность [SizeX \* SizeY].

## 8.15.5.10 sqrt\_P\_chol\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::mat KalmanFilters::CKalmanUKF< SizeX, SizeY >::sqrt_P_chol_ [protected]
```

Корень из матрицы P.

## 8.15.5.11 w0\_

```
template<size_t SizeX, size_t SizeY>
```

```
double KalmanFilters::CKalmanUKF< SizeX, SizeY >::w0_ [protected]
```

Параметр разброса сигма-точек (0..1)

## 8.15.5.12 weightedSumMeasurementSigmas\_

```
template<size_t SizeX, size_t SizeY>
```

```
std::function<arma::vec( const arma::vec &weights, const arma::mat &sigmaPoints )> KalmanFilters::CKalmanUKF< SizeX, SizeY >::weightedSumMeasurementSigmas_ [protected]
```

Вычисление взвешенной суммы сигма-точек пространства Y.

## 8.15.5.13 weightedSumStateSigmas\_

```
template<size_t SizeX, size_t SizeY>
```

```
std::function<arma::vec( const arma::vec &weights, const arma::mat &sigmaPoints )> KalmanFilters::CKalmanUKF< SizeX, SizeY >::weightedSumStateSigmas_ [protected]
```

Вычисление взвешенной суммы сигма-точек пространства X

## 8.15.5.14 weights\_covariance\_

```
template<size_t SizeX, size_t SizeY>
```

```
arma::vec KalmanFilters::CKalmanUKF< SizeX, SizeY >::weights_covariance_ [protected]
```

Веса ковариации

8.15.5.15 `weights_mean_`

```
template<size_t SizeX, size_t SizeY>
arma::vec KalmanFilters::CKalmanUKF< SizeX, SizeY >::weights_mean_ [protected]
Веса среднего
```

8.15.5.16 `x_est_sigma_points_`

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanUKF< SizeX, SizeY >::x_est_sigma_points_ [protected]
Матрица сигма-точек (сигма-точки - столбцы) в пространстве X на текущем такте, размерность [SizeX,k_sigma_points_].
```

8.15.5.17 `x_pred_sigma_points_`

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanUKF< SizeX, SizeY >::x_pred_sigma_points_ [protected]
Матрица сигма-точек (сигма-точки - столбцы) в пространстве X, экстраполированный на текущий такт, размерность [SizeX,k_sigma_points_].
```

8.15.5.18 `y_pred_sigma_points_`

```
template<size_t SizeX, size_t SizeY>
arma::mat KalmanFilters::CKalmanUKF< SizeX, SizeY >::y_pred_sigma_points_ [protected]
Матрица сигма-точек (сигма-точки - столбцы) в пространстве Y, экстраполированный на текущий такт, размерность [SizeX,k_sigma_points_].
Объявления и описания членов класса находятся в файле:
```

- `kalman_filter_unscented.h`

# Предметный указатель

~CKalmanCKF KalmanFilters::CKalmanCKF< SizeX, SizeY >, <a href="#">22</a>	CalculateInnovationCovarianceS KalmanFilters::CKalmanLKF< SizeX, SizeY >, <a href="#">48</a>
~CKalmanECKF KalmanFilters::CKalmanECKF< SizeX, SizeY >, <a href="#">30</a>	checkBordersMeasurement_ KalmanFilters::CKalmanLKF< SizeX, SizeY >, <a href="#">56</a>
~CKalmanEKF KalmanFilters::CKalmanEKF< SizeX, SizeY >, <a href="#">34</a>	checkBordersStateAfterCorrection_ KalmanFilters::CKalmanLKF< SizeX, SizeY >, <a href="#">56</a>
~CKalmanEUKF KalmanFilters::CKalmanEUKF< SizeX, SizeY >, <a href="#">42</a>	checkBordersStateAfterPrediction_ KalmanFilters::CKalmanLKF< SizeX, SizeY >, <a href="#">56</a>
