

# Отчёт по практическому заданию по ДГСП

Лазар Владислав Игоревич, 416 группа

11 апреля 2025 г.

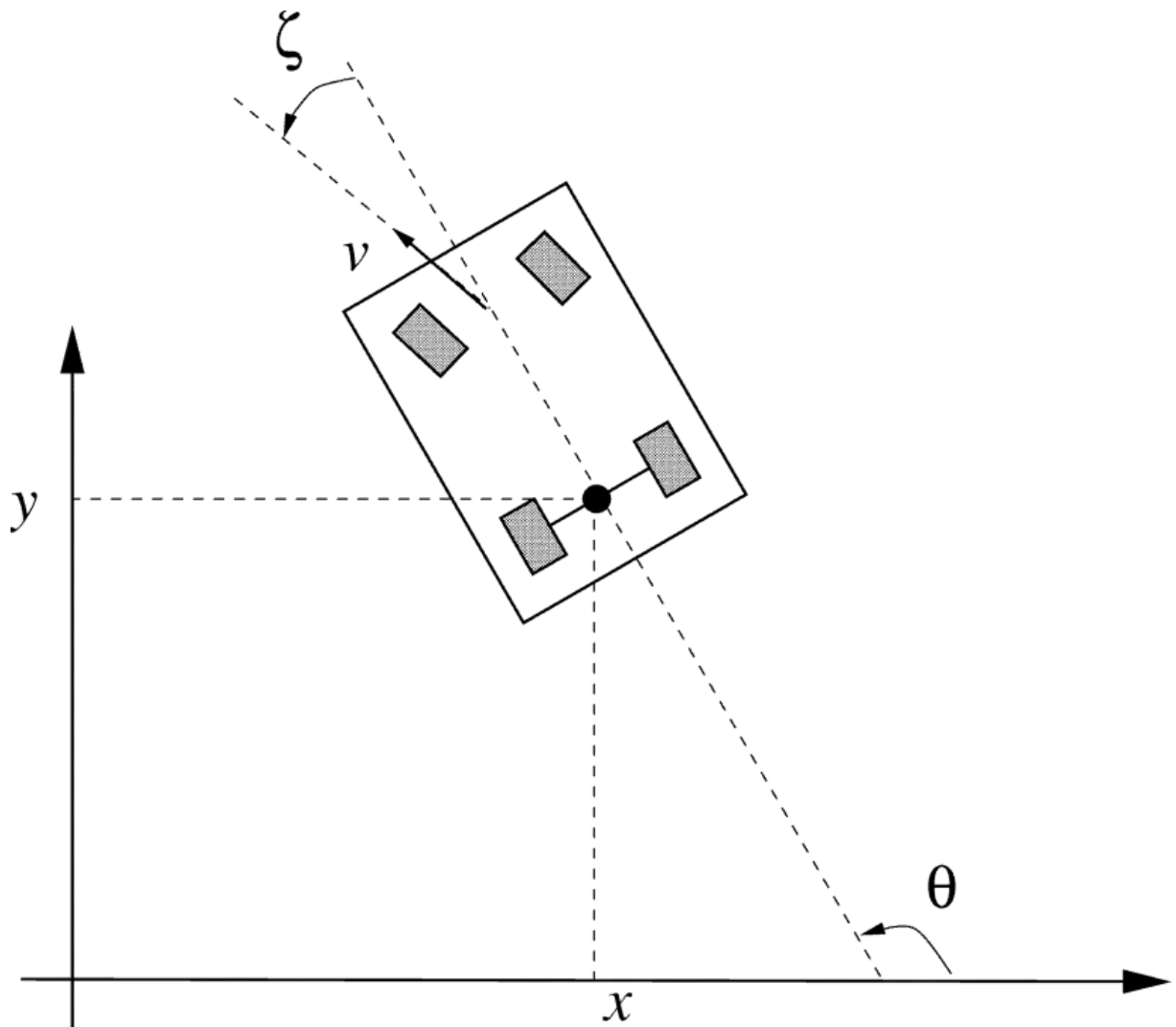
# Содержание

<b>1</b>	<b>Теоретическая часть</b>	<b>3</b>
1.1	Исследуемое явление . . . . .	3
1.2	Система наблюдения . . . . .	4
<b>2</b>	<b>Практическая часть</b>	<b>6</b>
2.1	Значения параметров системы . . . . .	6
2.2	Сравнение работы алгоритмов . . . . .	6
2.2.1	Оценки траектории . . . . .	7
2.2.2	Ошибки оценивания $x_1$ . . . . .	11
2.2.3	Ошибки оценивания $x_2$ . . . . .	15
2.2.4	Ошибки оценивания $\theta$ . . . . .	19
2.2.5	Ошибки оценивания $\phi$ . . . . .	23
2.2.6	Графики ошибок для нерасходящихся траекторий . . . . .	27
2.2.7	Графики ошибок для всех траекторий . . . . .	31
2.2.8	Численные результаты . . . . .	35
<b>3</b>	<b>Выводы по расчётам</b>	<b>36</b>

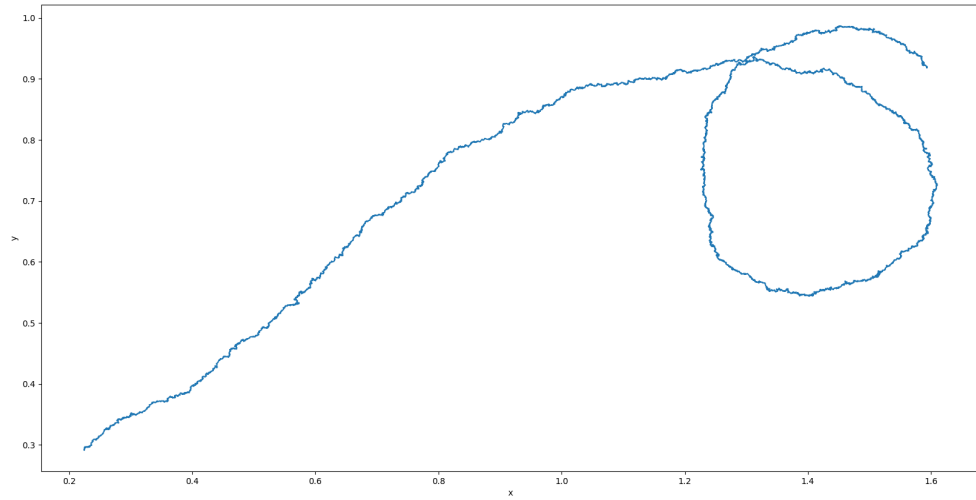
# 1 Теоретическая часть

## 1.1 Исследуемое явление

В данной работе рассматривается задача наблюдения за перемещающейся по плоскости тележкой. Состоянием тележки является её положение на плоскости ( $x$ ,  $y$ , ориентация в пространстве) и угол поворота колёс. На плоскости в начале координат установлен локатор, измеряющий расстояние до тележки и угол направления на неё.



Далее будет описана используемая математическая модель. Типичная траектория получаемая с её помощью выглядит следующим образом:



## 1.2 Система наблюдения

Математическая модель описывается системой с дискретным временем:

$$\begin{cases} x_{1,k} = x_{1,k-1} + T \cos(\theta_{k-1}) \cos(\phi_{k-1}) u_{1,k-1} \\ x_{2,k} = x_{2,k-1} + T \sin(\theta_{k-1}) \cos(\phi_{k-1}) u_{1,k-1} \\ \theta_k = \theta_{k-1} + T \sin(\phi_{k-1}) \frac{u_{1,k-1}}{l} + \omega_{3,k} \\ \phi_k = \phi_{k-1} + T u_{2,k-1} + \omega_{4,k} \end{cases}$$

где

- $x_1$  - координата объекта по оси x
- $x_2$  - координата объекта по оси y
- $\theta$  - угол между направлением движения тележки и положительной полуосью Oх (ориентация на плоскости)
- $\phi$  - угол поворота колёс тележки относительно направления движения
- $T$  - параметр дискретизации системы по времени
- $l$  - расстояние между осями тележки
- $\omega$  - шум в модели динамики,  $\omega_k \sim \mathcal{N}(0, Q)$
- $u_1$  - управляемая линейная скорость
- $u_2$  - управляемая угловая скорость

Наблюдения определяются следующим образом:

$$\begin{cases} r_k = \sqrt{x_{1,k}^2 + x_{2,k}^2} + \nu_{1,k} \\ \alpha_k = \arctan(\frac{x_{2,k}}{x_{1,k}}) + \nu_{2,k} \end{cases}$$

где

- $r_k$  - расстояние от лоатора до цели
- $\alpha_k$  - угол направления от лоатора к цели
- $\nu$  - шум в наблюдениях,  $\nu \sim \mathcal{N}(0, R)$

## 2 Практическая часть

### 2.1 Значения параметров системы

В численных экспериментах установим следующие параметры системы:

$$l = 0.1, \quad u_1 = 3, \quad u_2 = 0$$

Шаги по времени:

$$\delta_1 = \delta_2 = 10^{-3}, \quad \delta_1 = 10^{-2}, \quad T_{max} = 1$$

Заметим, что равенство  $\delta_1$  и  $\delta_2$  следует из того, что система заранее дискретизирована.

Также для наблюдений вместо  $\arctan$  возьмём  $\arctan_2$ . Это обусловлено тем, что область действия  $\arctan$  -  $[-\frac{\pi}{2}; \frac{\pi}{2}]$ , а  $\arctan_2$  - весь тригонометрический круг. Благодаря этому мы сможем избежать ошибки наблюдений с неправильным определением полуплоскости.

Матрицы шумов возьмём следующими:

$$Q = \begin{bmatrix} 3 \cdot 10^{-7} & 0 & 0 & 0 \\ 0 & 3 \cdot 10^{-7} & 0 & 0 \\ 0 & 0 & 3 \cdot 10^{-5} & 0 \\ 0 & 0 & 0 & 3 \cdot 10^{-5} \end{bmatrix}$$

$$R = 5 \cdot 10^{-3} \cdot I_2$$

Начальное значение траектории моделируется следующим образом:

$$X_0 \sim \mathcal{N}(0, 0.3 \cdot I_4)$$

Все необходимые для фильтрации параметры были посчитаны аналитически во время работы программы с помощью соответствующих пакетов Python (`filterpy`, `sympy`). Реализации всех фильтров также взяты из соответствующих пакетов на Python. Также для достижения большей точности были использованы 128-битные числа с плавающей точкой (стандарт IEEE 754).

