Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Факультет математики, механики и компьютерных технологий
Кафедра прикладной математики и программирования
Направление подготовки Прикладная математика и информатика

РАБОТА ПРОВЕРЕНА	ДОПУСТИТЬ К ЗАЩИТЕ
Рецензент,	Заведующий кафедрой, д.ф.-м.н., доцент
________________________	_______________А.А. Замышляева
«____»_____________2018г.	«____»_____________2018 г.

Моделирование случайных процессов методом формирующего фильтра

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ–01.03.02.2018.46.ПЗ ВКР

Руководитель работы, доцент
__________/А.Ю. Алексеева
«____»_____________2018 г.
Автор работы
студент группы ЕТ-412
________/ А.М. Голубчиков
«____»_____________2018 г.
Нормоконтролер, к.э.н., доцент
_________/Д.А. Дрозин
«____»_____________2018 г.

Челябинск 2018

АННОТАЦИЯ

Голубчиков А.М. Моделирование случайных процессов методом формирующего фильтра. – Челябинск: ЮУрГУ, ЕТ-412, 53 с., 21 ил., 1 табл., библиогр. список – 29 наим., 1 прил.

Цель данной работы – спроектировать формирующий фильтр для цифровой обработки сигнала с целью получить сигнал первоначальной, не искаженной шумами, формы. В рамках работы были рассмотрены методы моделирования случайных процессов, в частности метод формирующего фильтра. Было рассмотрено понятие цифровой фильтрации сигналов и приведен алгоритм работы фильтра Калмана, а также его упрощенный алгоритм. На основе чего была спроектирована программная система и ее интерфейс, выполнено тестирование и отладка системы.

В первом разделе рассматривается предметная область, описаны основные положения случайных величин и процессов. Описаны методы моделирования случайных процессов. Описана оценка параметров сигнала. Также, описаны научные работы, в которых использовался метод формирующего фильтра.

Второй раздел содержит алгоритм и формулы, необходимые для генерации случайного процесса и проектирования формирующего фильтра.

Третий раздел посвящен разработке программы. В нем приведены блок-схемы работы программы, описание интерфейса программы и описание функций, использующиеся в программе.

В четвертом разделе приведены результаты экспериментального тестирования программы.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В различных областях техники форму сигналов связывают с объектом исследования. В качестве примера этому может служить радиолокация, техническая и медицинская диагностика и др. Чаще всего, здесь имеют место случайные сигналы малой продолжительности во времени. Такие сигналы можно обрабатывать, например, с помощью линейного цифрового фильтра. В результате такой обработки форма сигналов и содержащаяся в них полезная информация могут быть сильно искажены. Поэтому особую актуальность приобретает разработка алгоритмов цифровой фильтрации сигналов, которые направлены на сохранение их первоначальной (не искаженной шумами) формы.

Преобладающим методом анализа и исследования сигналов является метод моделирования. Метод моделирования представляет собой способ теоретического анализа, а также практического действия, который направлен на разработку и работу с моделями. Под моделью же понимается образ некого реального объекта или процесса в идеальной, то есть описанной на каком-либо языке, форме, либо в материальной, который отражает существенные свойства моделируемого объекта или процесса. Данный образ используется во время различного рода исследований и экспериментов.

Некоторые измерительный приборы могут обладать различного рода погрешностью и на них могут быть оказаны влияния от большого количества как внешних, так и внутренних воздействий. Все это приводит к тому, что полезная информация, обрабатываемая измерителем, оказывается зашумлённой. Такие данные с измерителя обработать тем сложнее, чем сильнее зашумлены данные. Поэтому для решения подобных задач необходимы фильтры. Фильтр – это алгоритм обработки данных, который убирает шумы и лишнюю информацию.

Во время цифровой обработки сигналов обычно необходимо создавать средства численного преобразования массива заданного процесса изменения некоторой непрерывной физической величины, измеренного в дискретные моменты времени, для того, чтобы извлечь из него полезную информацию о другой физической величине, которая содержится в измеренном сигнале [1].

Целью данной работы является проектирование формирующего фильтра для цифровой обработки сгенерированного случайного процесса, а также проектирование интерфейса программы для фильтрации случайного процесса. Наиболее полная характеристика случайного процесса – это n-мерная плотность распределения вероятностей . При этом, чем больше значение n, тем более детально происходит описание случайного процесса. Но для решения большинства задач необходимо знать более простые характеристики, такие как математическое ожидание , дисперсия , корреляционная функция или спектральная плотность . Сам случайный процесс имеет вид функции, отличающаяся тем, что значения, которая она принимает в любые произвольные моменты времени по параметру t являются случайными. В общем случае случайный процесс – есть упорядоченная последовательность случайный величин. Если параметр t имеет дискретное значение, то и случайный процесс является дискретным, но если t – это непрерывное время, то случайный процесс является непрерывным во времени.

1. ОСНОВНЫЕ ПОЛОЖЕНИЯ МОДЕЛИРОВАНИЯ СЛУЧАЙНЫХ ПРОЦЕССОВ

Математическая модель являет собой формализованное описание системы на некотором абстрактном языке, которое обеспечивает имитацию работы системы на уровне, довольно близком к настоящему поведению, получаемому во время реальных испытаний систем. Математические модели описывают объекты, явления и процессы с некоторой степенью приближения к действительности [2].

Целью математического моделирования является анализ реальных процессов, протекающих в природе или происходящих, например, на производстве математическими методами. Такая модель может иметь вид математического выражения с переменными, где поведение этих переменных совпадает с поведением реальной системы. Также, модель может содержать случайные величины.

1.1 Моделирование случайных величин

Во время разработки и использования моделей, в частности имитационных, не редко приходится моделировать важный класс факторов – случайные величины разного вида.

Случайная величина ξ – это такая переменная, которая в результате некоторого эксперимента или испытания принимает заранее неизвестное значение.

Различают три вида случайной величины: дискретные, непрерывные и смешанного типа.

Дискретная случайная величина – это такая случайная величина, у которой множество значений не более чем счетно, то есть ее значения можно пересчитать. Кроме того, значения такой случайной величины не содержат какой-либо непрерывный интервал на числовой прямой.

Непрерывной случайной величиной называют случайную величину, которая может принимать всевозможные значения из некоторого конечного или бесконечного промежутка. Кроме того, число таких значений бесконечно.

Смешанная случайная величина характерна тем, что ее функция распределения на некоторых участках непрерывна, а в некоторых отдельных точках имеет разрывы.

Для случайной величины ξ важной характеристикой является ее закон распределения. Он может быть задан по-разному: для всех типов случайных величин – это функция распределения; для непрерывных случайных величин – плотность вероятности или плотность распределения; для дискретных случайных величин – это таблица или ряд распределения.

Моделирование случайной величины – есть процесс получения выборочных значений . Данные значения статистически независимы, при этом они имеют одинаковое вероятностное распределение, которое совпадает с распределение случайной величины ξ.

Для формирования случайных величин на ЭВМ с различными законами распределения используют случайные числа, которые распределены на интервале (0, 1), вырабатывающиеся программным датчиком случайных чисел [2]. Такие случайные величины называются псевдослучайными.

Рассмотрим моделирование нормально распределенных случайных величин, так как нормальный закон распределения можно встретить довольно часто в природе. Формула распределения значений случайной величины ξ по нормальному закону имеет следующий вид:

= ,

где y( ) – вероятность принятия случайной величиной значения , а – случайная величина.

Из этой формулы можно увидеть, что нормальное распределение имеет два параметра – это математическое ожидание и среднеквадратичное отклонение .