~CKalmanLKF KalmanFilters::CKalmanLKF< SizeX, SizeY >, <a href="#">48</a>	checkDeltaMeasurement_ KalmanFilters::CKalmanLKF< SizeX, SizeY >, <a href="#">56</a>
~CKalmanSRCKF KalmanFilters::CKalmanSRCKF< SizeX, SizeY >, <a href="#">62</a>	checkDeltaState_ KalmanFilters::CKalmanLKF< SizeX, SizeY >, <a href="#">56</a>
~CKalmanSRECKF KalmanFilters::CKalmanSRECKF< SizeX, SizeY >, <a href="#">69</a>	checkMatrixDiagPositive KalmanFilters::CKalmanLKF< SizeX, SizeY >, <a href="#">48</a>
~CKalmanSRECKFB KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >, <a href="#">73</a>	CKalmanCKF KalmanFilters::CKalmanCKF< SizeX, SizeY >, <a href="#">21</a> , <a href="#">22</a>
~CKalmanSREKF KalmanFilters::CKalmanSREKF< SizeX, SizeY >, <a href="#">76</a>	CKalmanECKF KalmanFilters::CKalmanECKF< SizeX, SizeY >, <a href="#">29</a>
~CKalmanSREUKF KalmanFilters::CKalmanSREUKF< SizeX, SizeY >, <a href="#">81</a>	CKalmanEKF KalmanFilters::CKalmanEKF< SizeX, SizeY >, <a href="#">34</a>
~CKalmanSREUKFB KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >, <a href="#">85</a>	CKalmanEUKF KalmanFilters::CKalmanEUKF< SizeX, SizeY >, <a href="#">41</a> , <a href="#">42</a>
~CKalmanSRUKF KalmanFilters::CKalmanSRUKF< SizeX, SizeY >, <a href="#">89</a>	CKalmanLKF KalmanFilters::CKalmanLKF< SizeX, SizeY >, <a href="#">47</a>
~CKalmanSRUKFB KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >, <a href="#">97</a>	CKalmanSRCKF KalmanFilters::CKalmanSRCKF< SizeX, SizeY >, <a href="#">61</a>
~CKalmanUKF KalmanFilters::CKalmanUKF< SizeX, SizeY >, <a href="#">102</a>	CKalmanSRCKFB KalmanFilters::CKalmanSRCKFB< SizeX, SizeY >, <a href="#">67</a>
alpha_ KalmanFilters::CKalmanUKF< SizeX, SizeY >, <a href="#">106</a>	CKalmanSRECKF KalmanFilters::CKalmanSRECKF< SizeX, SizeY >, <a href="#">69</a>
beta_ KalmanFilters::CKalmanUKF< SizeX, SizeY >, <a href="#">106</a>	CKalmanSRECKFB KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >, <a href="#">72</a>
	CKalmanSREKF

- KalmanFilters::CKalmanSREKF< SizeX, SizeY >, [76](#)  
 CKalmanSREUKF  
 KalmanFilters::CKalmanSREUKF< SizeX, SizeY >, [80](#)  
 CKalmanSREUKFB  
 KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >, [84](#), [85](#)  
 CKalmanSRUKF  
 KalmanFilters::CKalmanSRUKF< SizeX, SizeY >, [89](#)  
 CKalmanSRUKFB  
 KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >, [96](#)  
 CKalmanUKF  
 KalmanFilters::CKalmanUKF< SizeX, SizeY >, [101](#), [102](#)  
 Correction  
 KalmanFilters::CKalmanCKF< SizeX, SizeY >, [22](#)  
 KalmanFilters::CKalmanECKF< SizeX, SizeY >, [30](#)  
 KalmanFilters::CKalmanEKF< SizeX, SizeY >, [34](#)  
 KalmanFilters::CKalmanEUKF< SizeX, SizeY >, [42](#)  
 KalmanFilters::CKalmanLKF< SizeX, SizeY >, [49](#)  
 KalmanFilters::CKalmanSRCKF< SizeX, SizeY >, [62](#)  
 KalmanFilters::CKalmanSRCKFB< SizeX, SizeY >, [67](#)  
 KalmanFilters::CKalmanSRECKF< SizeX, SizeY >, [69](#)  
 KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >, [73](#)  
 KalmanFilters::CKalmanSREKF< SizeX, SizeY >, [77](#)  
 KalmanFilters::CKalmanSREUKF< SizeX, SizeY >, [81](#)  
 KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >, [85](#)  
 KalmanFilters::CKalmanSRUKF< SizeX, SizeY >, [90](#)  
 KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >, [97](#)  
 KalmanFilters::CKalmanUKF< SizeX, SizeY >, [102](#)  
 CorrectionCKF  
 KalmanFilters::CKalmanCKF< SizeX, SizeY >, [22](#)  
 CorrectionSRCKF  
 KalmanFilters::CKalmanSRCKF< SizeX, SizeY >, [62](#)  
 CorrectionSRUKF  
 KalmanFilters::CKalmanSRUKF< SizeX, SizeY >, [90](#)  
 CorrectionUKF  
 KalmanFilters::CKalmanUKF< SizeX, SizeY >, [102](#)  
 createSignMatrices  
 KalmanFilters::CKalmanSREUKF< SizeX, SizeY >, [81](#)  
 KalmanFilters::CKalmanSRUKF< SizeX, SizeY >, [90](#)  
 createSignMatricesBlock  
 KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >, [85](#)  
 KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >, [97](#)  
 DeltaY\_  
 KalmanFilters::CKalmanLKF< SizeX, SizeY >, [56](#)  
 deltaY\_isSet  
 KalmanFilters::CKalmanLKF< SizeX, SizeY >, [57](#)  
 dXcal\_  
 KalmanFilters::CKalmanCKF< SizeX, SizeY >, [25](#)  
 KalmanFilters::CKalmanUKF< SizeX, SizeY >, [106](#)  
 dYcal\_  
 KalmanFilters::CKalmanCKF< SizeX, SizeY >, [25](#)  
 KalmanFilters::CKalmanUKF< SizeX, SizeY >, [106](#)  
 F\_  
 KalmanFilters::CKalmanLKF< SizeX, SizeY >, [57](#)  
 fixMatrixMainDiagonalSymmetry  
 KalmanFilters::CKalmanLKF< SizeX, SizeY >, [49](#)  
 gamma\_  
 KalmanFilters::CKalmanCKF< SizeX, SizeY >, [25](#)  
 KalmanFilters::CKalmanUKF< SizeX, SizeY >, [106](#)  
 GetDeltaY  
 KalmanFilters::CKalmanLKF< SizeX, SizeY >, [49](#)  
 GetEstimatedCovarianceMatrixP  
 KalmanFilters::CKalmanLKF< SizeX, SizeY >, [49](#)  
 GetEstimatedVectorX  
 KalmanFilters::CKalmanLKF< SizeX, SizeY >, [49](#)  
 GetEstimatedVectorY  
 KalmanFilters::CKalmanLKF< SizeX, SizeY >, [50](#)  
 GetInnovationCovarianceMatrixS  
 KalmanFilters::CKalmanLKF< SizeX, SizeY >, [50](#)  
 GetKalmanGainMatrixK

- KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 50
- GetSizeX
  - KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 50
- GetSizeY
  - KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 50
- GetVersion
  - KalmanFilters, 18
- H\_
  - KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 57
- I\_
  - KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 57
- J\_
  - KalmanFilters::CKalmanSREUKF< SizeX,  
SizeY >, 82
  - KalmanFilters::CKalmanSRUKF< SizeX,  
SizeY >, 93
- Jcorrect\_
  - KalmanFilters::CKalmanSREUKF< SizeX,  
SizeY >, 83
  - KalmanFilters::CKalmanSRUKF< SizeX,  
SizeY >, 93
- JcorrectBlock\_
  - KalmanFilters::CKalmanSREUKFB< SizeX,  
SizeY >, 86
  - KalmanFilters::CKalmanSRUKFB< SizeX,  
SizeY >, 98
- Jpredict\_
  - KalmanFilters::CKalmanSRUKF< SizeX,  
SizeY >, 93
- K\_
  - KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 57
- k\_sigma\_points\_
  - KalmanFilters::CKalmanCKF< SizeX, SizeY  
>, 25
  - KalmanFilters::CKalmanUKF< SizeX, SizeY  
>, 106
- Kalman\_filters, 15
- KalmanFilters, 17
  - GetVersion, 18
- KalmanFilters::CKalmanCKF< SizeX, SizeY >, 19
  - ~CKalmanCKF, 22
  - CKalmanCKF, 21, 22