### 2.2 Сравнение работы алгоритмов

### 2.2.1 Оценки траектории

Рис. 1: Оценка  $x_1$

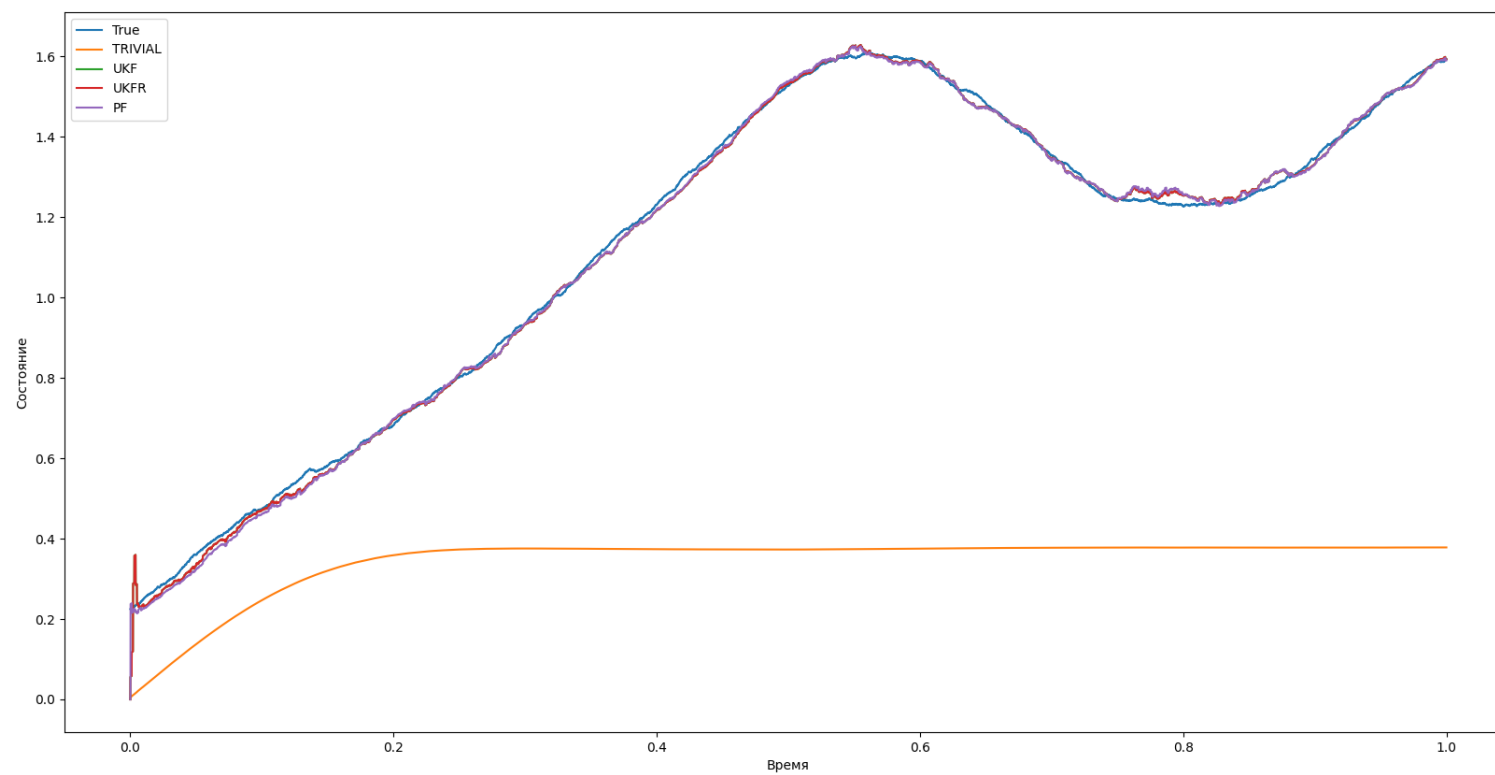


Рис. 2: Оценка  $x_2$

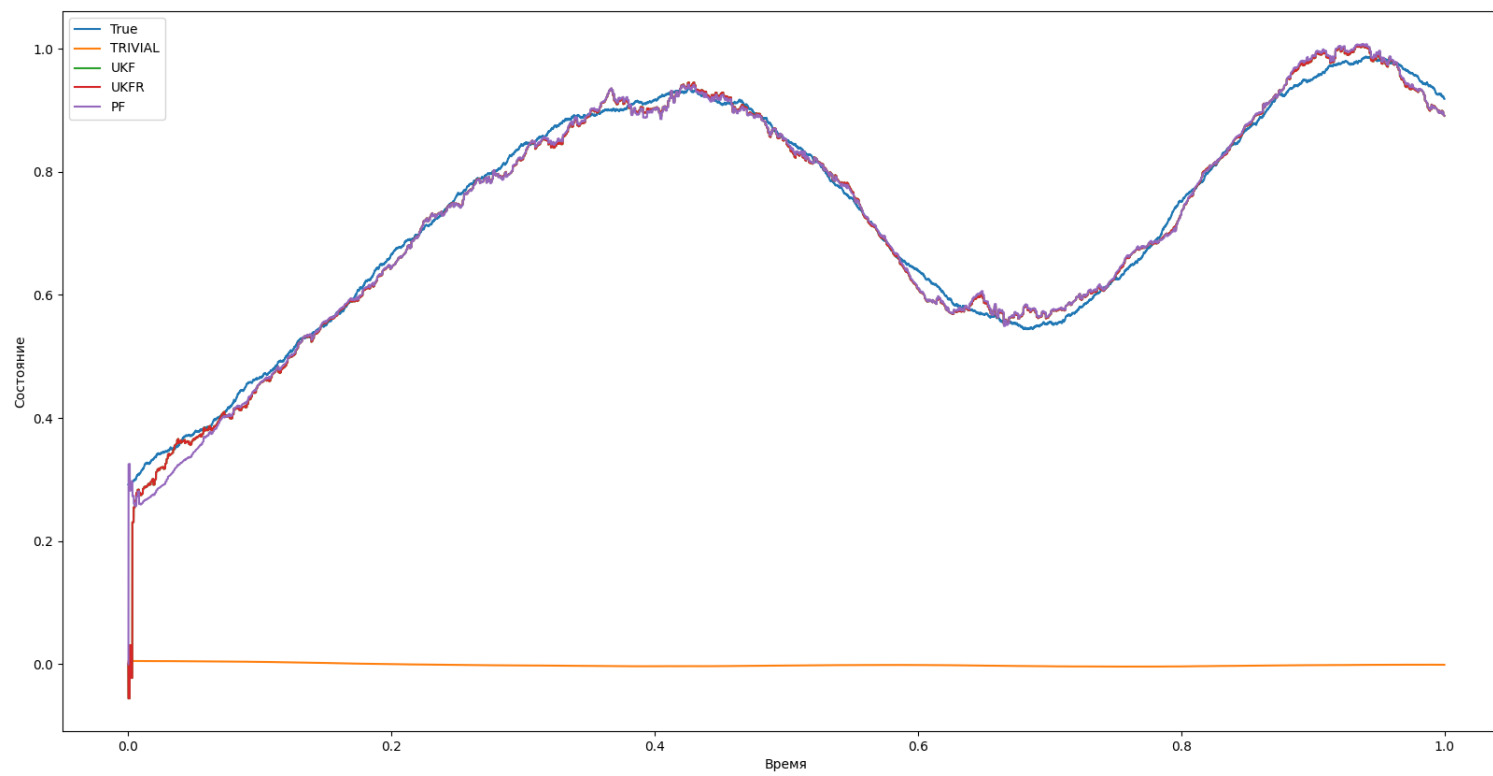