Нормальное распределение может быть нормализованным. В этом случае у нормального распределения , а . Кроме того, по нормализованному распределению можно получить и другое нормальное распределение с математическим ожиданием и среднеквадратичным отклонением, если использовать формулу:

.

1.2 Моделирование случайных процессов

Известно, что под случайным процессом понимается тот процесс, который описывается случайной функцией (t). Ее значение, в свою очередь, в любой момент времени t являет собой случайную величину с определенным законом распределения плотности вероятности. К наиболее полной характеристике случайного процесса (t) относится n-мерная плотность распределения вероятностей [3]:

.

Чем больше значение n, тем более точно описывается случайный процесс, но для решения некоторых задач будет достаточно и более простых характеристик, таких как:

- математическое ожидание случайного процесса ξ(t)

;

- дисперсия случайного процесса

;

- корреляционная функция случайного процесса

, где M{ } – это оператор математического ожидания.

В случае стационарного случайного процесса m(t)=m; D(t)=D; R( )=R(τ).

Также, важной характеристикой случайного процесса является спектральная плотность мощности G(ω). Так, для центрированного случайного процесса, у которого математическое ожидание m равно нулю, G(ω) можно представить в виде преобразования Фурье от корреляционной функции R(τ):

. (1)

При этом имеет место обратное преобразование Фурье:

— . (2)

Случайные процессы различают по степени однородности их протекания во времени [3].

Первый класс случайных процессов носит название нестационарных. В таких процессах функции математического ожидания, дисперсии и корреляции могут изменяться во времени, то есть зависеть от момента времени t.

Следующий класс случайных процессов – это стационарные процессы. В таких процессах плотность вероятности не имеет зависимости от начала отсчет времени, а на интервале его существования выполняется условие постоянства для математического ожидания и дисперсии. При этом функция корреляции – есть функция только разности аргументов τ = :

,

—— .

По последнему выражение можно сделать вывод о четности корреляционной функции и корреляционной функции коэффициентов.

Наконец, существуют эргодические процессы. Большинство стационарных процессов обладают эргодическим свойством. Суть его заключается в следующем: по одной длинной реализации процесса можно сделать вывод о всех статистических свойствах этого процесса так же, как по любому количеству реализаций, то есть закон распределения случайных величин в таком процессе может быть одинаковым и по сечению для ансамбля реализаций, и по координате развития. Такого вида процессы и являются эргодическими. Для них имеет место:

– ,

– ,

– .

Свойство эргодичности является важным свойством случайных процессов, при этом только стационарных. Для эргодического случайного процесса математическое ожидание равно постоянной составляющей его реализации, притом любой. Дисперсия же – это мощность флюктуационной составляющей. Определение функции происходит по ограниченным статистическим данным одной реализации, которое является некоторым приближением к соответствующим фактическим функциям процессов. Поэтому такие функции можно называть статистическими.

Важно и то, что свойства эргодичности могут возникать только по отношению к первым двум моментам случайного процесса. Этого достаточно, чтобы использовать соответствующие методики исследования процессов. На практике, чтобы проверить исследуемый процесс на эргодичность, выполняют проверку выполнения условия Слуцкого:

$$- \quad .$$

Если при возрастании значения аргумента τ ковариационная функция стремится к 0, то такой процесс относится к эргодическим относительно моментов первого и второго порядков.

1.3.1 Методы моделирования случайных процессов

Принцип моделирования случайных процессов совпадает с принципом моделирования случайных величин. Только в этом случае при моделировании используют модель базового случайного процесса – стационарного нормального белого шума. Затем при помощи нелинейного преобразования и линейного формирующего фильтра из базового процесса формируется случайный процесс с необходимым законом распределения и корреляционной функцией. К наиболее распространенным методам относятся:

- метод скользящей средней;
- рекуррентные моделирующие алгоритмы;
- метод формирующего фильтра.

Во время моделирования случайного процесса вида белого шума требуется датчик последовательности { }, i=1,2,… независимых случайных величин. Данная последовательность – это дискретный широкополосный случайный процесс. Тогда корреляционная функция будет иметь вид:

$$- \quad , \tag{3}$$

где – интервал дискретизации, – дисперсия случайной величины .

Энергетический спектр равен:

$$\frac{}{} . \tag{4}$$

На частотах, которые близки к нулевой энергетический спектр равен:

$$.$$

Смысл алгоритма моделирования центрированного белого шума состоит в том, что здесь используется базовый датчик величины x с равномерным законом распределения на интервале (0, 1):

(5)

– это необходимая спектральная плотность белого шума. Сам белый шум такого вида используется в имитационной модели системы с эффективной полосой пропускания, которая в свою очередь меньше ширины спектра (4).

Ранее упоминалось, что модель случайного процесса с заданной одномерной плотностью распределения вероятностей q(y) и корреляционной функцией получается из базовой модели случайного процесса. А преобразование базового случайного процесса происходит при помощи нелинейного функционального преобразователя и линейного формирующего фильтра. Нелинейное безынерционное преобразование влечет получение заданного закона распределения q(y), а преобразование при помощи формирующего фильтра обеспечивает получение заданной корреляционной функции .

На Рисунке 1 представлена структура моделирования:

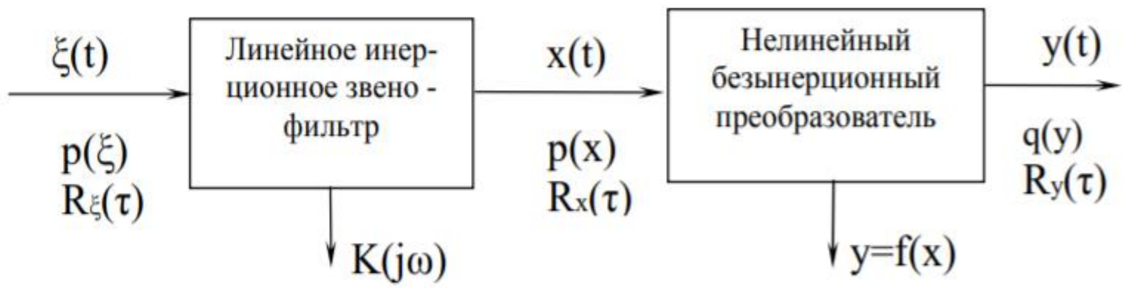

Рисунок 1 – Структура моделирования

- знаная p(ξ), p(x), q(y), найдем нелинейное преобразование y=f(x);
- зная и f(x), найдем необходимую корреляционную функцию ;
- зная , , найдем комплексную частотную характеристику формирующего фильтра K(jω);
- осуществим цифровое моделирование по приведенной схеме.

При этом, наибольшие затруднения возникают при реализации 2-го этапа – определение .

Известные методы можно разделить на две группы: точные и приближенные. К точным относится, например, метод рекуррентных алгоритмов дискретизации. В таких методах отсутствует методическая ошибка по корреляционной функции.

К приближенным методам относятся методы формирующего фильтра, скользящего суммирования. Для них равенство заданных и воспроизводимых на ЭВМ характеристик выдерживается с некоторой погрешностью.

Метод факторизации заданной спектральной плотности приводит к разностному уравнению формирующего фильтра. Сначала происходит факторизация

. (6)

После чего по найдем передаточную функцию .

Затем с помощью метода z-преобразования по определим передаточную функцию дискретного фильтра в виде:

. (7)

Учтем свойства обратного z-преобразования:

запишем разностное уравнение:

.