  - Correction, 22
  - CorrectionCKF, 22
  - dXcal\_, 25
  - dYcal\_, 25
  - gamma\_, 25
  - k\_sigma\_points\_, 25
  - operator=, 22, 23
- P\_xy\_, 25
- Prediction, 23
- PredictionCKF, 23
- SetObservationJacobianH, 24
- SetStateTransitionJacobianF, 24
- SetupDesignParametersCubatureBaseSet, 24
- SetWeightedSumMeasurementSigmas, 24
- SetWeightedSumStateSigmas, 24
- sqrt\_P\_chol\_, 25
- swap, 24
- weightedSumMeasurementSigmas\_, 25
- weightedSumStateSigmas\_, 26
- weights\_covariance\_, 26
- weights\_mean\_, 26
- x\_est\_sigma\_points\_, 26
- x\_pred\_sigma\_points\_, 26
- y\_pred\_sigma\_points\_, 26
- KalmanFilters::CKalmanECKF< SizeX, SizeY >, 26
  - ~CKalmanECKF, 30
  - CKalmanECKF, 29
  - Correction, 30
  - operator=, 30
  - Prediction, 31
  - SetObservationJacobianH, 31
  - SetStateTransitionJacobianF, 31
  - swap, 31
- KalmanFilters::CKalmanEKF< SizeX, SizeY >, 32
  - ~CKalmanEKF, 34
  - CKalmanEKF, 34
  - Correction, 34
  - observationJacobianH\_, 37
  - observationModel\_, 38
  - operator=, 35
  - Prediction, 35
  - PredictionEKF, 36
  - SetObservationJacobianH, 36
  - SetObservationMatrixH, 36
  - SetObservationModel, 36
  - SetStateTransitionJacobianF, 36
  - SetStateTransitionJacobianLinearF, 36
  - SetStateTransitionMatrixF, 37
  - SetStateTransitionModel, 37
  - stateTransitionJacobianF\_, 38
  - stateTransitionModel\_, 38
  - swap, 37
- KalmanFilters::CKalmanEUKF< SizeX, SizeY >, 39
  - ~CKalmanEUKF, 42
  - CKalmanEUKF, 41, 42
  - Correction, 42
  - operator=, 42
  - Prediction, 43
  - SetObservationJacobianH, 43
  - SetStateTransitionJacobianF, 43
  - swap, 43
- KalmanFilters::CKalmanLKF< SizeX, SizeY >, 44
  - ~CKalmanLKF, 48

- CalculateInnovationCovarianceS, 48
- checkBordersMeasurement\_, 56
- checkBordersStateAfterCorrection\_, 56
- checkBordersStateAfterPrediction\_, 56
- checkDeltaMeasurement\_, 56
- checkDeltaState\_, 56
- checkMatrixDiagPositive, 48
- CKalmanLKF, 47
- Correction, 49
- DeltaY\_, 56
- deltaY\_isSet, 57
- F\_, 57
- fixMatrixMainDiagonalSymmetry, 49
- GetDeltaY, 49
- GetEstimatedCovarianceMatrixP, 49
- GetEstimatedVectorX, 49
- GetEstimatedVectorY, 50
- GetInnovationCovarianceMatrixS, 50
- GetKalmanGainMatrixK, 50
- GetSizeX, 50
- GetSizeY, 50
- H\_, 57
- I\_, 57
- K\_, 57
- operator=, 50, 51
- P\_, 57
- Prediction, 51
- prediction\_isDone, 57
- Q\_, 57
- R\_, 58
- S\_, 58
- SetCheckBordersMeasurement, 51
- SetCheckBordersStateAfterCorrection, 52
- SetCheckBordersStateAfterPrediction, 52
- SetCheckDeltaMeasurement, 52
- SetCheckDeltaState, 52
- SetDeltaY, 52
- SetEstimateCovarianceMatrixP, 53
- SetEstimateCovarianceMatrixPdiag, 53
- SetEstimatedVectorX, 53
- SetEstimatedVectorY, 53
- SetMeasuredVectorY, 54
- SetObservationCovarianceMatrixR, 54
- SetObservationCovarianceMatrixRdiag, 54
- SetObservationMatrixH, 54
- SetProcessCovarianceMatrixQ, 55
- SetProcessCovarianceMatrixQdiag, 55
- SetStateTransitionJacobianLinearF, 55
- SetStateTransitionMatrixF, 55
- SizeX\_, 58
- SizeY\_, 58
- stateTransitionJacobianLinearF\_, 58
- swap, 56
- X\_est\_, 58