Рис. 3: Оценка  $\theta$

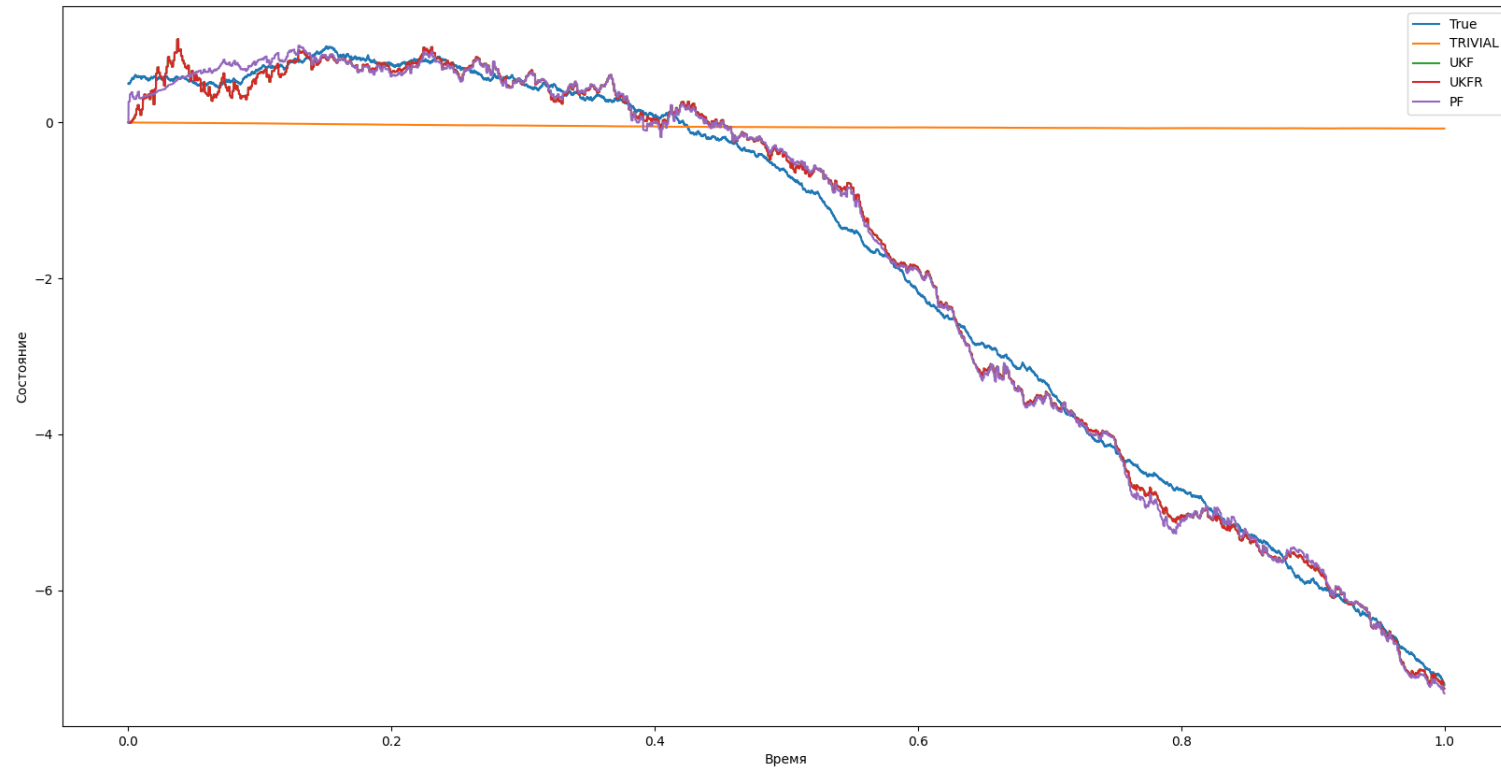
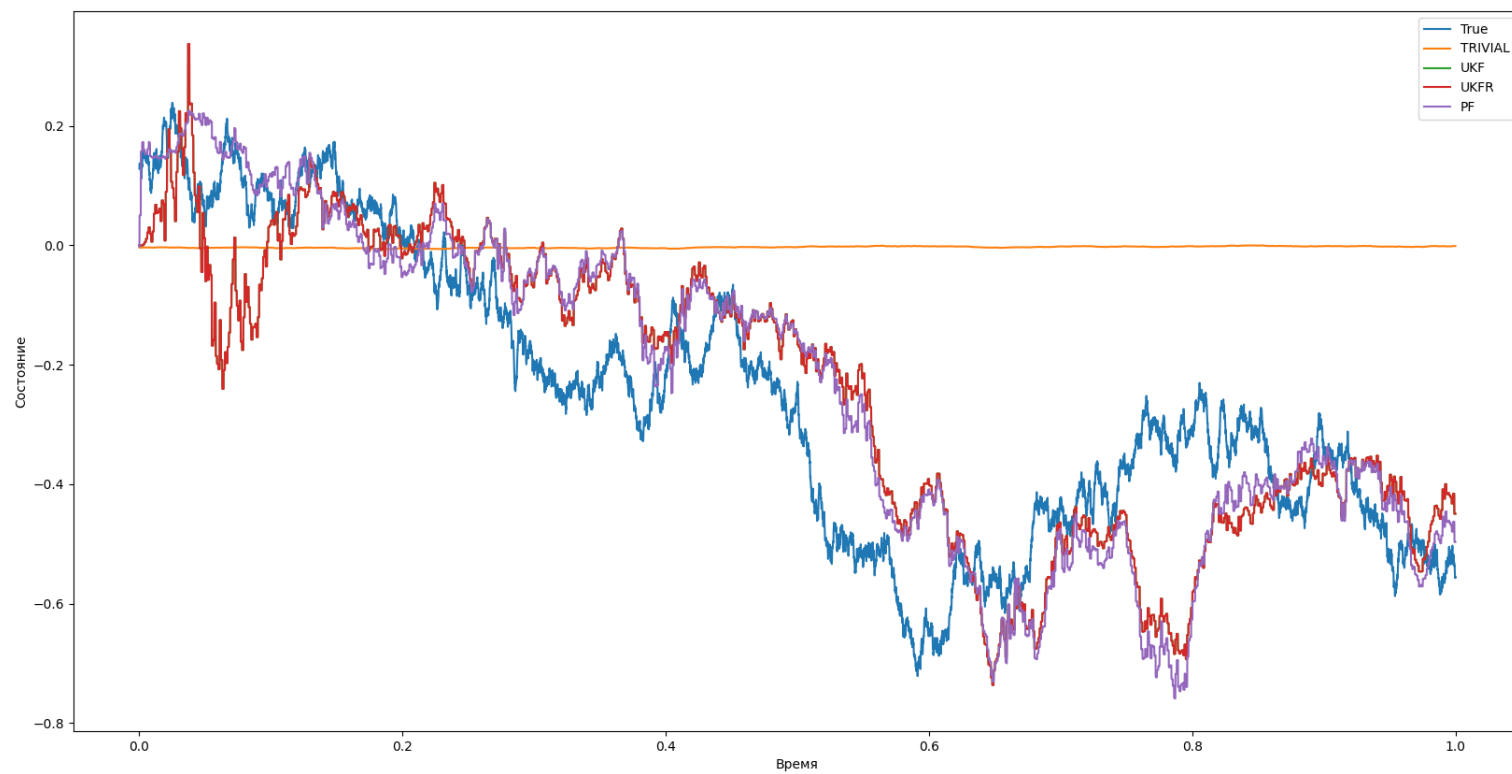
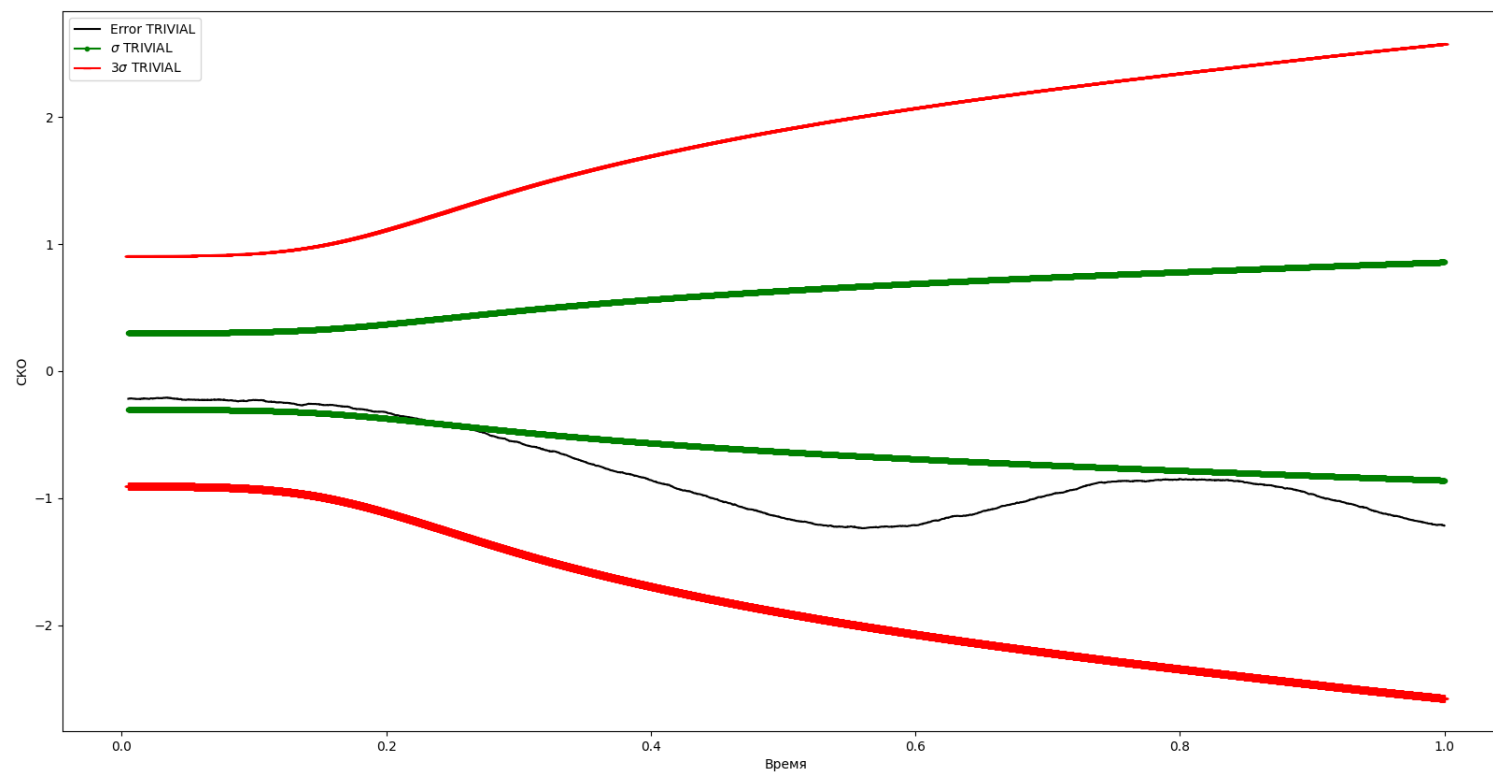
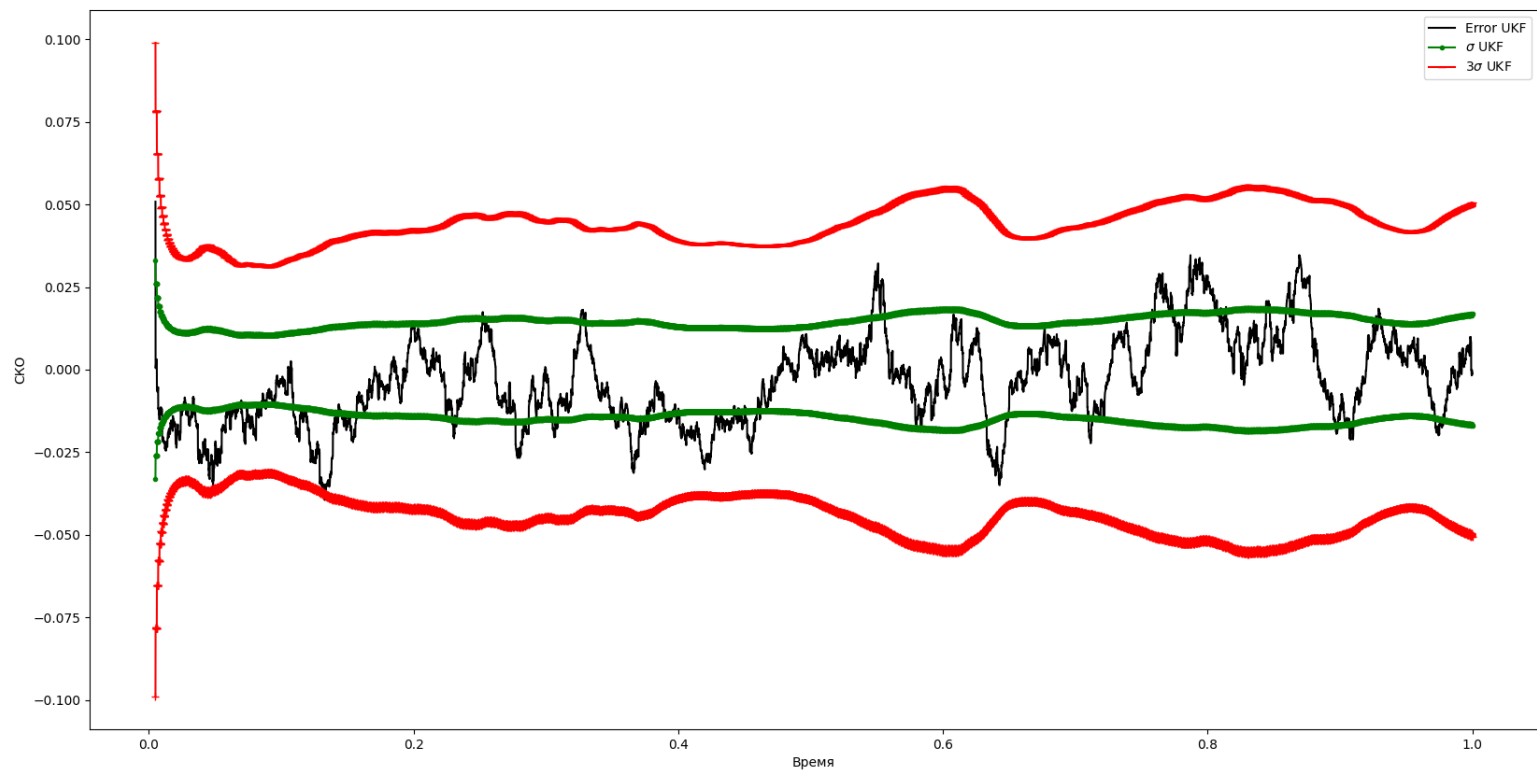