Начальные значение при к < 0 принимают нулевыми.

Рассмотрим пример получения разностного уравнения для формирующего фильтра, где на выходе случайный процесс имеет спектральную плотность, равную: . Далее после факторизации получаем:

следовательно,

. (8)

При переходе к воспользуемся методом δ-аппроксимации, т.е. когда дискретное звено получаем при добавлении к непрерывному звену на его выходе корректирующего элемента в форме безынерционного ключа с коэффициентом передачи . Воспользовавшись таблицами z-преобразования, для , с учетом (8), получим:

, (9)

где = , d=exp( ).

Из (9), с учетом (7), получим разностное уравнение формирующего фильтра:

.

При этом рекомендуемое значение шага .

Другой метод, который имеет название метод скользящего суммирования применяется в том случае, если факторизация спектральной плотности затруднительна. Алгоритм такого моделирования имеет вид:

(10)

где – случайные некоррелированные числа с единичной дисперсией, нулевым средним, а также произвольным законом распределения; – это весовые коэффициенты; .

Число N определяется исходя из требований близости дисперсии чисел к заданной дисперсии моделируемого процесса { }, k=1,2,…

Данный метод применим к процессам с не дробно-рациональными спектральными плотностями.

Далее рассмотрим следующий пример случайного процесса с гауссовой формой энергетического спектра и корреляционной функцией . Такой процесс может быть смоделирован по алгоритму (10), если взять .

Для класса случайных процессов, для которых спектры имеют вид дробно-рациональной функции S(ω)=A(ω)/B(ω), где A(ω) и B(ω) – это полиномы степени l ($m>l$), оператор линейного преобразования формирующего фильтра имеет вид:

,

где c[k] – это весовые коэффициенты, которые определяются заданной функцией корреляции (или энергетическим спектром и определяется по следующей формуле:

,

где – это модуль комплексной частотной характеристики формирующего фильтра.

Известно, что в спектре S(ω) на частотах | |> составляющими можно пренебречь, поэтому во время вычисления c[k] можно использовать нормированную частоту x= / . В этом случае:

,

где = .

Отсюда получаем, что c[k] – это коэффициенты Фурье в разложении функции на интервале (-1,1). Если взять к стремящимся к бесконечности, то c[k] достаточно быстро стремится к нулю. В этом случае можно взять усеченную сумму и использовать ее в формуле скользящего суммирования:

.

Смысл алгоритма метода скользящего суммирования заключается в том, что элемент u[n] моделируемого процесса формируется суммированием N=2p+1 независимых величин ξ, которые являются последовательно формируемыми. Тогда сетка весовых коэффициентов c[k] «скользит» относительно базовой последовательности случайной величины ξ. При этом статистическая зависимость u[n] и u[n+m] является следствием наличия общих слагаемых в суммах с различными весами. Чтобы ход вычислений проходил удобнее можно использовать другую запись алгоритма (ввиду того, что процессы ξ(t) и u(t) стационарные):

где N=2p+1, =c[k-p-1].

Параметр p ограничивает количество слагаемых, поэтому его можно выбирать исходя из условия:

где – это заданное малое число или ошибка алгоритма; – это дисперсия процесса u(t). Отметим, что так как весовая функция c[k] является четной (c[k]=c[-k]), во время моделирования в памяти ЭВМ можно хранить лишь c[k] при k≥0.

Далее перечислим основные этапы моделирования:

- При заданных спектрах S(ω), , рассчитывается и запоминается (p+1) весовых коэффициентов c[k] при k≥0;
- Выбирается последовательность нормальных случайных величин ξ с нулевым математическим ожиданием ( ) и дисперсией, равной единице ( );
- По формуле скользящего суммирования выбирается случайная величина u[n];

Рассмотрим пример моделирования стационарного нормального процесса с треугольной корреляционной матрицей:

1) рассчитывая c[k], получаем, что одинаковые весовые коэффициенты: c[k]=c[0]= , где N=[1/ϒ]+1, [x] – это целая часть числа x, а . При этом, методическая ошибка отсутствует, если 1/ϒ – целое число;
2) формирование случайной величины ξ;
3) расчет случайной величины – равновесное скользящее суммирование.

В качестве базовых случайных величин ξ при N>8 выбирать случайные величины с равномерным законом распределения на интервале (0,1) можно только в случае равновесного скользящего суммирования, потому что нормализация исходной последовательности и формирование корреляционных связей происходит в один момент времени. Тогда алгоритм моделирования принимает вид:

1.4 Оценка параметров

Оценкой параметра θ – это такая функция результатов наблюдений над случайной величиной X (иначе — статистику) [4,5], с помощью которой судят о значениях параметра θ.

Оценка является случайной величиной (в отличии от самого параметра). Такая величина обладает наименьшим рассеянием относительно оцениваемого параметра θ, например, должна быть наименьшей величиной математического ожидания квадрата отклонения оценки от параметра, который оценивается: .

Оценка параметра θ может обладать свойством несмещенности, состоятельности и эффективности.

Оценка является несмещенной в том случае, если ее математическое ожидание – есть сам оцениваемый параметр: . Если это условие не выполняется, то оценка является смещенной. В этом случае оценка , полученная по некоторым выборкам, в среднем будет завышать значение оцениваемого параметра (если) или занижать его (если). Отсюда следует, что требование несмещенности является гарантией того, что систематические ошибки будут отсутствовать при оценивании.

Оценка удовлетворяет свойству состоятельности, если она удовлетворяет закону больших чисел, то есть по вероятности сходится к оцениваемому параметру :

.

При использовании состоятельных оценок оправдывается увеличение объема выборки, потому что значительные ошибки становятся маловероятными при оценивании. Отсюда, с практической точки зрения, имеют смысл только состоятельные оценки.

Кроме того, несмещенная оценка может обладать свойством эффективности, если ее дисперсия является наименьшей из всех возможных других несмещенных оценок параметра , которые рассчитаны по выборкам одинакового объема n.

Для несмещенной оценки ее дисперсия – это . Поэтому эффективность является главным свойством, определяющим качество оценки.

Для того, чтобы найти оценки параметров (характеристик) генеральной совокупности существует ряд методов. К ним относится, например, метод максимального правдоподобия и медиана абсолютного отклонения.

1.4.1 Метод максимального правдоподобия

Метод максимального правдоподобия – это наиболее важный с теоретической точки зрения метод нахождения функции f (которая определяет решающее правило оценивания), преобразующей выборочные данные X в значение оценки =f(X) [6]. Как частный случай данный метод впервые был использован К.Гауссом. Что касается общего метода, то он был предложен американским статистиком Р.Фишером в 1912 году.

При использовании метода требуется задать статистическое описание совокупности сигнала и шума, которые имеют вид условной плотности вероятности W(X/), которая зависит от некоторых выборочных данных и от измеряемых параметров . При этом значения параметров являются не известными.

Для того, чтобы суть метода представлялась в понятном виде, рассмотрим простой пример.

Необходимо измерить постоянное напряжение u на фоне гауссовского шума, где дисперсия по выборочному значению x известна. При этом плотность выборки предстает в виде:

$$- \quad \text{—} \quad \text{—} \; . \tag{11}$$

Далее пусть зафиксировано некоторое выборочное значение x. Требуется исследовать, какая информация о параметре u будет получена, учитывая, что плотность вероятности задана соотношением (11). Чтобы решить эту задачу, нам поможет логарифмическая функция правдоподобия . Данная функция получается, если прологарифмировать выражение (11).

$$- \quad \text{—} \quad \text{—}$$

и отбросить слагаемое, которое не зависит от u,

$$\text{—} \; . \tag{12}$$

На Рисунке 2 изображена логарифмическая функция правдоподобия при =1. У этой функции существует максимум при u=x. Так как для нормального закона распределения P(|x-u|≤σ) = 0.68, то при = 1 это приводит к выполнения неравенства: x-1<u<x+1 с доверительной вероятностью 0.68.