- X\_pred\_, 58
- Y\_est\_, 59
- Y\_msd\_, 59
- Y\_msd\_isSet, 59
- Y\_pred\_, 59
- KalmanFilters::CKalmanSRCKF< SizeX, SizeY >, 59
  - ~CKalmanSRCKF, 62
  - CKalmanSRCKF, 61
  - Correction, 62
  - CorrectionSRCKF, 62
  - operator=, 62, 63
  - Prediction, 63
  - SetObservationJacobianH, 63
  - SetStateTransitionJacobianF, 63
  - swap, 64
- KalmanFilters::CKalmanSRCKFB< SizeX, SizeY >, 64
  - CKalmanSRCKFB, 67
  - Correction, 67
- KalmanFilters::CKalmanSRECKF< SizeX, SizeY >, 67
  - ~CKalmanSRECKF, 69
  - CKalmanSRECKF, 69
  - Correction, 69
  - operator=, 70
  - Prediction, 70
  - swap, 70
- KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >, 71
  - ~CKalmanSRECKFB, 73
  - CKalmanSRECKFB, 72
  - Correction, 73
  - operator=, 73
  - swap, 74
- KalmanFilters::CKalmanSREKF< SizeX, SizeY >, 74
  - ~CKalmanSREKF, 76
  - CKalmanSREKF, 76
  - Correction, 77
  - operator=, 77
  - Prediction, 78
  - PredictionSREKF, 78
  - swap, 78
- KalmanFilters::CKalmanSREUKF< SizeX, SizeY >, 78
  - ~CKalmanSREUKF, 81
  - CKalmanSREUKF, 80
  - Correction, 81
  - createSignMatrices, 81
  - J\_, 82
  - Jcorrect\_, 83
  - operator=, 81, 82
  - Prediction, 82
  - swap, 82
- KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >, 83
  - ~CKalmanSREUKFB, 85
  - CKalmanSREUKFB, 84, 85
  - Correction, 85
  - createSignMatricesBlock, 85
  - JcorrectBlock\_, 86

- operator=, 85, 86
- swap, 86
- KalmanFilters::CKalmanSRUKF< SizeX, SizeY >, 87
  - ~CKalmanSRUKF, 89
  - CKalmanSRUKF, 89
  - Correction, 90
  - CorrectionSRUKF, 90
  - createSignMatrices, 90
  - J\_, 93
  - Jcorrect\_, 93
  - Jpredict\_, 93
  - negativeZeroCovWeight\_, 93
  - operator=, 90
  - Prediction, 91
  - SetObservationJacobianH, 91
  - SetStateTransitionJacobianF, 91
  - SetupDesignParametersCDKF, 91
  - SetupDesignParametersMeanSet, 92
  - SetupDesignParametersScaledSet, 92
  - swap, 92
- KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >, 93
  - ~CKalmanSRUKFB, 97
  - CKalmanSRUKFB, 96
  - Correction, 97
  - createSignMatricesBlock, 97
  - JcorrectBlock\_, 98
  - operator=, 97
  - swap, 98
- KalmanFilters::CKalmanUKF< SizeX, SizeY >, 98
  - ~CKalmanUKF, 102
  - alpha\_, 106
  - beta\_, 106
  - CKalmanUKF, 101, 102
  - Correction, 102
  - CorrectionUKF, 102
  - dXcal\_, 106
  - dYcal\_, 106
  - gamma\_, 106
  - k\_sigma\_points\_, 106
  - kappa\_, 106
  - lambda\_, 107
  - operator=, 102, 103
  - P\_xy\_, 107
  - Prediction, 103
  - PredictionUKF, 103
  - SetObservationJacobianH, 104
  - SetStateTransitionJacobianF, 104
  - SetupDesignParametersCDKF, 104
  - SetupDesignParametersMeanSet, 104
  - SetupDesignParametersScaledSet, 104
  - SetWeightedSumMeasurementSigmas, 105
  - SetWeightedSumStateSigmas, 105
  - sqrt\_P\_chol\_, 107
  - swap, 105
  - w0\_, 107
  - weightedSumMeasurementSigmas\_, 107
  - weightedSumStateSigmas\_, 107
  - weights\_covariance\_, 107
  - weights\_mean\_, 107
  - x\_est\_sigma\_points\_, 108
  - x\_pred\_sigma\_points\_, 108
  - y\_pred\_sigma\_points\_, 108
- kappa\_
  - KalmanFilters::CKalmanUKF< SizeX, SizeY >, 106