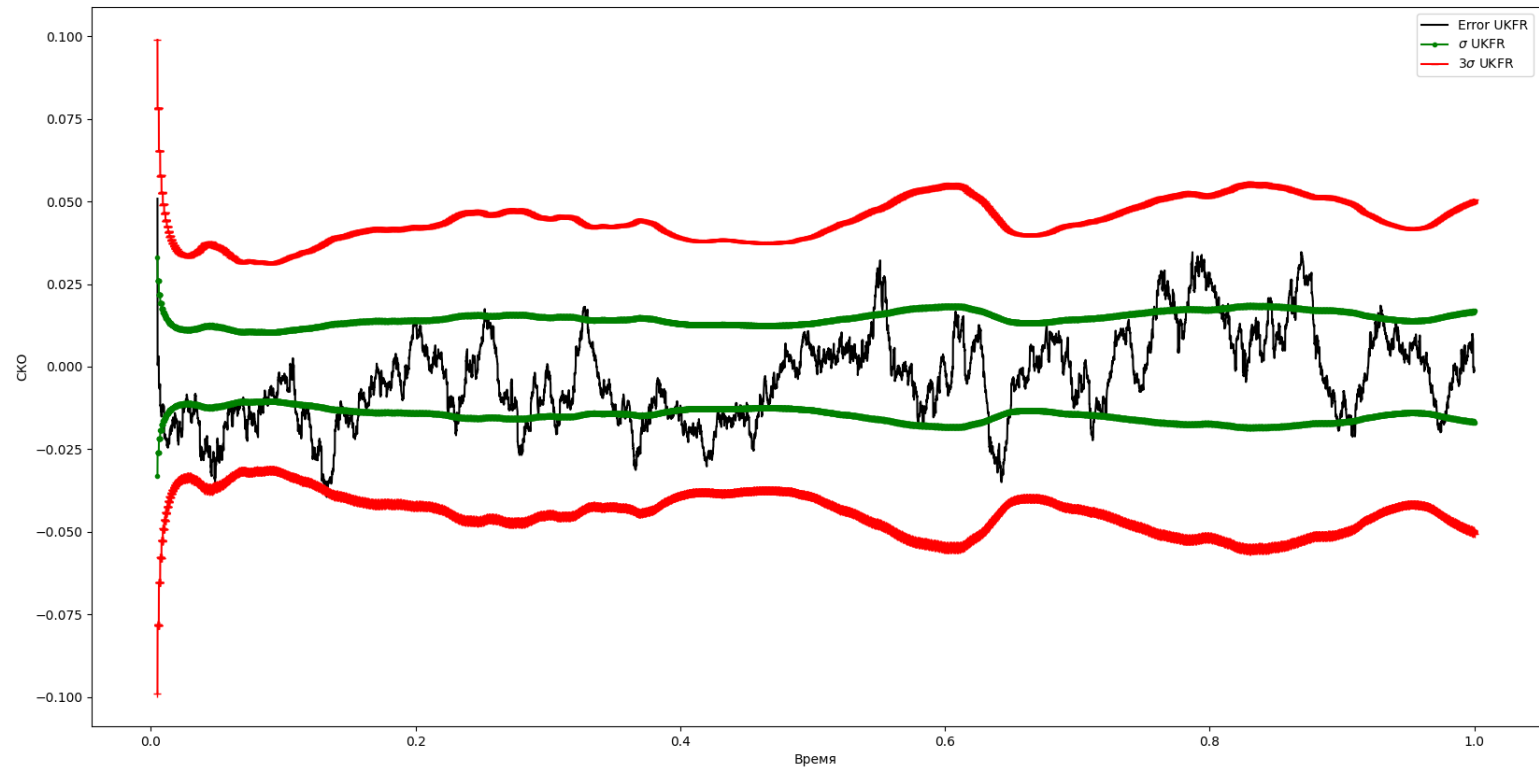
Рис. 4: Оценка  $\phi$

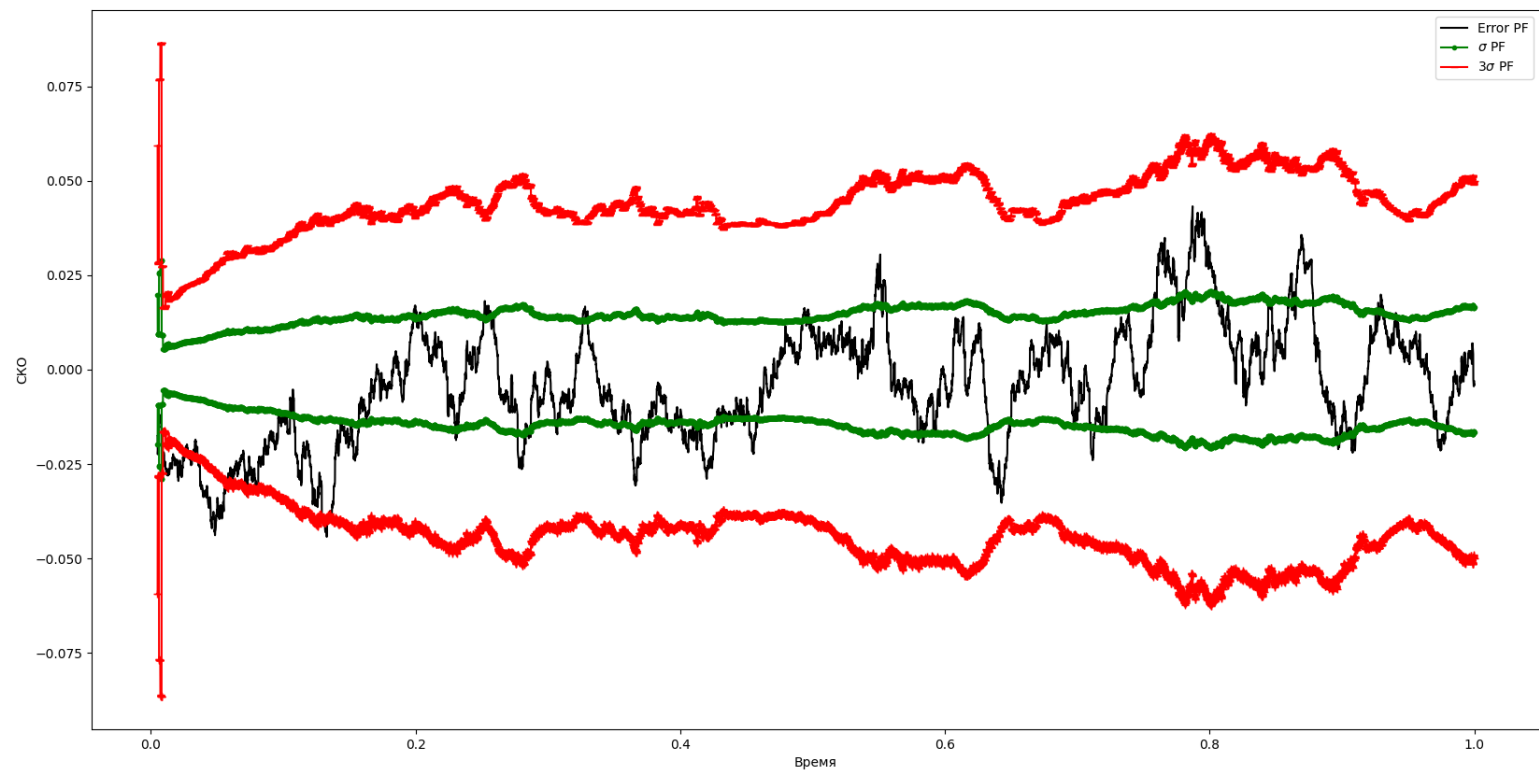


### 2.2.2 Ошибки оценивания $x_1$

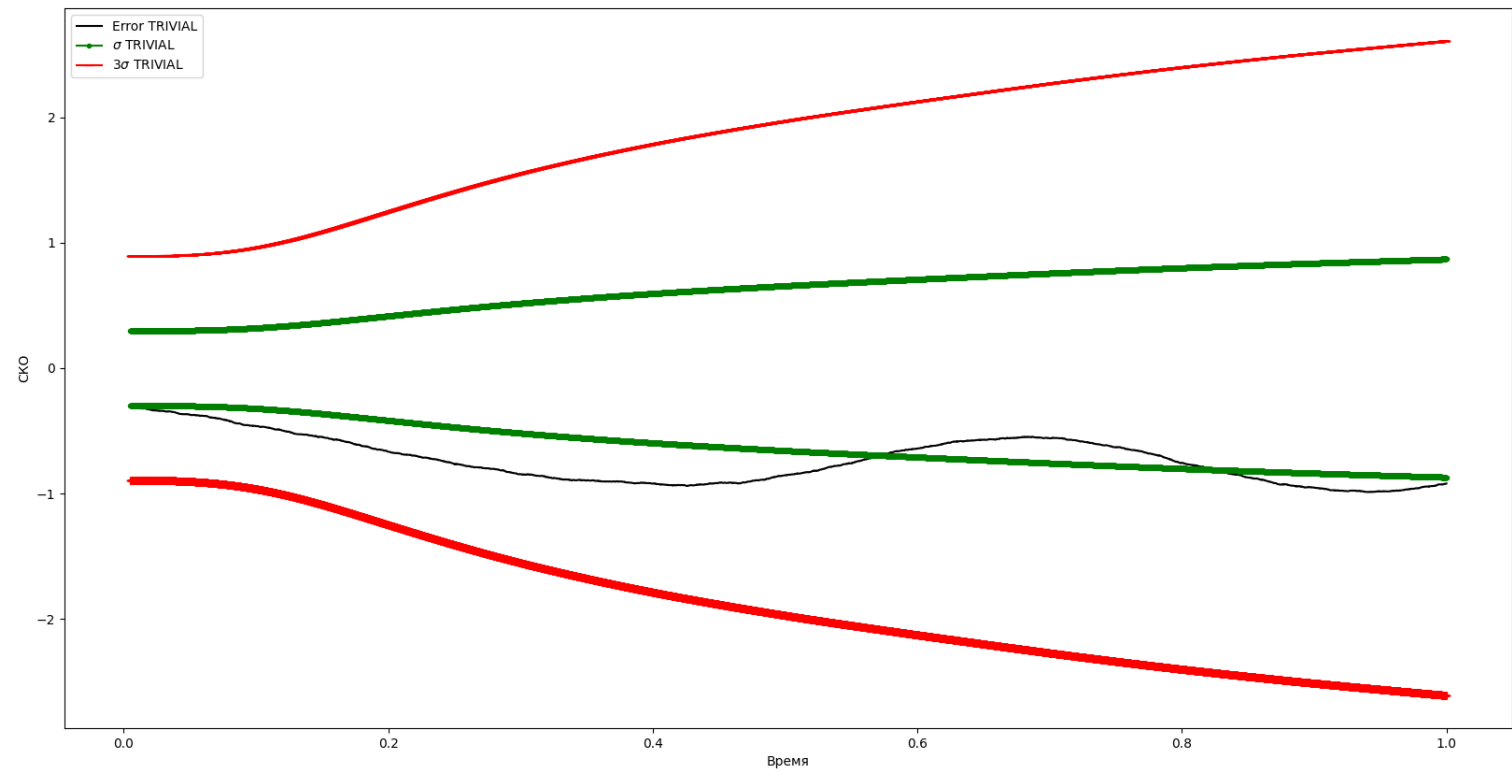


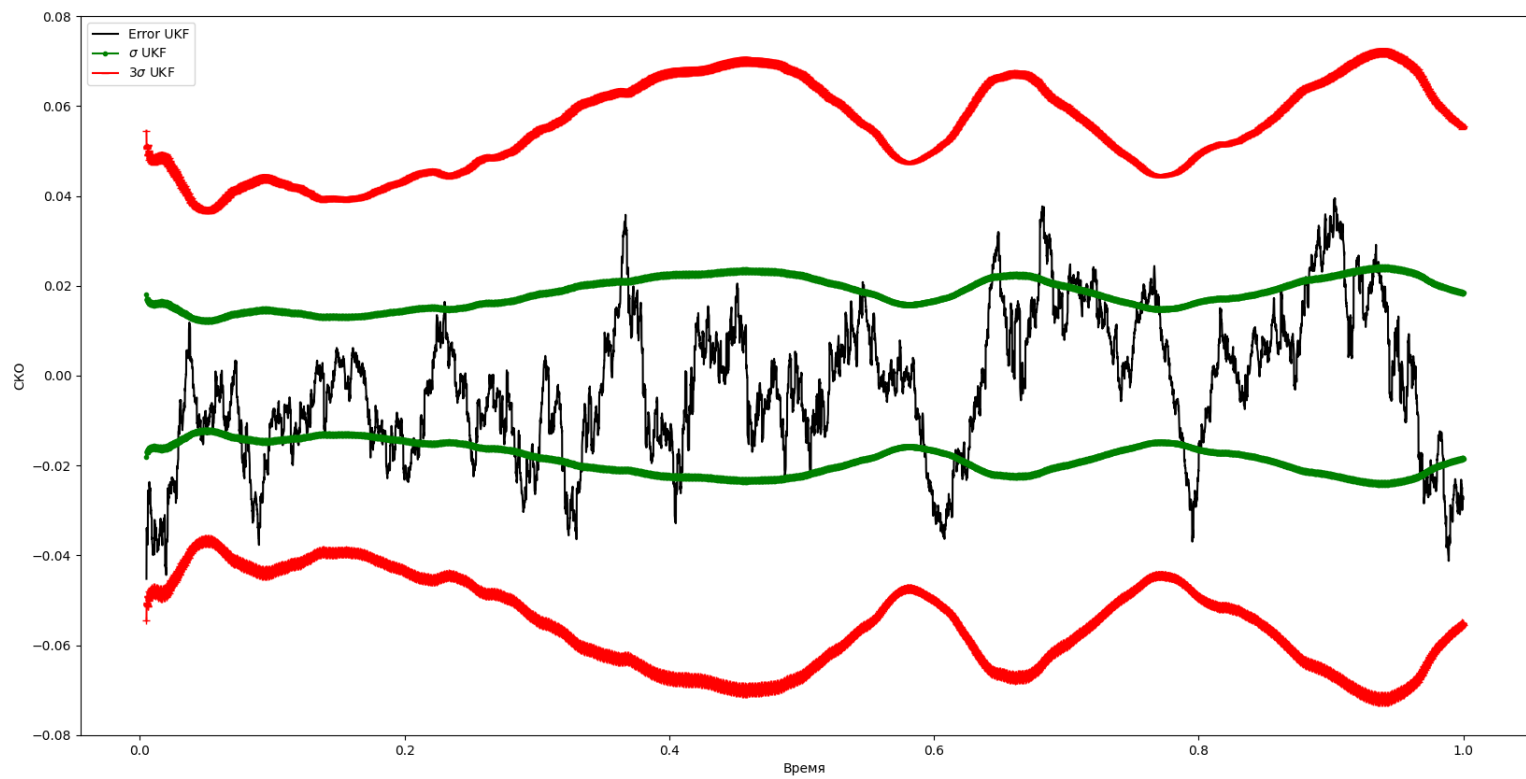