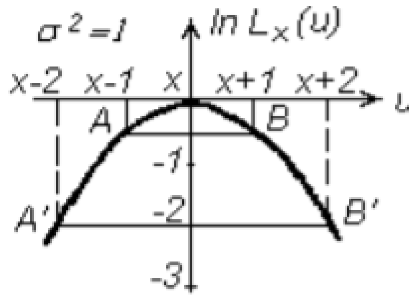

Рисунок 2 – Логарифмическая функция правдоподобия.

Анализируя результат, можно прийти к выводу, что известный вид плотности вероятности демонстрирует, что искомый параметр u расположен на интервале x-1 ≤ u ≤ x+1 с вероятностью 0.68. Возвращаясь к рисунку 1, это отрезок прямой AB, который соединяет ветви параболы на уровне

–. А отрезок A`B`, который проведен на уровне , характеризует доверительный интервал (x-2 < u < x+2). В этот интервал параметр u попадает с вероятностью 0.95. В качестве оценочного значения параметра u можно взять и само значение , являющееся серединой доверительного интервала, что соответствует максимуму функции правдоподобия.

Таким образом, анализируя логарифмическую функцию правдоподобия для среднего значения гауссовского закона распределения, можно сделать вывод, что выборочные данные объединяются в выборочное среднее, которое является значением параметра u. При данном параметре достигается максимум логарифмической функции правдоподобия, поэтому его можно принимать за оценку параметра – . Также, выборочное среднее компактнее группируется вблизи истинного значения параметра u при увеличении объема выборки. Об этом свидетельствует обострение функции правдоподобия. Так как выборочное среднее имеет одинаковое значение, по сравнению с средним значением самого параметра, то оценка – является несмещенной. Далее, оценка является состоятельной, так как разброс выборочного среднего уменьшается с увеличением объема выборки.

Полученные на основе анализа функции правдоподобия, «хорошие» свойства оценки среднего значения нормального закона распределения – дают возможность перейти к формулировке общего метода нахождения оценки максимального правдоподобия. Суть метода заключается в том, что наблюдаемую выборку подставляем в выражение для условной плотности вероятности W(X/). Затем после подстановки условная плотность вероятности W(X/) рассматривается как функция параметра . На этом этапе обычно вводят специальное обозначение такой функции: , которая называется функцией правдоподобия. Далее находится значение параметра , при котором данная функция достигает максимума. Такое значение параметра принимается за оценку.

. (13)

Далее опишем свойства оценок максимального правдоподобия.

Оценки максимального правдоподобия имеют несколько полезных свойств. Во-первых, они являются состоятельными. Это означает, что при увеличении объема выборки значения оценки объединяются компактнее в окрестности значения оцениваемого параметра. Во-вторых, если брать общий случай, то такие оценки являются асимптотически несмещенными, то есть при увеличении объема выборки смещение оценки стремится к нулю. Кроме того, оценки максимального правдоподобия являются асимптотически эффективными. Данное свойство означает, что при увеличении объема выборки среднеквадратическая ошибка оценки стремится к минимально возможной.

Также, описываемые оценки имеют свойство инвариативности к монотонным преобразованиям. Смысл этого свойства заключается в том, что возможно найти максимально правдоподобную оценку какого угодно параметра распределения W(X/), который связан с оцениваемым параметром монотонным преобразованием =g( ). При этом максимально правдоподобная оценка параметра определяется соотношением , где – это обратное преобразование. При увеличении объема выборки распределение оценки максимального правдоподобия приближается к нормальному закону.

1.4.2 Медиана абсолютного отклонения

Метод Median Absolute Deviation (MAD, Медиана абсолютного отклонения) [7] является наиболее часто используемых методов нахождения оценки параметров. На Рисунке 3 представлена схема действий, с помощью которой получают значения оценок математического ожидания и среднеквадратичного отклонения распределения по выборке реализаций в соответствии с данным методом.

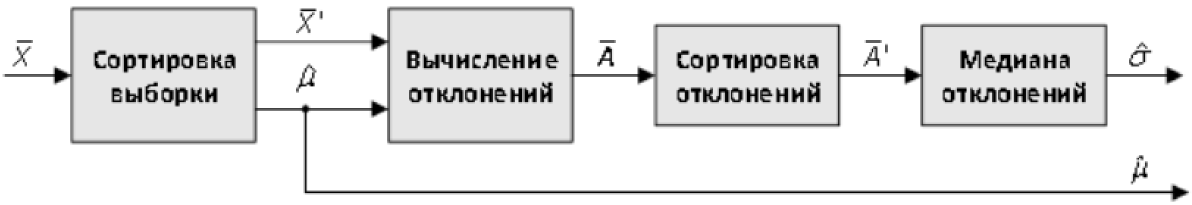

Рисунок 3 – Схема действий метода MAD

В рамках данного метода нахождение оценок начинается с сортировки поступающих отсчетов из . После сортировки формируется упорядоченный набор:

, (14)

где n – объем и исходной выборки , и сформированной выборки .

Чтобы получить оценку математического ожидания используется медиана [8], которая является центральным элементом ранжированного множества , если n – нечетное число. Если n – четное число, медиана – есть полусумма двух ее центральных элементов:

_ (15)

_ __ __

где _, __, __ – это элементы из множества .

Далее работа метода направлена на нахождение устойчивой оценки среднеквадратичного отклонения . Для начала формируем множество абсолютных отклонений элементов выборки от полученного значения математического ожидания :

. (16)

После чего из полученных отклонений формируем ранжированный по возрастанию или убыванию ряд:

(17)

и конечное значение оценки получается в соответствии с выражением:

, (18)

где коэффициент 1.483 берется для коррекции смещения , где E – случайная величина.

В итоге, используя (14) - (18), нахождение требующихся оценок параметров распределения по наблюдаемому множеству элементов выборки можно представить в виде:

(19)

К плюсам описываемого метода можно отнести устойчивость получаемых оценок к засорению с обеих сторон. Кроме того, к важному достоинству можно отнести простоту реализации, а также высокую надежность метода. Вычислительные затруднения составляют процессы нахождения медиан, для которых есть как разные способы аппаратного построения [9, 10], так и большое количество алгоритмов [11].

К недостаткам метода относится низкая эффективность получаемых оценок. Однако, существуют методы повышения ее эффективности [12], но они усложняют метод, делая их трудно применимыми для обработки данных в реальном времени в составе автоматизированных систем обнаружения.

1.5 Выбор среды разработки

Ввиду того, что моделирование случайного процесса требует генерации псевдослучайных величин, целесообразно использовать среду разработки, в которой встроен специальный датчик генерации этих величин. Именно поэтому в качестве среды разработки будем использовать Matlab. Помимо этого, в данной среде удобно формировать, например, корреляционную матрицу и существуют полезные функции, вроде y = filter(a,b,x), которая фильтрует сигнал, заданный в виде одномерного массива x с параметрами a и b. Также, в Matlab можно строить графики, которые наглядно показывают реализацию случайного процесса, выборочных функций случайного процесса или формирование ковариационной функции случайного процесса. Все это делает среду Matlab наиболее приемлемым вариантом для реализации поставленной задачи.

