

### Министерство науки и высшего образования Российской Федерации

**федеральное государственное бюджетное образовательное учреждение высшего образования**

### «Московский государственный технологический университет «СТАНКИН» (ФГБОУ ВО «МГТУ «СТАНКИН»)

Институт автоматизации и робототехники Кафедра робототехники и мехатроники

Дисциплина «Мобильная робототехника»

# ОТЧЁТ

**по лабораторной работе № 1**

# «Фильтр Калмана»

# Вариант 7

## Выполнил:

студент группы АДМ-20-05

Пивкин Д.П.

(дата) (подпись) (ФИО)

## Принял

преподаватель:

## Собольников С. А.

(дата) (подпись) (ФИО)

## Оценка: Дата:

Москва 2022 г.

**Цель работы:** научиться реализовывать фильтр Калмана c помощью Scilab; исследовать зависимость свойств фильтра и ошибки оценивания от шумов.

### Фильтр Калмана

Фи́льтр Ка́лмана — эффективный рекурсивный фильтр, оценивающий вектор состояния динамической системы, используя ряд неполных и зашумленных измерений. Назван в честь Рудольфа Калмана.

Фильтр Калмана широко используется в инженерных и эконометрических приложениях: от радаров и систем технического зрения до оценок параметров макроэкономических моделей. Калмановская фильтрация является важной частью теории управления, играет большую роль в создании систем управления. Совместно с линейно- квадратичным регулятором фильтр Калмана позволяет решить задачу линейно- квадратичного гауссовского управления. Фильтр Калмана и линейно-квадратичный регулятор — возможное решение большинства фундаментальных задач в теории управления.

В большинстве приложений размерность вектора состояния объекта превосходит размерность вектора данных наблюдения. И при этом фильтр Калмана позволяет оценивать полное внутреннее состояние объекта.

Фильтр Калмана предназначен для рекурсивного дооценивания вектора состояния априорно известной динамической системы, то есть для расчёта текущего состояния системы необходимо знать текущее измерение, а также предыдущее состояние самого фильтра. Таким образом, фильтр Калмана, подобно другим рекурсивным фильтрам, реализован во временно́м, а не в частотном представлении, но в отличие от других

подобных фильтров, фильтр Калмана оперирует не только оценками состояния, а ещё и

оценками неопределённости (плотности распределения) вектора состояния, опираясь на формулу Байеса условной вероятности.

Алгоритм работает в два этапа. На этапе прогнозирования фильтр Калмана экстраполирует значения переменных состояния, а также их неопределённости. На втором этапе по данным измерения (полученного с некоторой погрешностью) результат экстраполяции уточняется. Благодаря пошаговой природе алгоритма, он может в реальном времени отслеживать состояние объекта (без заглядывания вперед, используя только текущие замеры и информацию о предыдущем состоянии и его неопределенности).

Для правильной работы фильтра Калмана якобы не требуется гауссовское распределение входных данных. В исходной работе Калмана результаты о минимуме ковариации фильтра были получены на базе ортогональных проекций, без предположений о гауссовости ошибок измерений. Затем просто было показано, что для специального случая распределения ошибок по Гауссу фильтр дает точную оценку условной вероятности распределения состояния системы.

Наглядный пример возможностей фильтра — получение оптимальных, непрерывно обновляемых оценок положения и скорости некоторого объекта по результатам временного ряда неточных измерений его местоположения. Например, в радиолокации стоит задача сопровождения цели, определения её местоположения, скорости и ускорения, при этом результаты измерений поступают постепенно и сильно зашумлены. Фильтр Калмана использует вероятностную модель динамики цели, задающую тип вероятного движения объекта, что позволяет снизить воздействие шума и получить хорошие оценки положения объекта в настоящий, будущий или прошедший момент времени.