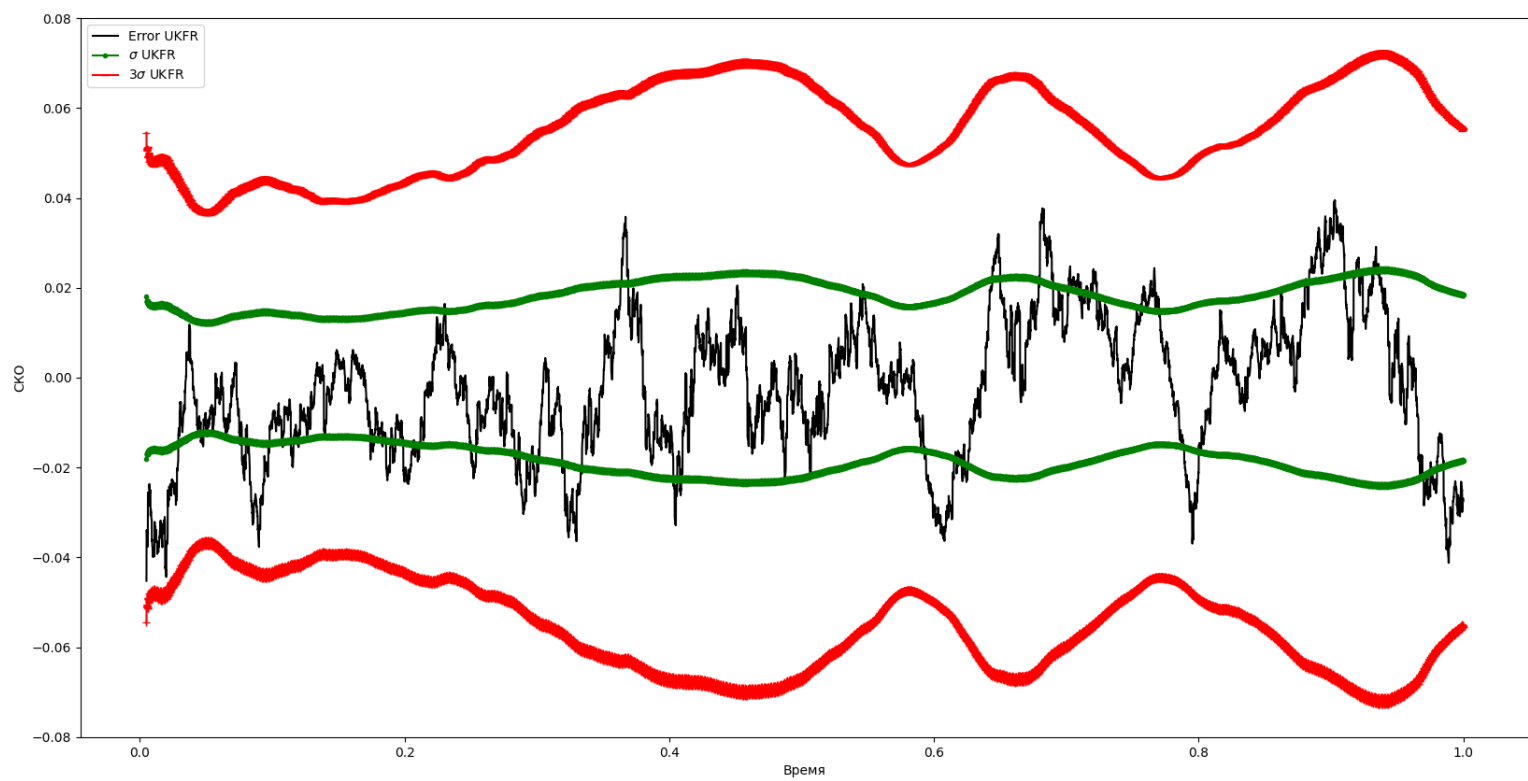


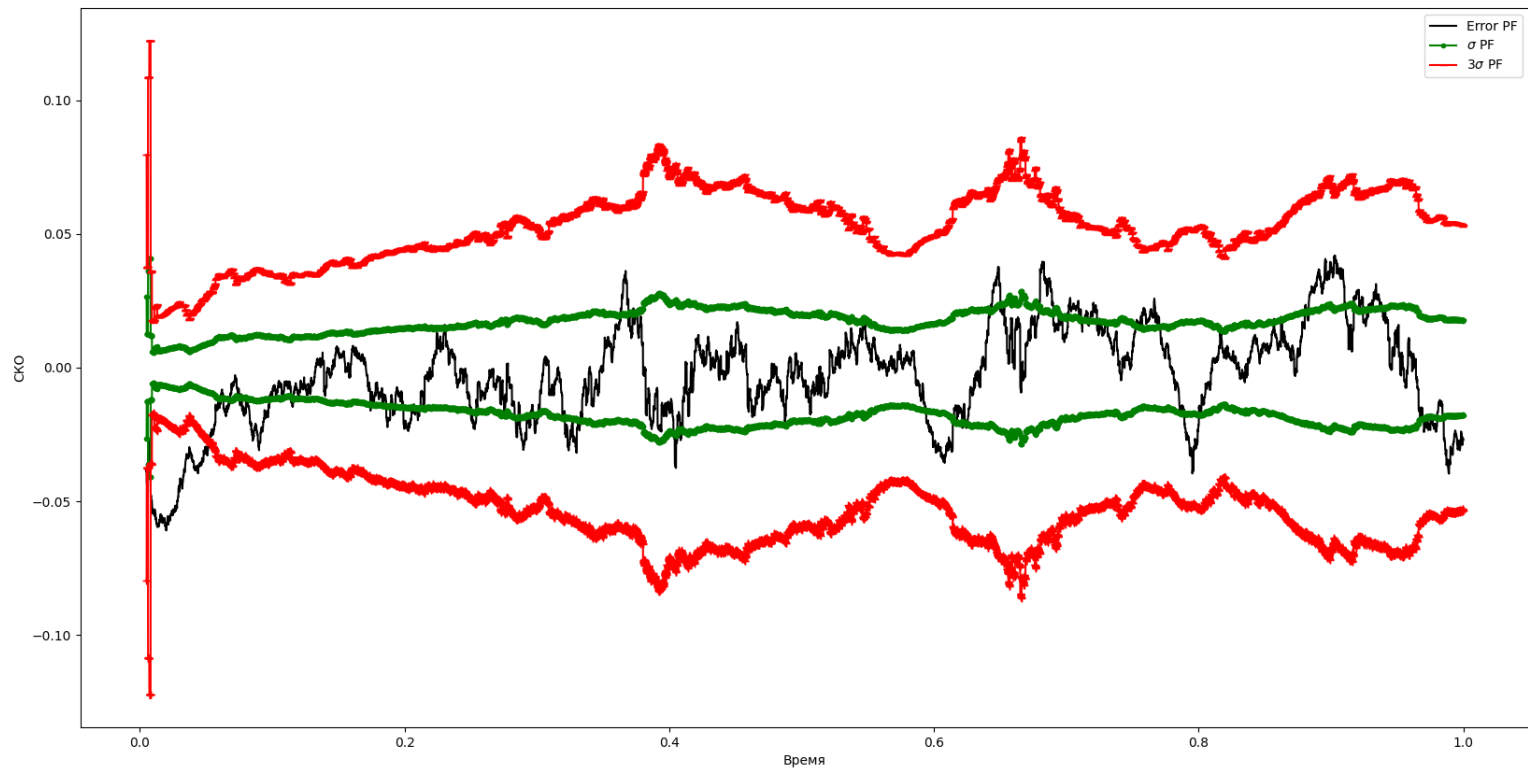
2.2.3 Ошибки оценивания  $x_2$



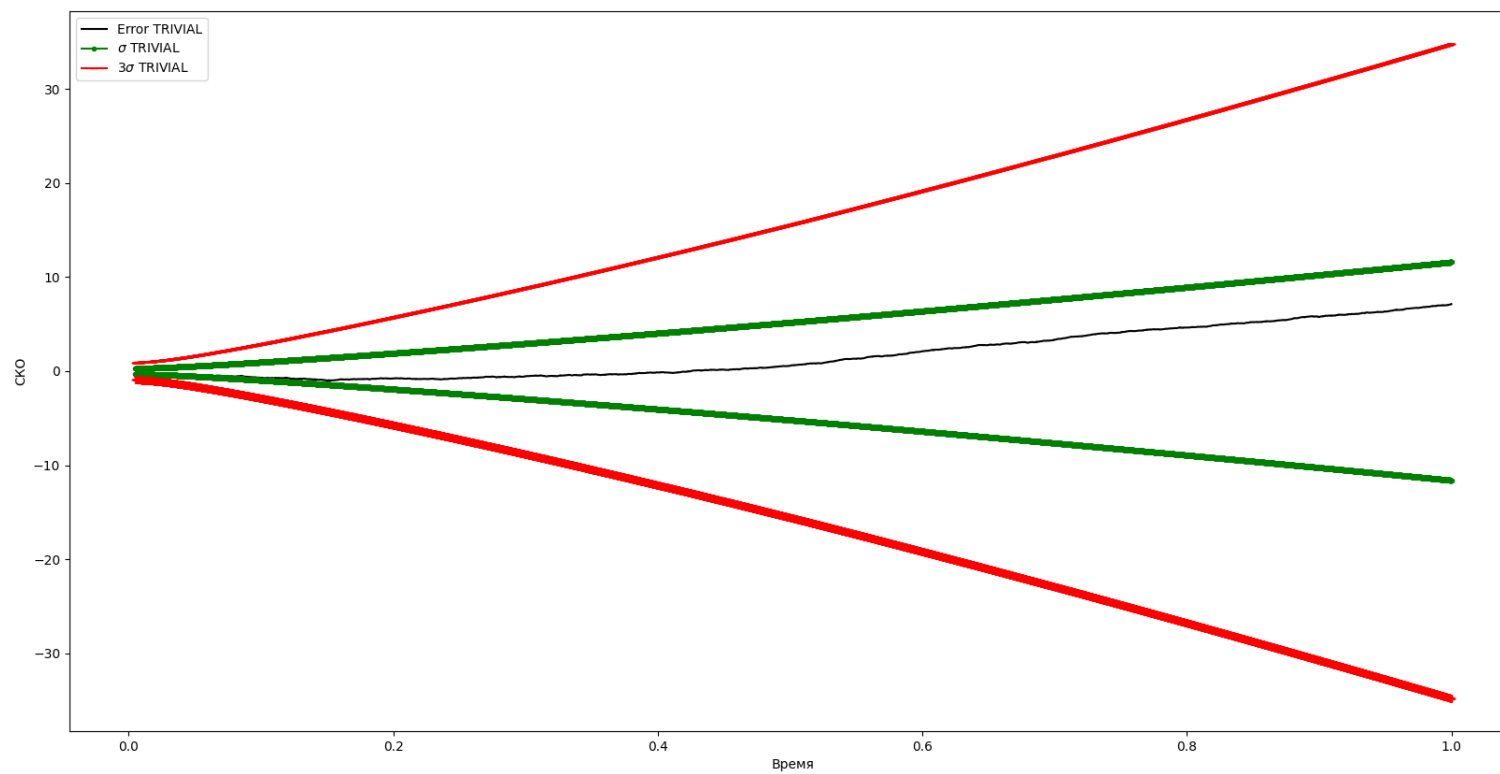






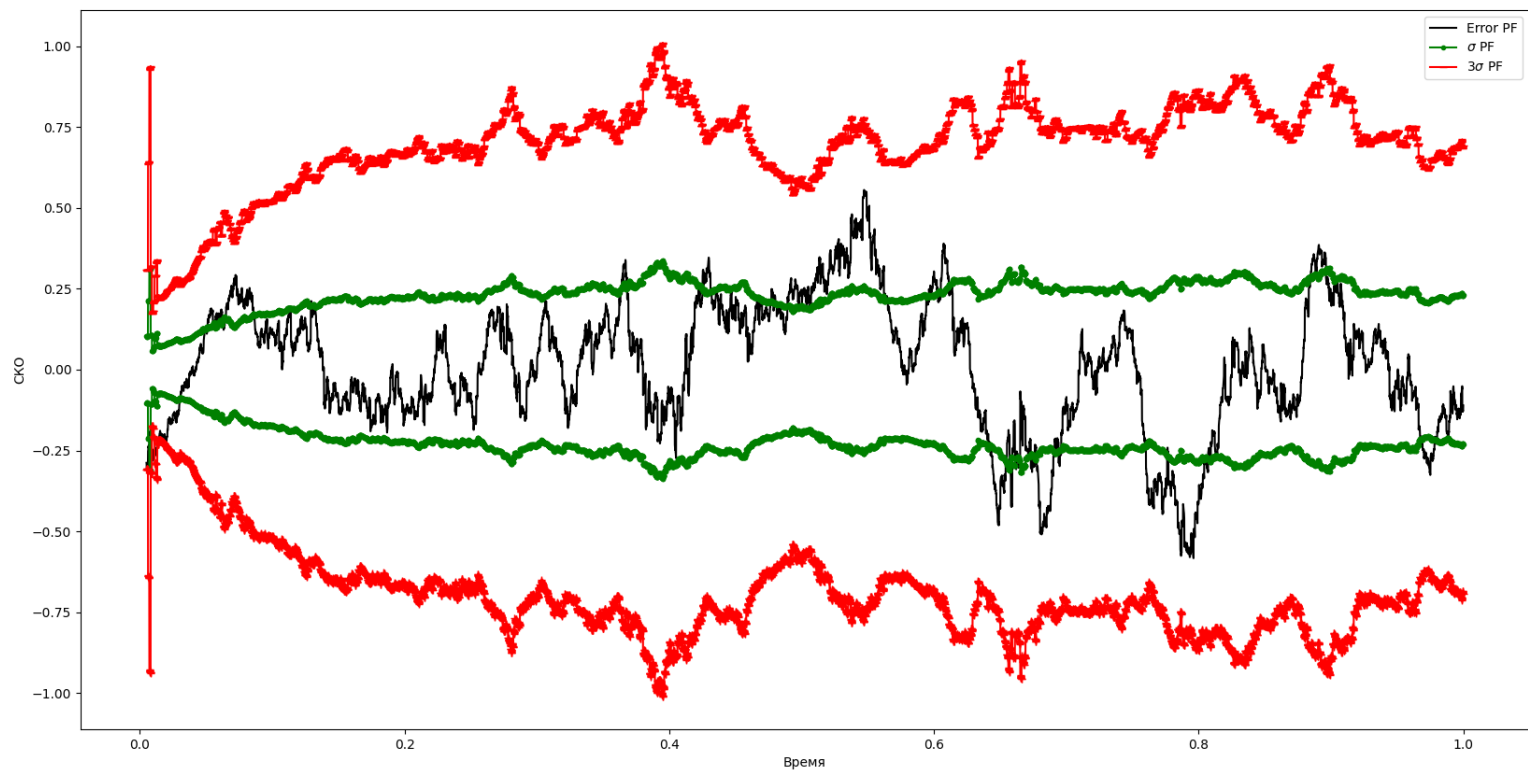