1.6 Обзор современных работ, использующих формирующие фильтры

В настоящее время существует ряд работ, в которых исследуется процесс моделирования случайных процессов методом формирующего фильтра. Рассмотрим некоторые такие работы.

В 2013 году вышла работа к.т.н. доц. Чабунина И.С. под названием «Моделирование случайного микропрофиля дорожной поверхности методом формирующего фильтра» [13]. В данной статье исследуется вопрос моделирования случайного микропрофиля дорожной поверхности при помощи метода формирующего фильтра. Задача моделирования микропрофиля заключалась в следующем: по известным характеристикам случайного процесса необходимо построить вычислительный алгоритм, который позволял бы получать на ЭВМ реализацию случайных процессов q(t) или последовательностей . В случае гауссовского стационарного случайного процесса данная задача – полностью определена. В результате работы были получены дифференциальные уравнения формирующего фильтра для различных функций корреляции и по описанной в статье методике можно смоделировать случайных процесс по любой спектральной плотности в виде дробно-рациональной функции.

Еще одна работа также вышла в свет в 2013 году. Она носит название «Компенсация межсимвольных искажений на основе формирующих фильтров в телекоммуникационных системах», а ее авторами являются Султанов А.Х., Багманов В.Х., Мешков И.К., Мешкова А.Г. и Ишмияров А.А. [14]. В данной работе предлагалось использовать разные виды формирующих фильтров для компенсации межсимвольных искажений и для повышения спектральной эффективности систем связи. Для того, чтобы получить импульсную характеристику предлагалось воспользоваться минимизацией среднеквадратического отклонения с учетом Критериев Найквиста для компенсации межсимвольной интерференции. Для того, чтобы оценить полученные данные использовались глаз-диаграммы. По ним определялись основные параметры сигнала, которые влияли на межсимвольные искажения в телекоммуникационных системах. Также, приводились выводы по возможности использования разных видов фильтров. В результате работы был сделан вывод, что исследование и применение разного рода формирующих фильтров для импульсов является эффективным средством увеличения производительности системы связи с ограниченной полосой пропускания. Влияние межсимвольной интерференции возможно частично подавить с помощью ограничения импульсной характеристики формирующего фильтра. В целом, данные, которые были получены в этой работе, могут повысить эффективность работы системы для снижения влияния межсимвольной интерференции, при этом не увеличивая сложность аппаратной реализации системы связи.

Выводы по разделу

В данном разделе для лучшего понимания задачи была описана предметная область, то есть приведены основные сведения о случайных величинах, случайных процессах и способах их моделирования, таких как: метод скользящей средней, метод формирующего фильтра, рекуррентные моделирующие алгоритмы.

Рассмотрены понятие оценки параметров случайных процессов и методы оценивания параметров. Были сделаны выводы, что оценка является случайной величиной в отличие от самого параметра и такая величина может обладать свойством несмещенности, состоятельности и эффективности. Для того, чтобы найти оценки параметров (характеристик) генеральной совокупности существует ряд методов. К ним относится, например, метод максимального правдоподобия и медиана абсолютного отклонения.

Обоснован выбор среды разработки Matlab для проектирования формирующего фильтра и генерации случайного процесса, так как в данной среде встроен специальный датчик генерации псевдослучайных величин, с помощью которого можно реализовывать случайные процессы, а также дополнительные функции, упрощающие проектирование формирующего фильтра.

2. ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ

2.1 Метод формирующего фильтра

Метод формирующего фильтра используют, когда некоторый сигнал может быть представлен как результат прохождения простого процесса через фильтр [15]. При этом, операторный коэффициент передачи фильтра считается известным и моделирование сводится к решению дифференциального или разностного уравнения, которое описывают работу формирующего фильтра.

Алгоритм моделирования случайного процесса с помощью метода формирующего фильтра основан на нормализации плотности распределения белого шума (t) при его прохождении через формирующий фильтр с комплексным коэффициентом передачи . И тогда спектральная плотность процесса на выходе фильтра имеет вид:

.

Например, пусть существует нормальный случайный процесс, у которого спектральная плотность: ——. Данный случайный процесс является результатом прохождения белого шума, у которого спектральная плотность равняется: , через фильтр с коэффициентом передачи

——.

Как разновидность формирующих фильтров, модели скользящего среднего (MA) – это стационарный процесс, который представляет собой линейную комбинацию последовательных значений белого шума [16].

Модель скользящей средней q-го порядка MA(q) представляет собой модель временного ряда в виде:

,

где – белый шум, а – это параметры модели.

Кроме того, в некоторых случаях в модель добавляют константу, но так как модели скользящей средней в основном используют для моделирования случайных ошибок временных рядов, константу можно считать параметром основной модели.

Сам процесс белого шума можно считать процессом скользящей средней нулевого порядка, т.е. MA(0).

На практике же чаще всего используют процесс скользящего среднего первого порядка, т.е. MA(1):

.

Отметим, что согласно теореме Вольда всякий “регулярный” стационарный процесс может быть представлен как некоторый процесс MA( ) с коэффициентами, сумма модулей которых должна быть конечной. Отсюда получаем, что любой “регулярный” стационарный процесс возможно сколь угодно приблизить некоторым MA(q)-процессом конечного порядка. Однако, такой способ требует достаточно большого порядка модели. Чтобы сократить количество параметров модели, используют модели авторегрессии скользящей средней.

Рассмотрим операторное представление.

Используя лаговый оператор описываемую модель можно представить в виде:

.

Временной ряд является обратимым, если корни характеристического полинома b(z) находятся вне единичного круга в комплексной плоскости, то есть по модулю больше единицы. Такой временной ряд можно представить, как бесконечный авторегрессионный процесс:

.

Условие обратимости для процесса MA(1) означает, что коэффициент b по модулю строго меньше единицы.

Далее рассмотрим автокорреляционную функцию. Для данного процесса функция автоковариации имеет вид:

.

Следовательно, данный процесс является стационарным, дисперсия которого равна:

.

Отсюда получаем, что функция автокорреляции имеет вид:

.

Отметим, что частная функция автокорреляции убывает экспоненциально с возможной осцилляцией. Получаем, что частная автокорреляция затухает, а обычная обнуляется после q. Данное свойство используют при идентификации модели скользящей средней.

2.2 Адаптивная обработка сигналов

Как уже упоминалось ранее во время формирования статистических моделей сигналов наиболее часто применяется принцип гауссовости (линейность, стационарность и нормальность), но данный принцип не всегда может быть выполнен на практике, а от адекватности исследуемой модели зависит качество приема сигнала. Одним из возможных решений такой проблемы является применение адаптивных фильтров [17-19]. Такие фильтры дают возможность системе подстраиваться под статистические параметры входного сигнала, при этом задание какие-либо моделей не предполагается.

Рассмотрим базовую идею адаптивной обработки сигнала. На Рисунке 4 показана общая структура адаптивного фильтра.

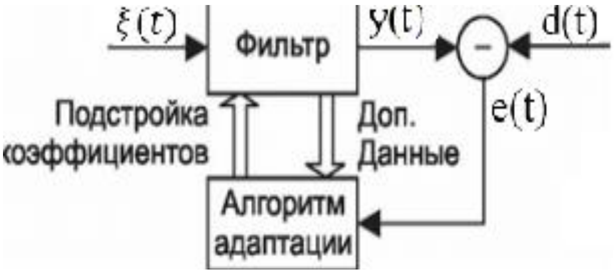

Рисунок 4 – Общая структура адаптивного фильтра.