### Результат выполнения лабораторной работы

**Листинг программы:**

1. clear; close(winsid());
2. mn = 100; //количество измерений
3. b\_size = mn+1;
4. dt = 1;//период измерения датчика
5. data = read("go7.txt", 3\*b\_size, 1);//читаем файл с данными
6. x(1,:) = data(1:b\_size);//фактическое измерение высоты каждую секунду(dt)
7. x(2,:) = data(b\_size+1:2\*b\_size);//фактическое измерение скорости
8. z = data(2\*b\_size+1:$);
9. x\_0 = [0; //начальное значение вектора состояния
10. 4];
12. P\_0 = [10 0; //начальное значение матрицы ковариации
13. 0 1];
15. A = [1 dt; //матрица динамики
16. 0 1];
18. R = [0 0; //матрица ковариации порождающих шумов
19. 0 3];
21. B = [0; 0];// матрица управления
22. u = 0; //вектор управления
23. C = [1 0]; //матрица измерений
24. Q = 2; //матрица ковариации шумов измерений
25. x\_est = zeros(2, b\_size); //матрица, которая для каждого вектора оценки сохраняет 2 значения
26. P\_est = zeros(2,2, b\_size);
27. x\_est(:,1) = x\_0;// инициализируем объект оценки и задаем первое значение
28. P\_est(:,:,1) = P\_0;
29. //задаем алгоритм фильтрации
30. for i = 2:b\_size
31. x\_pr = A\*x\_est(:,i-1) + B\*u;
32. P\_pr = A\*P\_est(:,:,i-1)\*A' + R;//матрица ковариации прогноза
33. K = P\_pr\*C'\*(C\*P\_pr\*C'+Q)^-1;
34. x\_est(:,i) = x\_pr + K\*(z(i) - C\*x\_pr);
35. P\_est(:,:,i) = (eye() - K\*C)\*P\_pr;
36. end
37. x\_est\_err = x\_est(1,:) - x(1,:); //ошибка оценивания
38. V\_est\_err = x\_est(2,:) - x(2,:);
39. //графики
40. figure(1); clf;
41. title('Истинное значение высоты, измерения и оценки высоты');
42. set(gca(),"auto\_clear", "off"); xgrid(1,0.1,10);
43. plot(0:mn, x(1,:),'b');
44. plot(0:mn, z, 'r\*');
45. plot(0:mn, x\_est(1,:), 'g');
46. xlabel('Время, с');
47. ylabel('Высота, м');
48. legend('Истинное значение высоты', 'Измерение', 'Оценка');
49. figure(2);clf;
50. subplot(1,2,1); set(gca(),"auto\_clear", "off"); xgrid(1,0.1,10);
51. title('СКО вертикальной скорости')
52. plot(0:mn, 3\*sqrt(squeeze(P\_est(2,2,:)))','r');
53. plot(0:mn, V\_est\_err, 'g');
54. plot(0:mn, -3\*sqrt(squeeze(P\_est(2,2,:)))','r');
55. xlabel('Время, с');
56. ylabel('СКО, м/c');
57. legend('3 sigma оценки','Действительная ошибка');
58. subplot(1,2,2); set(gca(),"auto\_clear", "off"); xgrid(1,0.1,10);
59. title('СКО высоты')
60. plot(0:mn, 3\*sqrt(squeeze(P\_est(1,1,:)))','r');
61. plot(0:mn, 3\*sqrt(Q)\*ones(1,b\_size),'b');
62. plot(0:mn, x\_est\_err, 'g');
63. plot(0:mn, -3\*sqrt(squeeze(P\_est(1,1,:)))','r');
64. plot(0:mn, -3\*sqrt(Q)\*ones(1,b\_size),'b');
65. xlabel('Время, с');
66. ylabel('СКО, м');
67. legend('3 sigma оценки', '3 sigma измерений','Действительная ошибка');
68. figure(3); clf;
69. title('Истинное значение высоты, измерения и оценки высоты');
70. set(gca(),"auto\_clear", "off"); xgrid(1,0.1,10);
71. plot(0:mn, x(2,:),'b');
72. plot(0:mn, x\_est(2,:), 'g');
73. xlabel('Время, с');
74. ylabel('Высота, м');
75. legend('Истинное значение скорости', 'Оценка скорости по измерениям');

Результат выполнения программы (Q=2):