## 2.2.4 Ошибки оценивания $\theta$









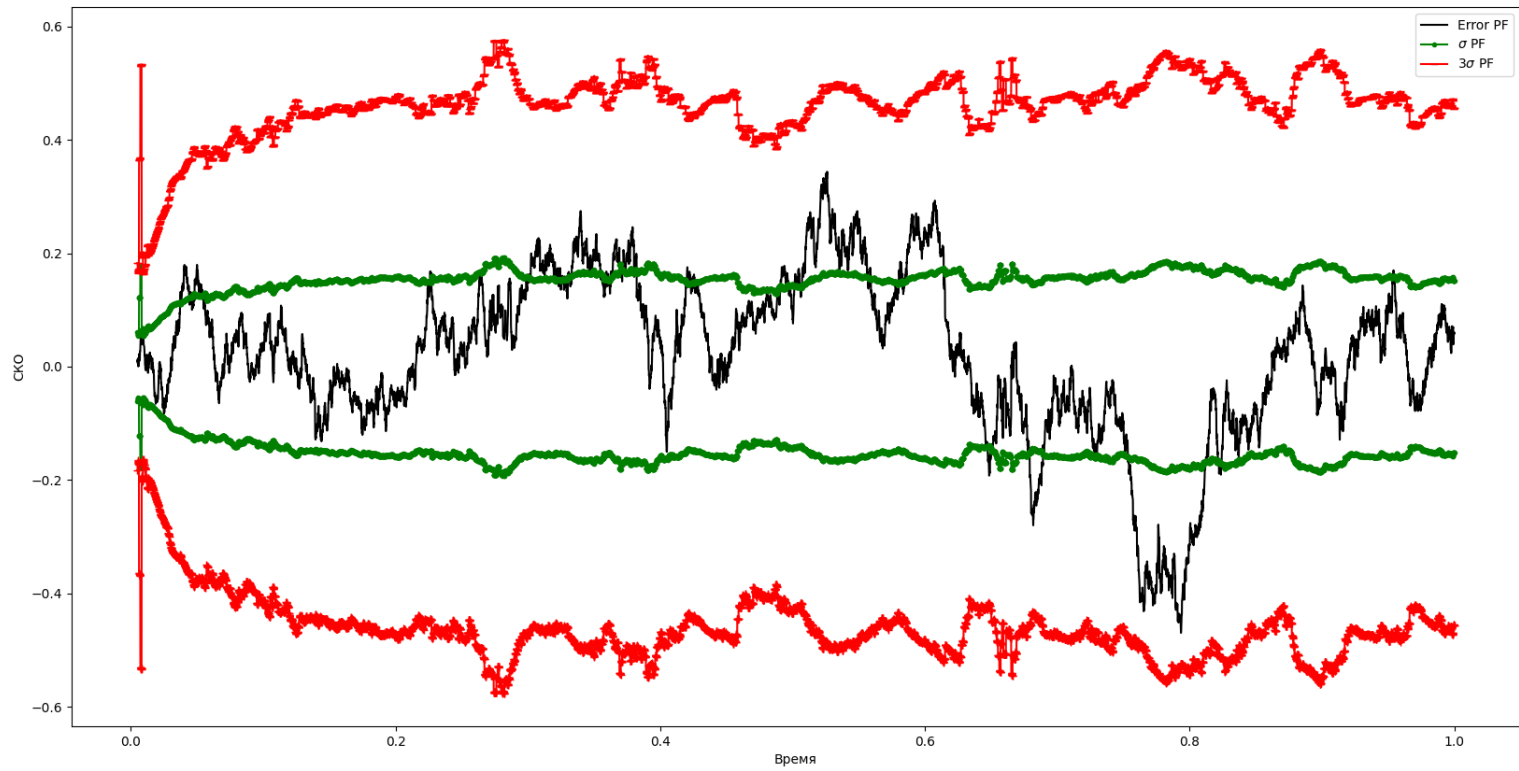
### 2.2.5 Ошибки оценивания $\phi$











## 2.2.6 Графики ошибок для нерасходящихся траекторий

Рис. 5: Ошибка  $x_1$