Входной дискретный сигнал ξ(t) обрабатывается дискретным фильтром, и на выходе получается сигнал y(t). Далее этот выходной сигнал сравнивается с *образцовым* сигналом d(t), а разность между ними образует сигнал ошибки e(t). Задача адаптивных фильтров заключается в том, чтобы минимизировать ошибку воспроизведения образцового сигнала. Для этого блок адаптации после обработки отсчетов анализирует сигнал ошибки и данные, которые поступают из фильтра, используя результаты анализа для подстройки параметров фильтра.

Возможно, с одной стороны, алгоритмы, использующие образцовые сигналы, могут показаться нерациональными с практической точки зрения, так как выходной сигнал должен быть заранее известен, но существуют различные практические задачи, при решении которых образцовых сигнал оказывается известен заранее. Отметим также, что бывают случаи, когда полезным сигналом оказывается не выходной сигнал, а сигнал ошибки.

В качестве фильтра, показанного на Рисунке 4, обычно используют нерекурсивный цифровой фильтр. К основным достоинствам данного фильтра относится то, что нерекурсивный фильтр – устойчив при любых коэффициентах. Однако алгоритм адаптации всегда вносит в систему обратную связь, что влечет к неустойчивости адаптивной системы.

Для рекурсивных фильтров также существуют адаптивные алгоритмы, но во время их разработки возникают проблемы с устойчивостью, в следствие чего такие фильтры не получили широкого распространения.

2.3 Модель авторегрессии скользящей средней

Модель авторегрессии скользящей средней или ARMA (англ. Autoregressive moving-average model) – это математическая модель, которая используется с целью анализа прогнозирования стационарных случайных процессов в статистике [20]. Данная модель является обобщением двух более простых моделей – модель авторегрессии (AR) и, рассмотренная в предыдущем пункте, модель скользящей средней (MA).

Под моделью ARMA(p, q) понимается процесс генерации случайного процесса   (t):

где c – это константа; p u q – целые числа; – белый шум; , – коэффициенты авторегрессии и скользящего среднего, которые являются действительными числами.

Рассмотрим операторное представление модели ARMA.

Введем в рассмотрение лаговый оператор . В таком случае модель можно записать так:

Теперь перенесем авторегрессионную часть в левую часть равенства и получим:

Далее для полиномов левой и правой части введем сокращенное обозначение:

.

Чтобы формируемый процесс был стационарным, корни характеристического многочлена части авторегрессии находились вне единичного круга в комплексной плоскости. Стационарный ARMA-процесс может быть представлен в виде бесконечного процесса модели скользящей средней:

Отсюда можно сделать вывод, что ARMA-процессы можно считать MA-процессами бесконечного порядка с определенными ограничениями на структуру коэффициентов. Они дают возможность описывать процессы сложной структуры, при этом используя малое число параметров. Любые стационарные процессы можно приблизить ARMA-моделью некоторого порядка при помощи меньшего числа порядков.

2.3 Фильтр Калмана

Фильтр Калмана – это эффективный рекурсивный фильтр, который оценивает вектор состояния динамической системы, используя ряд неполных и зашумленных измерений [21-23]. Назван он в честь Рудольфа Калмана. Фильтра Калмана используется во многих областях науки и техники. Благодаря своей простоте и эффективности его можно встретить в GPS-приёмниках, обработчиках показаний датчиков, при реализации систем управления.

Перед тем, как перейти собственно к самому алгоритму Калмана, сформулируем постановку задачи.

Обозначим за величину, которую мы будем измерять, а затем фильтровать. Данная величина может иметь смысл координаты, ускорения, скорости и т.д. Изменение данной величины будет подчиняться следующему закону:

,

где – это некоторый член, который отвечает за контроль системы извне.

В реальных условиях мы не можем учитывать в расчетах разного рода возмущения, которые действуют на моделируемую величину, поэтому реальная величина будет отличной от расчетной. Тогда к правой части уравнения добавится случайная величина :

.

Далее, у нас есть некоторый измеритель, который пытается мерить истинную координату исследуемой величины. Очевидно, он не может измерить ее точно, т.е. мерит он ее с ошибкой , являющаяся, также, случайной величиной. Отсюда с измерителя мы получаем ошибочные данные:

.

Задача заключается в следующем: зная неверные показания измерителя найти хорошее приближение для истинной координаты [24]. Такое хорошее приближение будем обозначать . В итоге получаем уравнения для координаты и показания измерителя:

(20)

(21)

Далее рассмотрим то, что на известно:

- – известная величина, контролирующая эволюцию системы. Ее значение зависит от построенной физической модели.
- – ошибка модели, – ошибка измерителя. Эти величины являются случайными, а их законы распределения зависят от времени (от номера итерации n).
- математическое ожидание ошибок равные нулю, т.е.
  .
- закон распределения случайных величин может быть не известен, однако дисперсии случайны величин и известны.
- кроме того, предполагается, что все случайные величины или ошибки независимы друг от друга, то есть какая ошибка будет в момент времени n не зависит от ошибки в другой момент времени.

Также, отметим, что в нашей задаче фильтрации мы стремимся получить наиболее близкое значение моделируемой величины к реальной координате.

Теперь перейдем к описанию алгоритма Калмана, с помощью которого мы будем проектировать наш фильтр.

Рассуждения будем проводить по индукции. Пусть на n-ом шаге уже найдено отфильтрованное значение с измерителя . Оно приближает истинную координату системы . Исходя из (2) мы знаем изменение неизвестной координаты, поэтому получается, что, еще не получая значения с измерителя, можно предположить, что на n+1 шаге системы эволюционирует по этому закону, и измеритель будем показывать значение, близкое к (предсказание того, что мы ожидаем увидеть от измерителя). Пока более точной информацией мы не владеем, но на шаге n+1 у нас будет неточное показание измерителя

Смысл Калмана заключается в том, что для того, чтобы получить наилучшее приближение к истинной координате , необходимо выбрать среднее значение между показаниями и . Дадим показанию измерителя вес К, а для предсказанного значения будет вес (1-К):

. (22)

Коэффициент К – это коэффициент, который носит название коэффициент Калмана. Он зависит от шага итерации, следовательно, корректнее будем писать :

. (23)

Коэффициент Калмана необходимо выбрать так, чтобы полученное оптимальное значение координаты было наиболее близко к истинной координате . Если измеритель будет иметь высокую точность, то его показанию имеет смысл доверять больше, а значению дадим больше весу, где К близко к единице. В противном случае, если измеритель обладает низкой точностью, то в этом случае необходимо больше доверять теоретическому предсказанию значению .

Для того, чтобы найти точное значение коэффициента Калмана, в общем случае, необходимо минимизировать ошибку:

. (24)

Перепишем данное выражение ошибки с учетом (1) и (2):

$$- \quad +1=1- \quad + \quad - \quad +1.$$

В итоге получаем окончательное выражение для ошибки:

. (25)

Отметим, что ошибка, также, является случайной величиной, которая каждый раз принимает разные значения. Поэтому для расчетов будем минимизировать математическое ожидание от квадрата ошибки:

(26)

Распишем данное выражение:

. (27)

Так как все случайные величины, которые входят в выражение для являются независимыми, а математическое ожидание ошибок измерителя и модели равно нулю: , то получается, что все «перекрестные» члены, также, равны нулю:

. (28)

Кроме того, с учетом , дисперсии ошибок выглядят следующим образом:

,

.