- lambda\_
  - KalmanFilters::CKalmanUKF< SizeX, SizeY >, 107
- negativeZeroCovWeight\_
  - KalmanFilters::CKalmanSRUKF< SizeX, SizeY >, 93
- observationJacobianH\_
  - KalmanFilters::CKalmanEKF< SizeX, SizeY >, 37
- observationModel\_
  - KalmanFilters::CKalmanEKF< SizeX, SizeY >, 38
- operator=
  - KalmanFilters::CKalmanCKF< SizeX, SizeY >, 22, 23
  - KalmanFilters::CKalmanECKF< SizeX, SizeY >, 30
  - KalmanFilters::CKalmanEKF< SizeX, SizeY >, 35
  - KalmanFilters::CKalmanEUKF< SizeX, SizeY >, 42
  - KalmanFilters::CKalmanLKF< SizeX, SizeY >, 50, 51
  - KalmanFilters::CKalmanSRCKF< SizeX, SizeY >, 62, 63
  - KalmanFilters::CKalmanSRECKF< SizeX, SizeY >, 70
  - KalmanFilters::CKalmanSRECKFB< SizeX, SizeY >, 73
  - KalmanFilters::CKalmanSREKF< SizeX, SizeY >, 77
  - KalmanFilters::CKalmanSREUKF< SizeX, SizeY >, 81, 82
  - KalmanFilters::CKalmanSREUKFB< SizeX, SizeY >, 85, 86
  - KalmanFilters::CKalmanSRUKF< SizeX, SizeY >, 90
  - KalmanFilters::CKalmanSRUKFB< SizeX, SizeY >, 97
  - KalmanFilters::CKalmanUKF< SizeX, SizeY >, 102, 103
- P\_
  - KalmanFilters::CKalmanLKF< SizeX, SizeY >, 57
- P\_xy\_
  - KalmanFilters::CKalmanLKF< SizeX, SizeY >, 57



- KalmanFilters::CKalmanCKF< SizeX, SizeY  
 >, [25](#)  
 KalmanFilters::CKalmanUKF< SizeX, SizeY  
 >, [107](#)  
 Prediction  
 KalmanFilters::CKalmanCKF< SizeX, SizeY  
 >, [23](#)  
 KalmanFilters::CKalmanECKF< SizeX,  
 SizeY >, [31](#)  
 KalmanFilters::CKalmanEKF< SizeX, SizeY  
 >, [35](#)  
 KalmanFilters::CKalmanEUKF< SizeX,  
 SizeY >, [43](#)  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [51](#)  
 KalmanFilters::CKalmanSRCKF< SizeX,  
 SizeY >, [63](#)  
 KalmanFilters::CKalmanSRECKF< SizeX,  
 SizeY >, [70](#)  
 KalmanFilters::CKalmanSREKF< SizeX,  
 SizeY >, [78](#)  
 KalmanFilters::CKalmanSREUKF< SizeX,  
 SizeY >, [82](#)  
 KalmanFilters::CKalmanSRUKF< SizeX,  
 SizeY >, [91](#)  
 KalmanFilters::CKalmanUKF< SizeX, SizeY  
 >, [103](#)  
 prediction\_isDone  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [57](#)  
 PredictionCKF  
 KalmanFilters::CKalmanCKF< SizeX, SizeY  
 >, [23](#)  
 PredictionEKF  
 KalmanFilters::CKalmanEKF< SizeX, SizeY  
 >, [36](#)  
 PredictionSREKF  
 KalmanFilters::CKalmanSREKF< SizeX,  
 SizeY >, [78](#)  
 PredictionUKF  
 KalmanFilters::CKalmanUKF< SizeX, SizeY  
 >, [103](#)  
 Q\_  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [57](#)  
 R\_  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [58](#)  
 S\_  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [58](#)  
 SetCheckBordersMeasurement  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [51](#)  
 SetCheckBordersStateAfterCorrection  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [52](#)  
 SetCheckBordersStateAfterPrediction  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [52](#)  
 SetCheckDeltaMeasurement  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [52](#)  
 SetCheckDeltaState  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [52](#)  