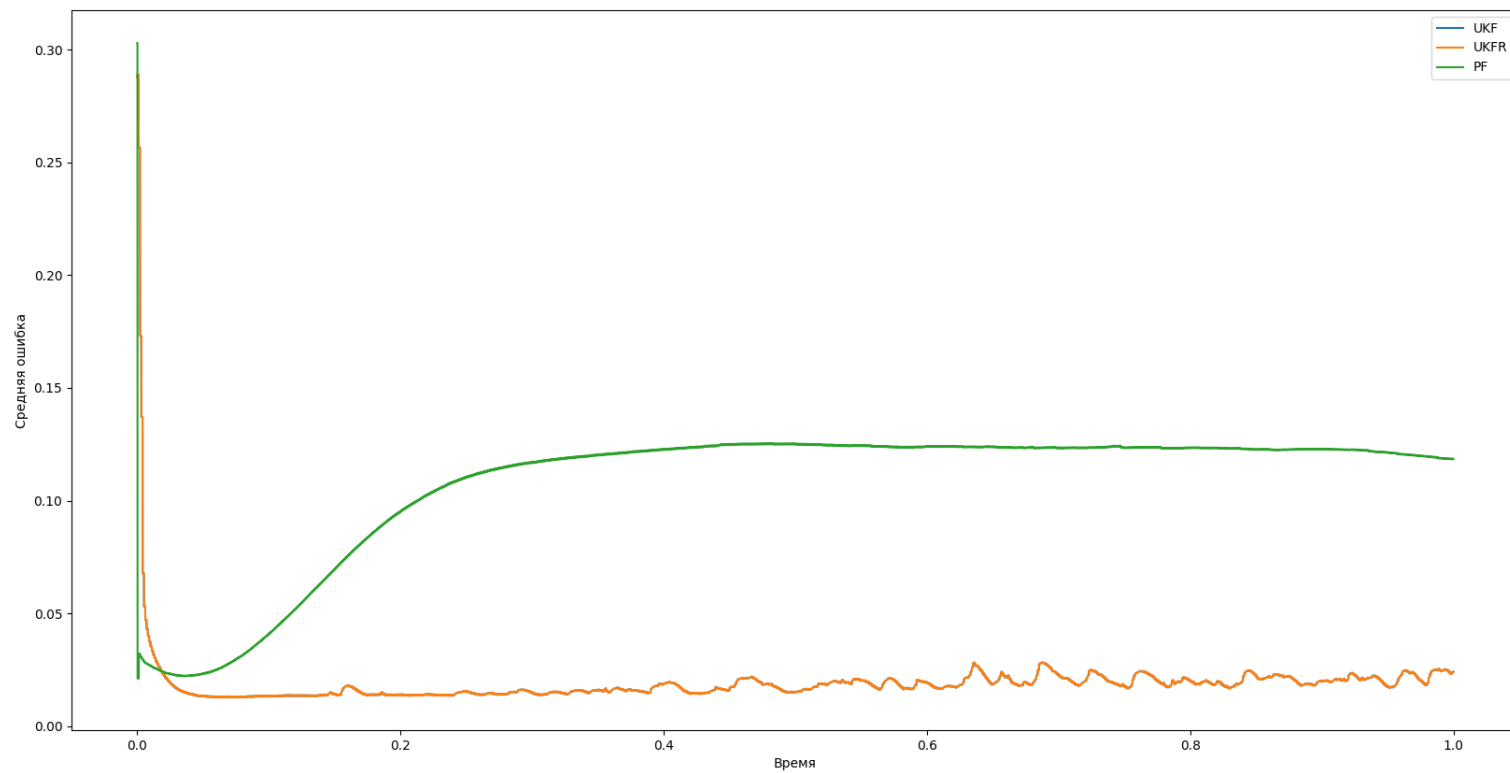


Рис. 6: Ошибка  $x_2$

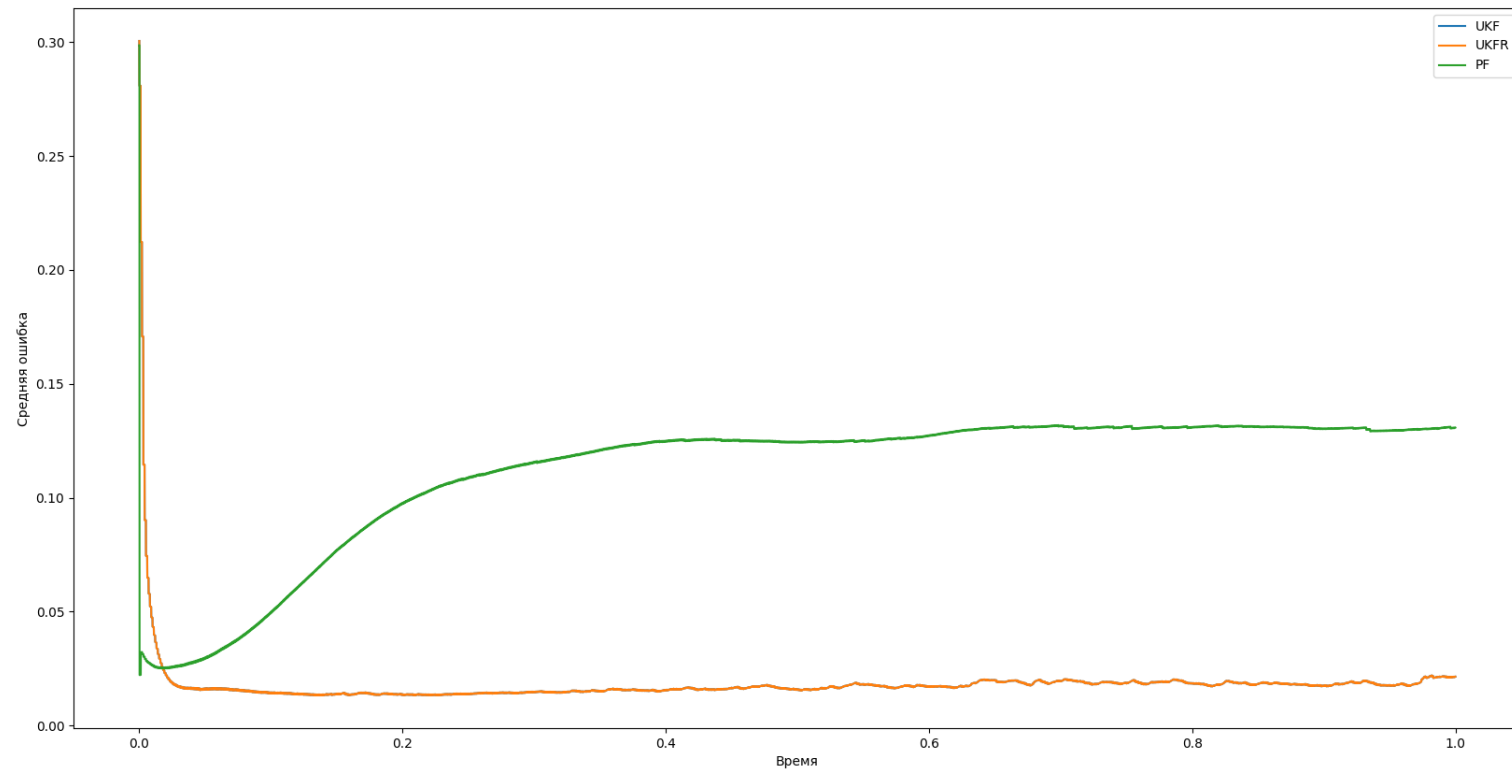


Рис. 7: Ошибка  $\theta$

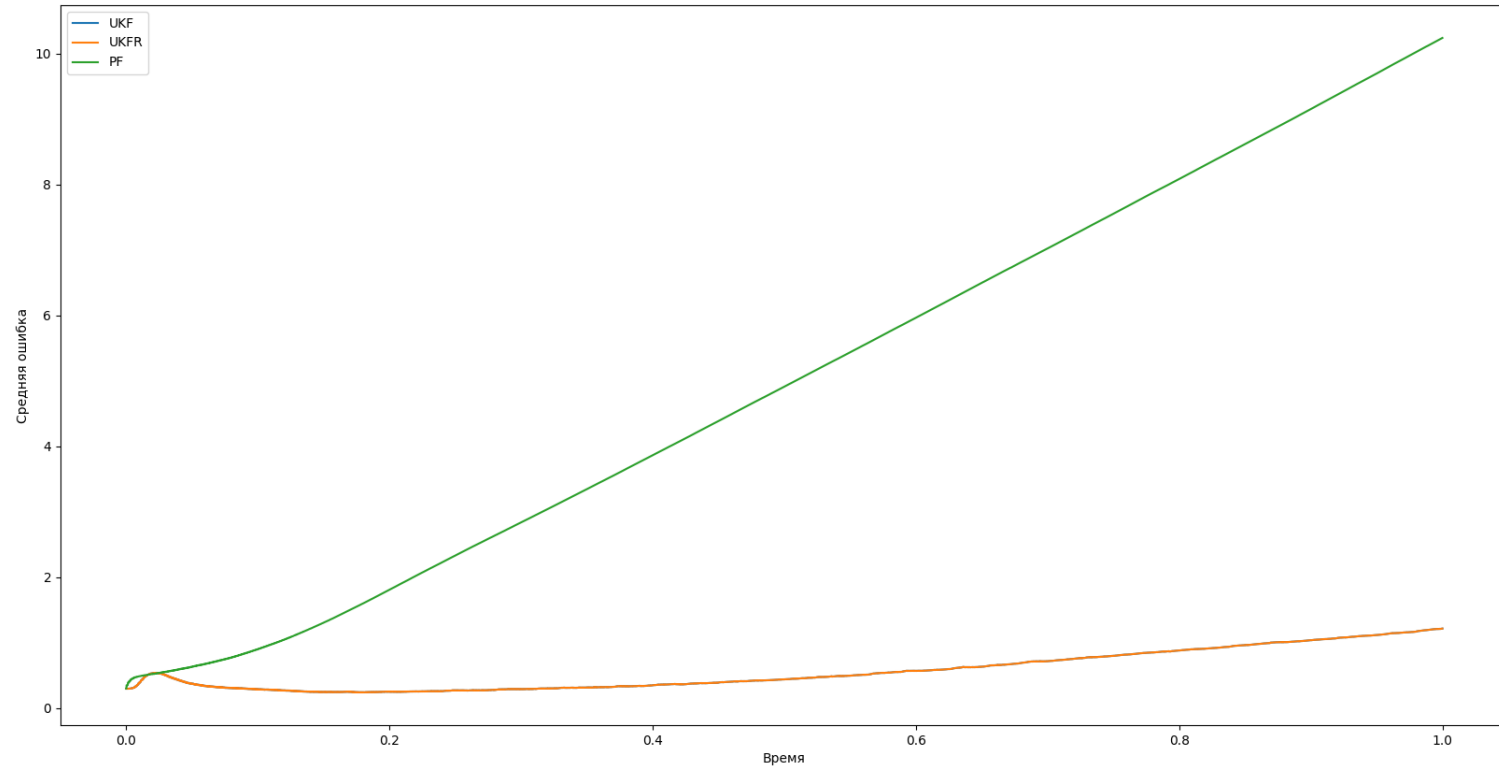
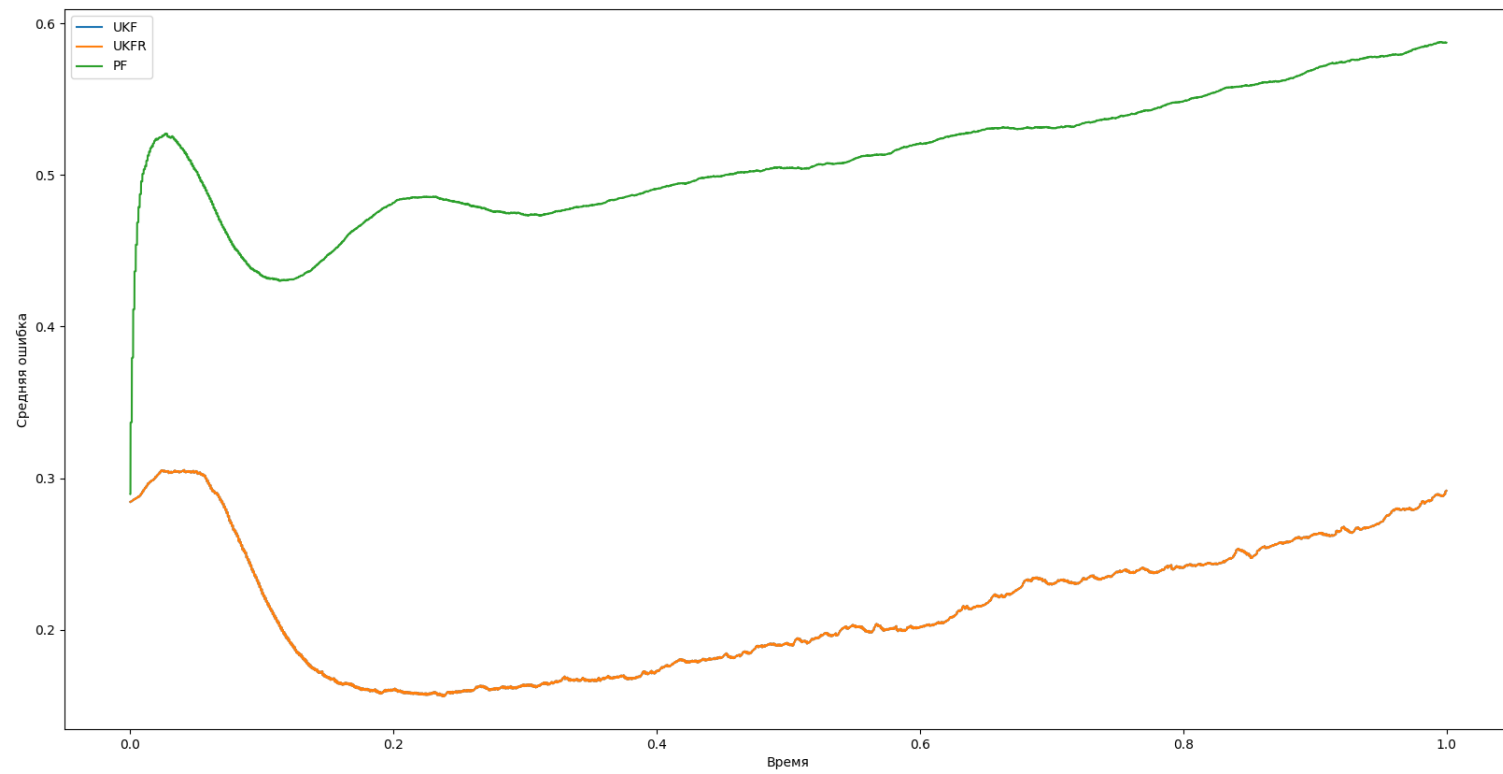