Выражение (27) принимает минимальное значение, когда

. (29)

Теперь подставим в выражение (29) минимизирующее значение коэффициента Калмана:

, (30)

или

(31)

В итоге, мы получили итерационную формулу для вычисления коэффициентов Калмана.

Проследим теперь, как изменяется коэффициента Калмана с шагом n. Можно заметить, что, начиная с определенного шага, значение коэффициента стабилизируется к некоторому значению . Покажем на примере, когда среднеквадратичные ошибки измерителя и модели относятся друг к другу как . В итоге график коэффициентов Калмана в зависимости от шага итерации выглядит следующим образом (Рисунок 5):

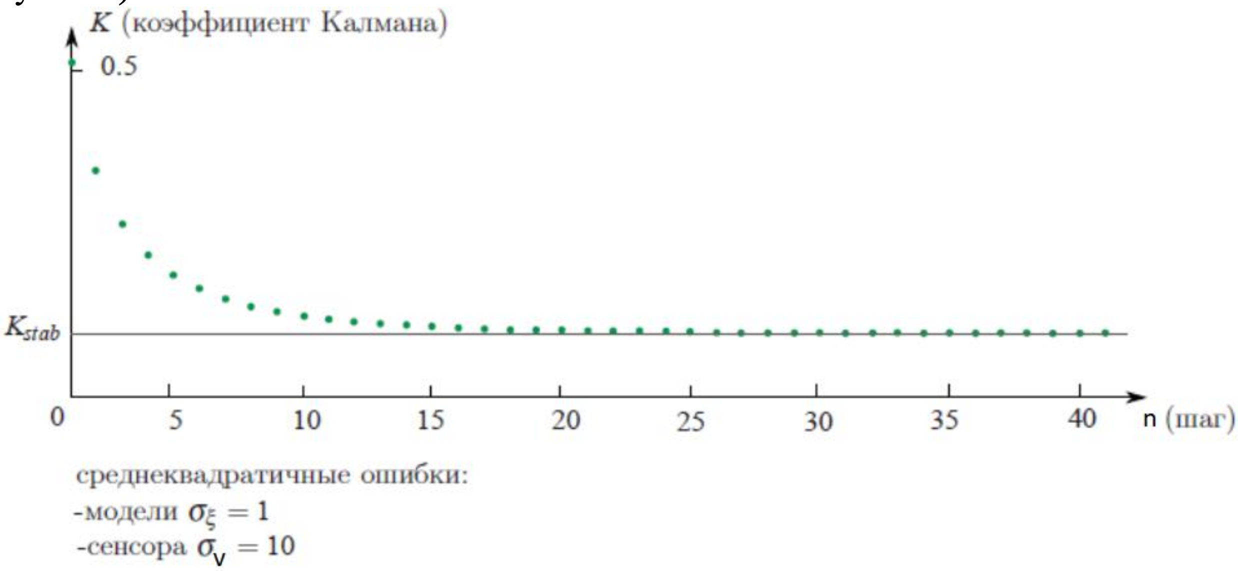

Рисунок 5 – График зависимости коэффициентов Калмана.

Рассмотрим теперь следующий случай, когда ничего не известно о физической модели того, что мы фильтруем. На практике такие случаи нередки. В такой ситуации все незнание модели можно отнести в случайную величину :

(32)

Однако, в этом случае такая система теперь не удовлетворяет условиям, которые были наложены на случайную величину : она содержит всю неизвестную информацию, следовательно, невозможно говорить, что в разные моменты времени ошибки модели независимы друг от друга, а их математическое ожидание равно нулю. Тогда теория калмановской фильтрации, вообще говоря, не применима.

С другой стороны, покажем, что, используя простой метод, можно прийти к результатам, достаточно близким к случаю, когда известна информация о физической составляющей исследуемой модели. Как было показано, коэффициенты Калмана при увеличении номера шага итерации всегда стабилизируется к некоторому значению . Суть заключается в том, что вместо того, чтобы находить по достаточно громоздким формулам коэффициент Калмана , можно считать данный коэффициент константой. Тогда необходимо будет только подбирать этот коэффициент. Это допущение почти ничего не испортит, так как коэффициент Калмана достаточно быстро стабилизируется к константе. Тогда итерационная формула будет иметь следующий вид:

(33)

На Рисунке 6 показан график, на котором отображены отфильтрованные двумя способами данные с измерителя.

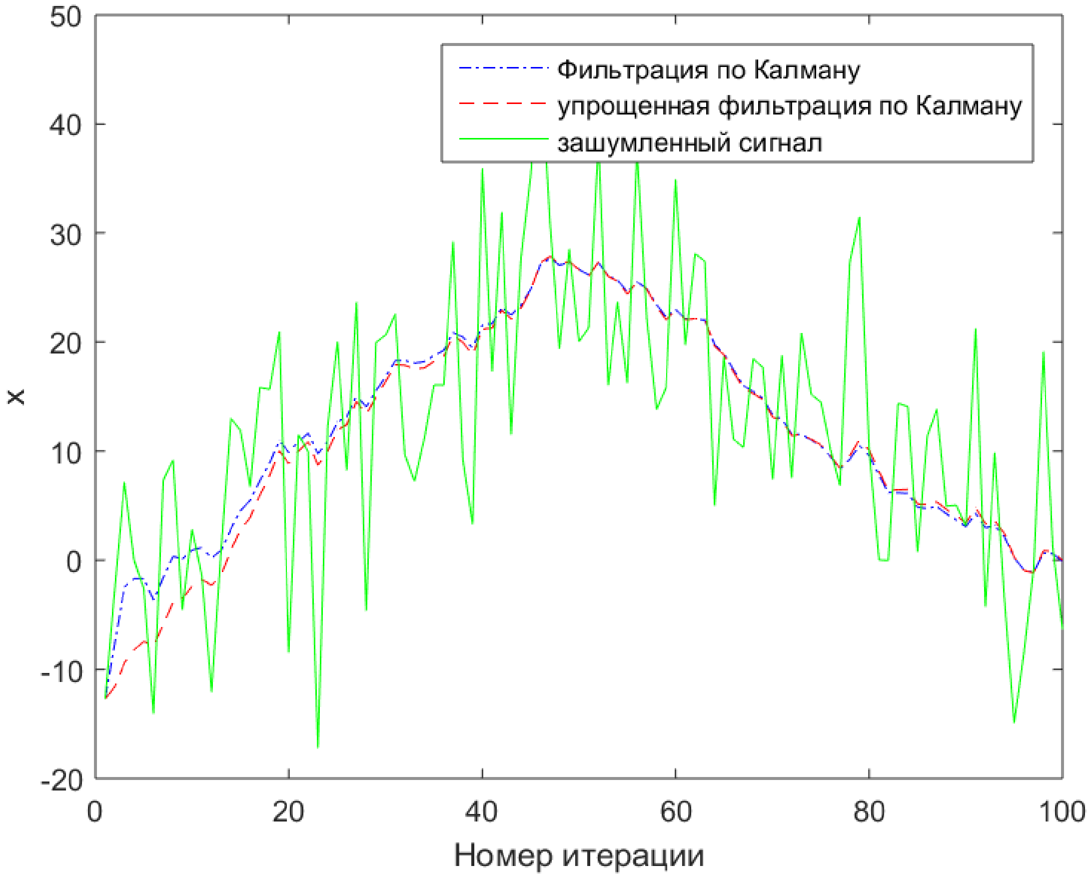

Рисунок 6 – График отфильтрованных двумя способами данных с измерителя.