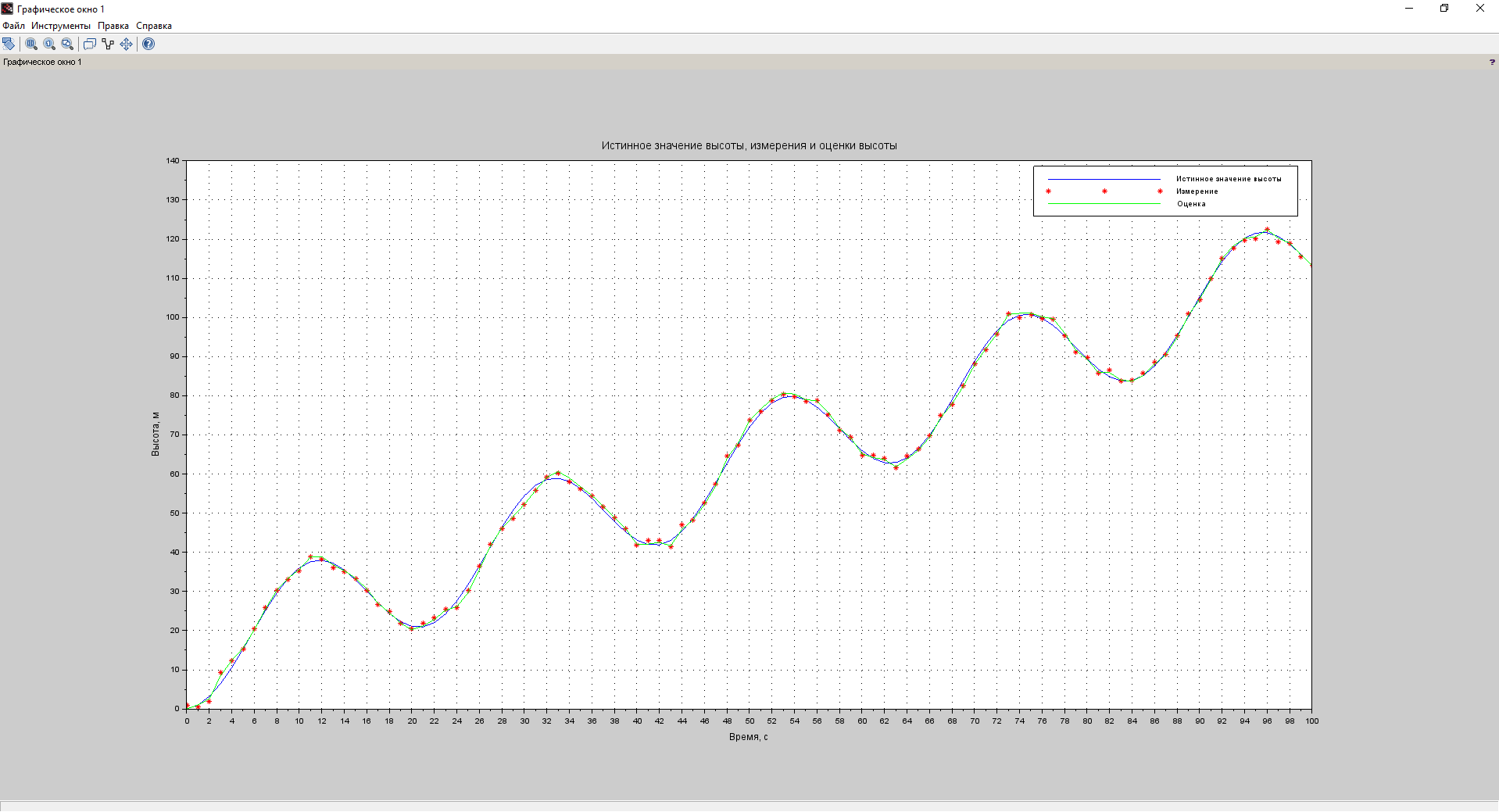


Рис.1 Истинное значение высоты, измерения и оценку высоты

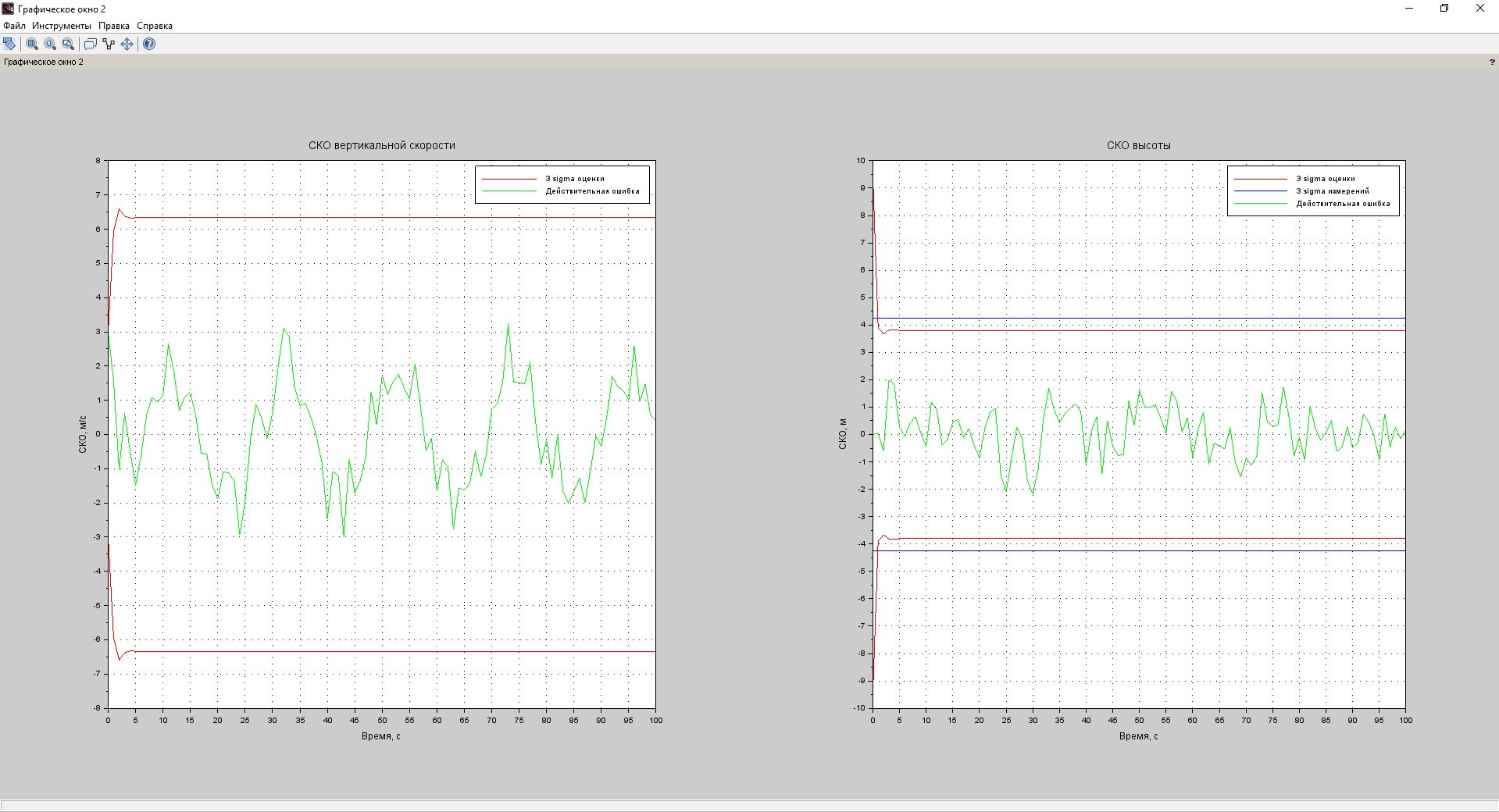


Рис.2 СКО вертикальной скорости; СКО высоты

## 

Рис.3 Истинное значение высоты, измерения и оценку высоты

Результат выполнения программы (Q=0.1):

## 

## Рис.4 Истинное значение высоты, измерения и оценку высоты

## 

## Рис.5 СКО вертикальной скорости; СКО высоты

## 

## Рис.6 Истинное значение высоты, измерения и оценку высоты

Результат выполнения программы (Q=10):

## 

## Рис.7 Истинное значение высоты, измерения и оценку высоты

## 

## Рис.8 СКО вертикальной скорости; СКО высоты

## 

## Рис.9 Истинное значение высоты, измерения и оценку высоты

## **Вывод:** в ходе выполнения лабораторной работы научился реализовывать фильтр Калмана с помощью Scilab, а также, исследовать зависимость свойств фильтра и ошибки оценивания от шумов