 SetDeltaY  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [52](#)  
 SetEstimateCovarianceMatrixP  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [53](#)  
 SetEstimateCovarianceMatrixPdiag  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [53](#)  
 SetEstimatedVectorX  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [53](#)  
 SetEstimatedVectorY  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [53](#)  
 SetMeasuredVectorY  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [54](#)  
 SetObservationCovarianceMatrixR  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [54](#)  
 SetObservationCovarianceMatrixRdiag  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [54](#)  
 SetObservationJacobianH  
 KalmanFilters::CKalmanCKF< SizeX, SizeY  
 >, [24](#)  
 KalmanFilters::CKalmanECKF< SizeX,  
 SizeY >, [31](#)  
 KalmanFilters::CKalmanEKF< SizeX, SizeY  
 >, [36](#)  
 KalmanFilters::CKalmanEUKF< SizeX,  
 SizeY >, [43](#)  
 KalmanFilters::CKalmanSRCKF< SizeX,  
 SizeY >, [63](#)  
 KalmanFilters::CKalmanSRUKF< SizeX,  
 SizeY >, [91](#)  
 KalmanFilters::CKalmanUKF< SizeX, SizeY  
 >, [104](#)  
 SetObservationMatrixH  
 KalmanFilters::CKalmanEKF< SizeX, SizeY  
 >, [36](#)  
 KalmanFilters::CKalmanLKF< SizeX, SizeY  
 >, [54](#)  
 SetObservationModel  
 KalmanFilters::CKalmanEKF< SizeX, SizeY  
 >, [36](#)



- SetProcessCovarianceMatrixQ  
KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 55
- SetProcessCovarianceMatrixQdiag  
KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 55
- SetStateTransitionJacobianF  
KalmanFilters::CKalmanCKF< SizeX, SizeY  
>, 24  
KalmanFilters::CKalmanECKF< SizeX,  
SizeY >, 31  
KalmanFilters::CKalmanEKF< SizeX, SizeY  
>, 36  
KalmanFilters::CKalmanEUKF< SizeX,  
SizeY >, 43  
KalmanFilters::CKalmanSRCKF< SizeX,  
SizeY >, 63  
KalmanFilters::CKalmanSRUKF< SizeX,  
SizeY >, 91  
KalmanFilters::CKalmanUKF< SizeX, SizeY  
>, 104
- SetStateTransitionJacobianLinearF  
KalmanFilters::CKalmanEKF< SizeX, SizeY  
>, 36  
KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 55
- SetStateTransitionMatrixF  
KalmanFilters::CKalmanEKF< SizeX, SizeY  
>, 37  
KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 55
- SetStateTransitionModel  
KalmanFilters::CKalmanEKF< SizeX, SizeY  
>, 37
- SetupDesignParametersCDKF  
KalmanFilters::CKalmanSRUKF< SizeX,  
SizeY >, 91  
KalmanFilters::CKalmanUKF< SizeX, SizeY  
>, 104
- SetupDesignParametersCubatureBaseSet  
KalmanFilters::CKalmanCKF< SizeX, SizeY  
>, 24
- SetupDesignParametersMeanSet  
KalmanFilters::CKalmanSRUKF< SizeX,  
SizeY >, 92  
KalmanFilters::CKalmanUKF< SizeX, SizeY  
>, 104
- SetupDesignParametersScaledSet  
KalmanFilters::CKalmanSRUKF< SizeX,  
SizeY >, 92  
KalmanFilters::CKalmanUKF< SizeX, SizeY  
>, 104
- SetWeightedSumMeasurementSigmas  
KalmanFilters::CKalmanCKF< SizeX, SizeY  
>, 24  
KalmanFilters::CKalmanUKF< SizeX, SizeY  
>, 105
- SetWeightedSumStateSigmas  
KalmanFilters::CKalmanCKF< SizeX, SizeY  
>, 24  
KalmanFilters::CKalmanUKF< SizeX, SizeY  
>, 105