Рис. 8: Ошибка  $\phi$



### 2.2.7 Графики ошибок для всех траекторий

Рис. 9: Ошибка  $x_1$

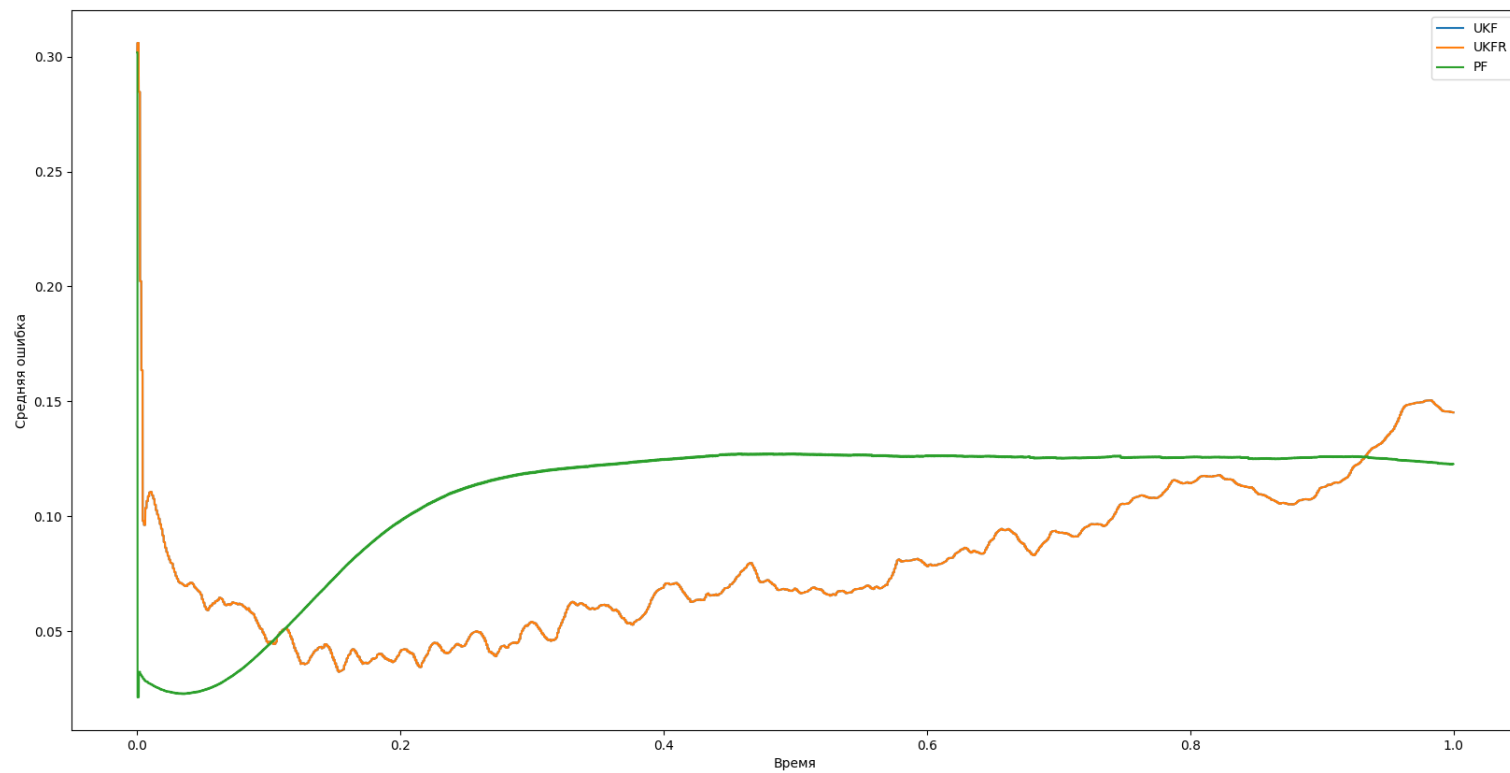


Рис. 10: Ошибка  $x_2$

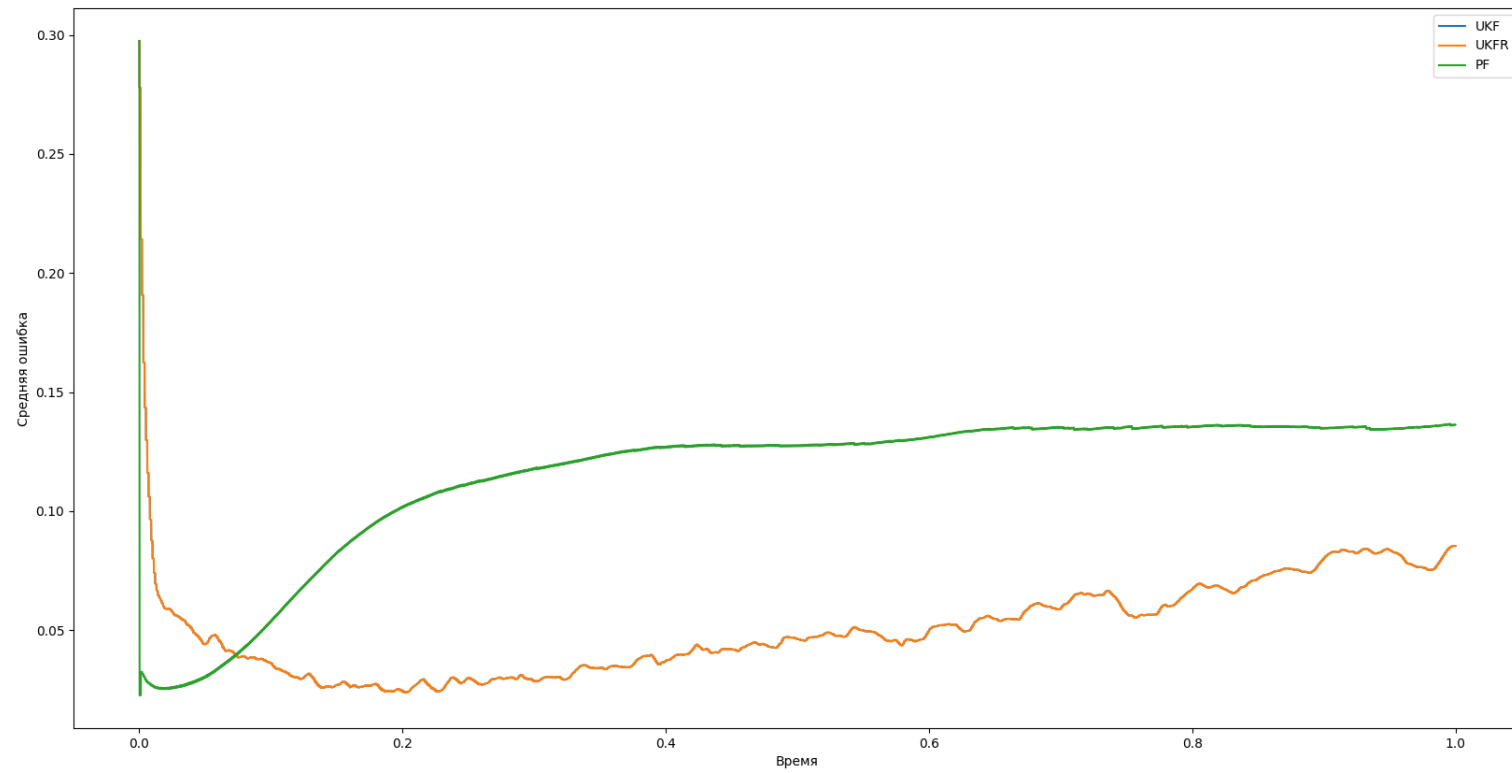




Рис. 11: Ошибка  $\theta$

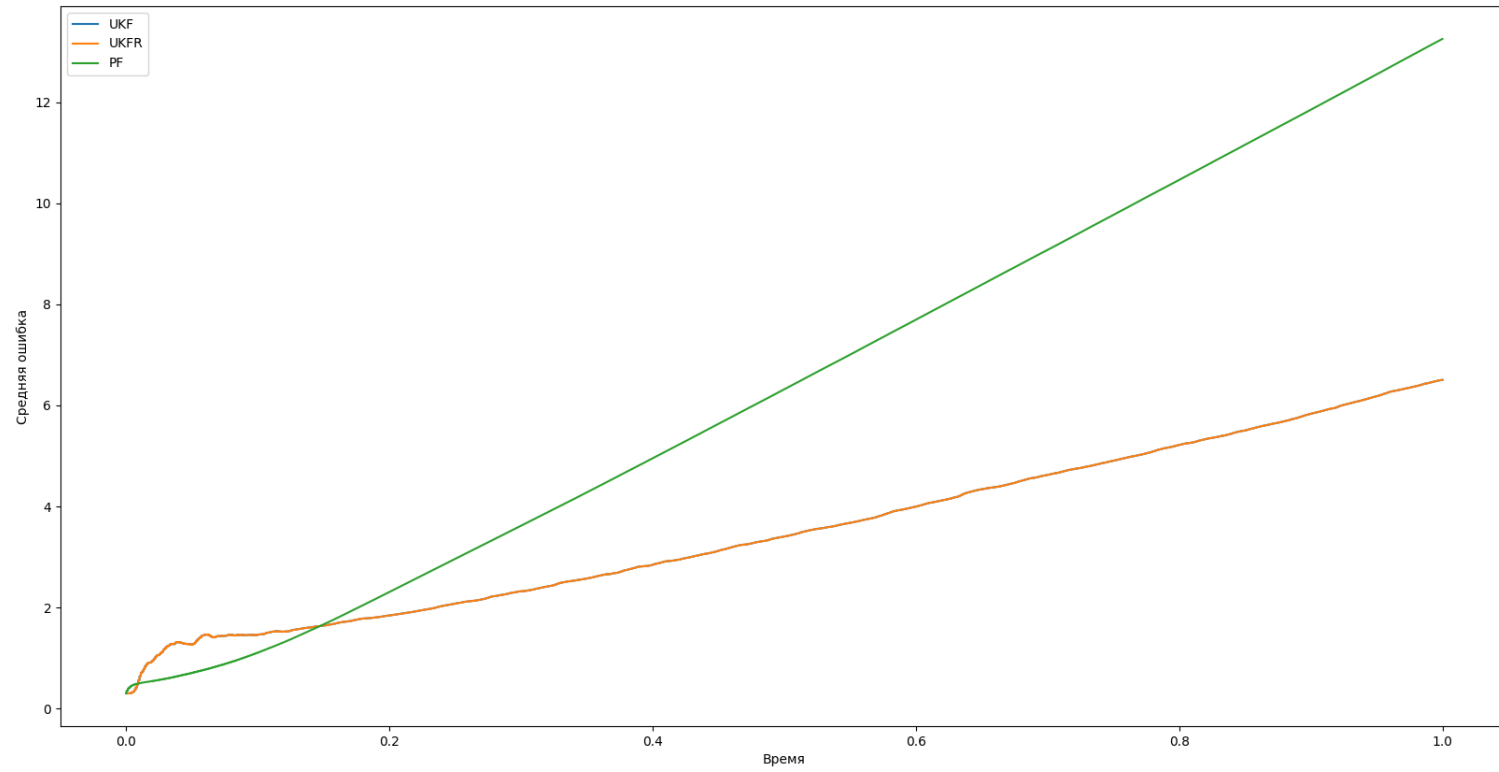
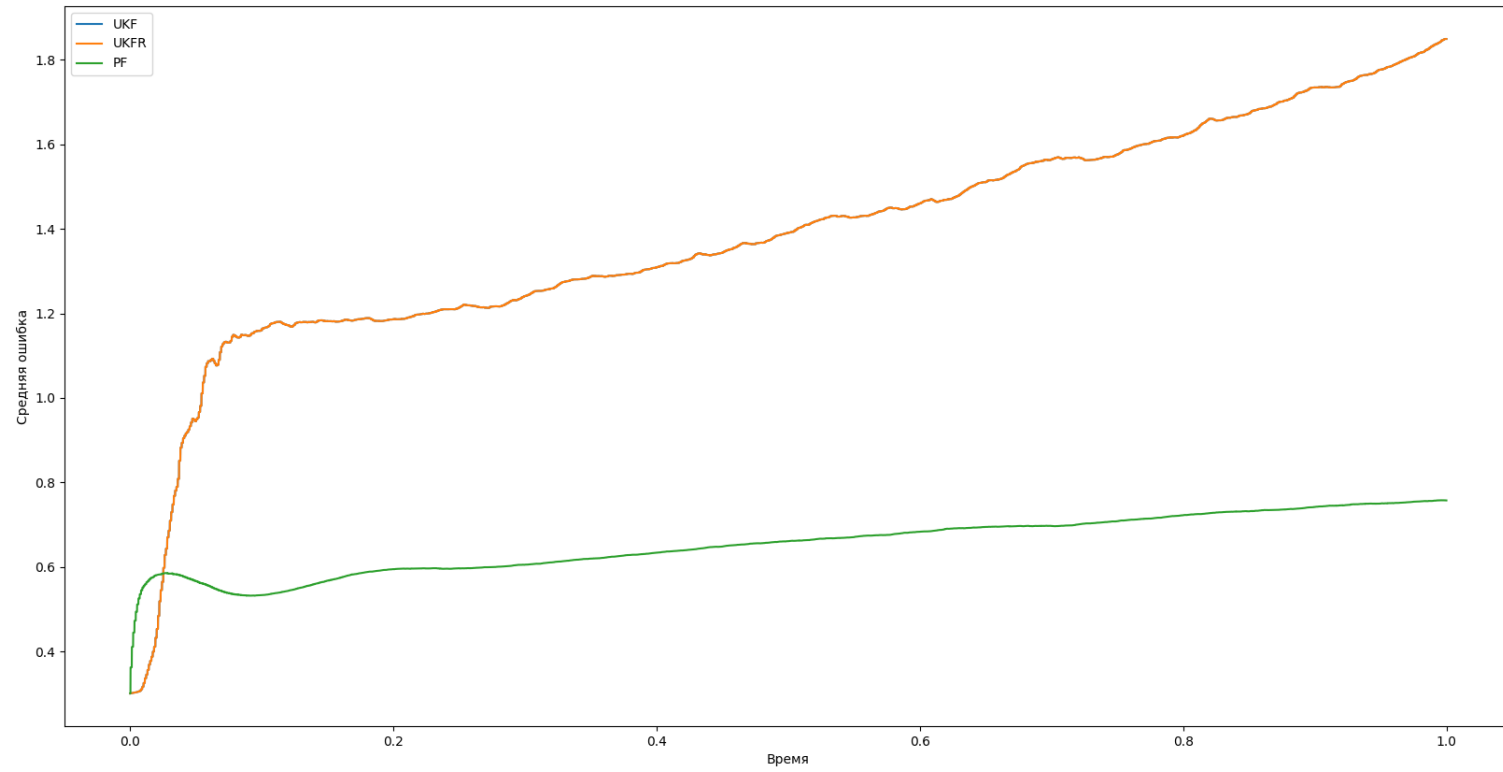


Рис. 12: Ошибка  $\phi$



### 2.2.8 Численные результаты

Метод	Процент расходящихся траекторий
UKF	16.72
UKFR	16.72
PF	6.09

Время	UKF	UKFR	PF
0.14	0.01454	0.01454	0.0401
0.43	0.01618	0.01618	0.05342
0.71	0.02032	0.02032	0.04768

**Таблица 1:** Средние ошибки фильтров на оценивания  $x_1$

Время	UKF	UKFR	PF
0.14	0.01348	0.01348	0.04384
0.43	0.01557	0.01557	0.05537
0.71	0.0187	0.0187	0.0506

**Таблица 2:** Средние ошибки фильтров оценивания  $x_2$

Время	UKF	UKFR	PF
0.14	0.51688	0.51688	0.88597
0.43	0.90071	0.90071	2.43857
0.71	1.44523	1.44523	4.00522

**Таблица 3:** Средние ошибки фильтров оценивания  $\theta$

Время	UKF	UKFR	PF
0.14	0.34427	0.34427	0.34907
0.43	0.4138	0.4138	0.35533
0.71	0.5228	0.5228	0.36586

**Таблица 4:** Средние ошибки фильтров оценивания  $\phi$

### 3 Выводы по расчётам

Из графиков видно, что оценки UKF и его корневой модификации ведут себя абсолютно одинаково. Это логично следует из вышеописанных особенностей реализации для 128-битных чисел. Более подробно о таком поведении фильтров Калмана написано в [соответствующем разделе документации](#).

Также можно заметить, что лучше всего фильтры справляются с оценкой местоположения тележки, а хуже всего - с углами  $\theta$  и  $\phi$ . Этому есть логичное объяснение: в качестве наблюдений мы получаем полярные координаты тележки. Благодаря этому мы достаточно точно можем строить оценку для  $x_1$  и  $x_2$ . В то же время, в наблюдениях никоим образом не участвуют углы, поэтому информацию о них мы получаем косвенно через перемещение тележки. Более того, поскольку угловая скорость  $u_2$  при расчётах взята равной нулю, поведение  $\phi$  описывается винеровским процессом с соответствующими параметрами. Из-за этого оценивать его сложнее всего, хотя и возможно из изменения  $\theta$ . Но, поскольку все наблюдения зашумлены, а при изменении состояния также добавляется шум, эффект накопления ошибки сильнее всего виден именно на этой компоненте состояния. Тем не менее, основную задачу можно признать выполненной - полученный алгоритм достаточно точно оценивает траекторию движения цели несмотря на помехи. Также из таблиц можно увидеть, что оценка, полученная фильтром частиц, часто оказывается хуже оценки UKF (ошибка больше примерно в 1.5-1.6 раза), но при этом процент расходящихся траекторий у PF сильно меньше (6 против почти 17). То есть, в данном случае большая вычислительная сложность оправдывается большей стабильностью. Если также посмотреть на графики средних ошибок на всех и на неразваливающихся траекториях, будет видно, что фильтр частиц ведёт себя и на тех и на других траекториях похожим образом, в то время как средняя ошибка сигма-точечного фильтра сильно зависит от типа траектории.

Если взглянуть на графики средних ошибок на всех траекториях, можно также заметить, что средняя ошибка растёт. Это может быть связано с тем, что, начиная с некоторого момента, тележка может начать кружить вокруг некоторой точки из-за неуправляемого роста угла поворота. В такие моменты становится сложно предсказать траекторию, так как малое изменение местоположения тележки по сравнению с погрешностью наблюдений сильно сказывается на работе фильтра, и для него такое движение с точки зрения постановки задачи становится мало отличимым от броуновского. Возможно, такого поведения цели можно избежать путём изменения управления с константного на зависящее от времени.