- KalmanFilters::CKalmanCKF< SizeX, SizeY  
>, 24  
KalmanFilters::CKalmanUKF< SizeX, SizeY  
>, 105  
SizeX\_  
KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 58  
SizeY\_  
KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 58  
sqrt\_P\_chol\_  
KalmanFilters::CKalmanCKF< SizeX, SizeY  
>, 25  
KalmanFilters::CKalmanUKF< SizeX, SizeY  
>, 107  
stateTransitionJacobianF\_  
KalmanFilters::CKalmanEKF< SizeX, SizeY  
>, 38  
stateTransitionJacobianLinearF\_  
KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 58  
stateTransitionModel\_  
KalmanFilters::CKalmanEKF< SizeX, SizeY  
>, 38  
swap  
KalmanFilters::CKalmanCKF< SizeX, SizeY  
>, 24  
KalmanFilters::CKalmanECKF< SizeX,  
SizeY >, 31  
KalmanFilters::CKalmanEKF< SizeX, SizeY  
>, 37  
KalmanFilters::CKalmanEUKF< SizeX,  
SizeY >, 43  
KalmanFilters::CKalmanLKF< SizeX, SizeY  
>, 56  
KalmanFilters::CKalmanSRCKF< SizeX,  
SizeY >, 64  
KalmanFilters::CKalmanSRECKF< SizeX,  
SizeY >, 70  
KalmanFilters::CKalmanSRECKFB< SizeX,  
SizeY >, 74  
KalmanFilters::CKalmanSREKF< SizeX,  
SizeY >, 78  
KalmanFilters::CKalmanSREUKF< SizeX,  
SizeY >, 82  
KalmanFilters::CKalmanSREUKFB< SizeX,  
SizeY >, 86  
KalmanFilters::CKalmanSRUKF< SizeX,  
SizeY >, 92  
KalmanFilters::CKalmanSRUKFB< SizeX,  
SizeY >, 98  
KalmanFilters::CKalmanUKF< SizeX, SizeY  
>, 105  
w0\_  
KalmanFilters::CKalmanUKF< SizeX, SizeY  
>, 107  
weightedSumMeasurementSigmas\_

KalmanFilters::CKalmanCKF< SizeX, SizeY  
     >, [25](#)  
 KalmanFilters::CKalmanUKF< SizeX, SizeY  
     >, [107](#)  
 weightedSumStateSigmas\_  
     KalmanFilters::CKalmanCKF< SizeX, SizeY  
         >, [26](#)  
     KalmanFilters::CKalmanUKF< SizeX, SizeY  
         >, [107](#)  
 weights\_covariance\_  
     KalmanFilters::CKalmanCKF< SizeX, SizeY  
         >, [26](#)  
     KalmanFilters::CKalmanUKF< SizeX, SizeY  
         >, [107](#)  
 weights\_mean\_  
     KalmanFilters::CKalmanCKF< SizeX, SizeY  
         >, [26](#)  
     KalmanFilters::CKalmanUKF< SizeX, SizeY  
         >, [107](#)  
  
 X\_est\_  
     KalmanFilters::CKalmanLKF< SizeX, SizeY  
         >, [58](#)  
 x\_est\_sigma\_points\_  
     KalmanFilters::CKalmanCKF< SizeX, SizeY  
         >, [26](#)  
     KalmanFilters::CKalmanUKF< SizeX, SizeY  
         >, [108](#)  
 X\_pred\_  
     KalmanFilters::CKalmanLKF< SizeX, SizeY  
         >, [58](#)  
 x\_pred\_sigma\_points\_  
     KalmanFilters::CKalmanCKF< SizeX, SizeY  
         >, [26](#)  
     KalmanFilters::CKalmanUKF< SizeX, SizeY  
         >, [108](#)  
  
 Y\_est\_  
     KalmanFilters::CKalmanLKF< SizeX, SizeY  
         >, [59](#)  
 Y\_msd\_  
     KalmanFilters::CKalmanLKF< SizeX, SizeY  
         >, [59](#)  
 Y\_msd\_isSet  
     KalmanFilters::CKalmanLKF< SizeX, SizeY  
         >, [59](#)  
 Y\_pred\_  
     KalmanFilters::CKalmanLKF< SizeX, SizeY  
         >, [59](#)  
 y\_pred\_sigma\_points\_  
     KalmanFilters::CKalmanCKF< SizeX, SizeY  
         >, [26](#)  
     KalmanFilters::CKalmanUKF< SizeX, SizeY  
         >, [108](#)