По графику видно, что методы мало чем отличаются. Отличие наблюдается на первых итерациях, когда коэффициент Калмана еще не стабилизировался.

Выводы по разделу

Подводя итоги, мы увидели, что основная идея фильтра Калмана заключается в том, что необходимо найти такой коэффициент Калмана, чтобы отфильтрованное значение

в среднем меньше всего бы отличалось от истинного значения . Было выяснено, что отфильтрованное значение – это линейная функция от показания измерителя и значения , которое было отфильтровано на предыдущем шаге. Само же значение – это линейная функция от показания измерителя и значения , которое было отфильтровано на предыдущем шаге. Процесс идет, пока цепь полностью не развернется. Таким образом, отфильтрованные значения зависят линейно от всех предыдущих значений измерителя:

.

Отсюда фильтр Калмана носит название линейного фильтра.

Также, показали, что на определенном шаге итерации коэффициент Калмана стабилизируется к некоторой постоянной величине.

Кроме того, показали метод, благодаря которому можно сократить вычисления при проектировании фильтра Калмана, используя коэффициент Калмана, как постоянную величину.

3. РЕАЛИЗАЦИЯ РАБОТЫ ПРОГРАММЫ

3.1 Общие сведения о программе

Наименование программы «Моделирование случайных процессов методом формирующего фильтра». Данная программа написана с использованием языка программирования Matlab в среде GUIDE, которая входит в состав Matlab. Среда GUIDE предназначена для создания приложения с графическим интерфейсом пользователя. Для работы программы необходимо иметь установленный пакет Matlab.

3.2 Функциональное назначение

Программа выполняет фильтрацию сгенерированного случайного процесса с помощью алгоритма Калмана, а также с помощью разработанного упрощенного алгоритма Калмана.

3.3 Описание интерфейса программы

После инициализации программы создаются все рабочие окна. Затем система инициирует цикл, который получает сообщения, где определяется для какого из окон оно предназначено и, исходя из этого, обрабатывается определенным способом. Сообщения поступают до тех пор, пока главное окно не будет закрыто. Основной алгоритм изображен на Рисунке 7, вспомогательный алгоритм – на Рисунке 8.

Основной алгоритм:

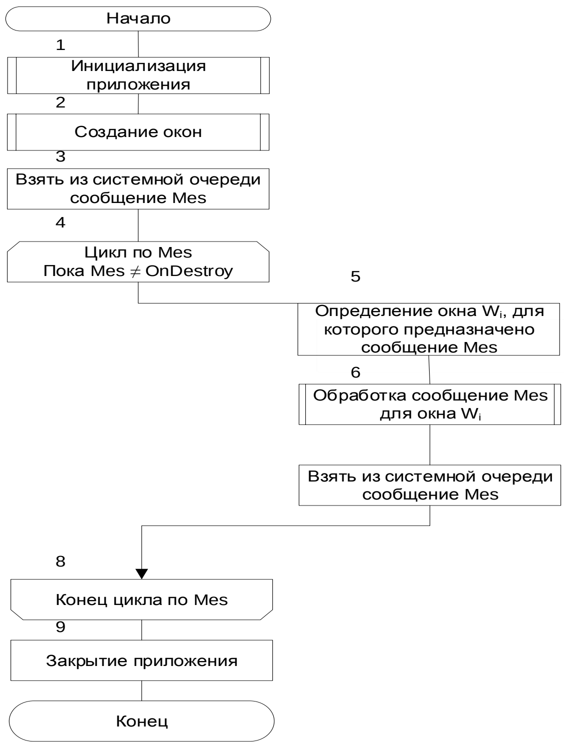

Рисунок 7 – Основной алгоритм.

Вспомогательный алгоритм

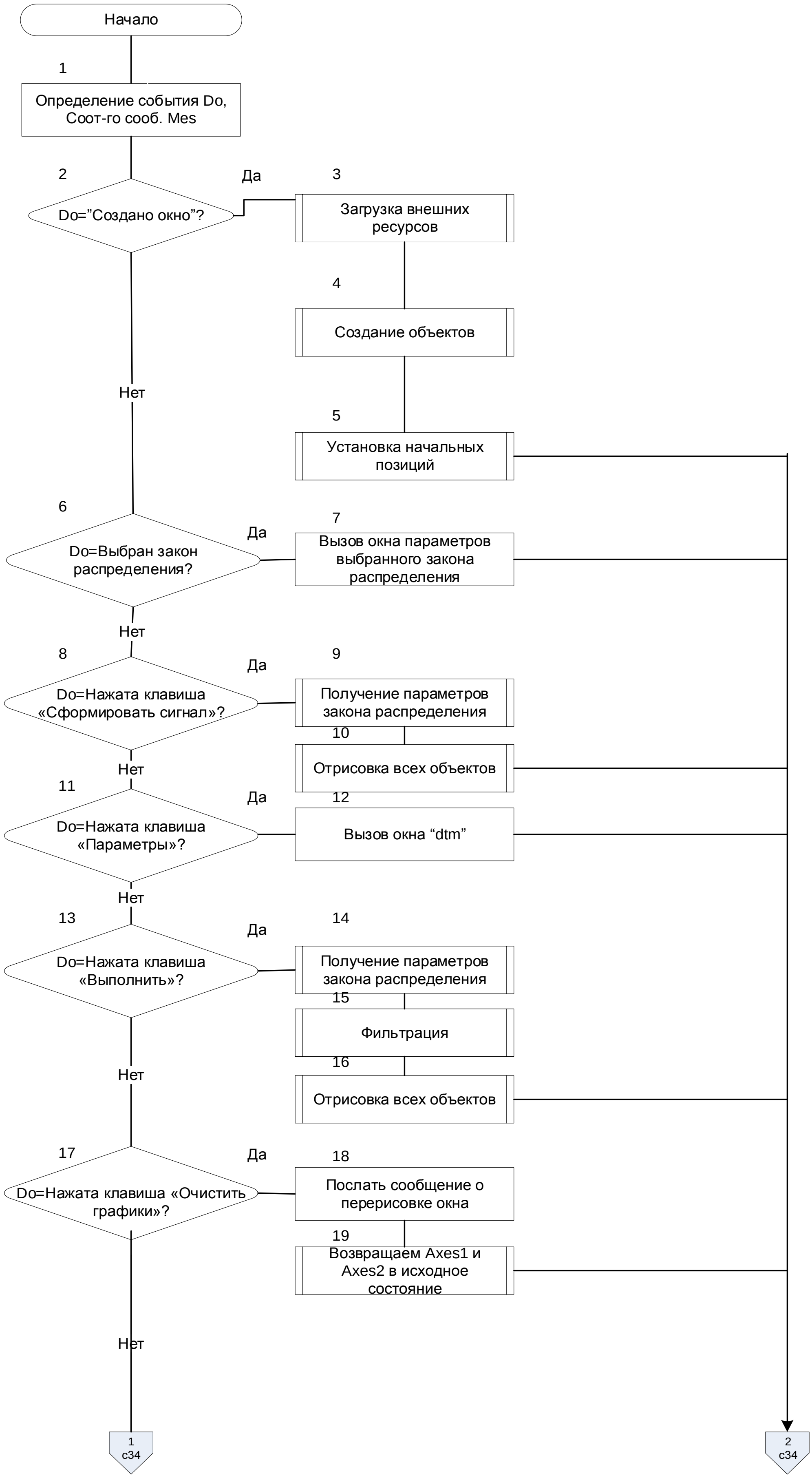

Рисунок 8 – Вспомогательный алгоритм.

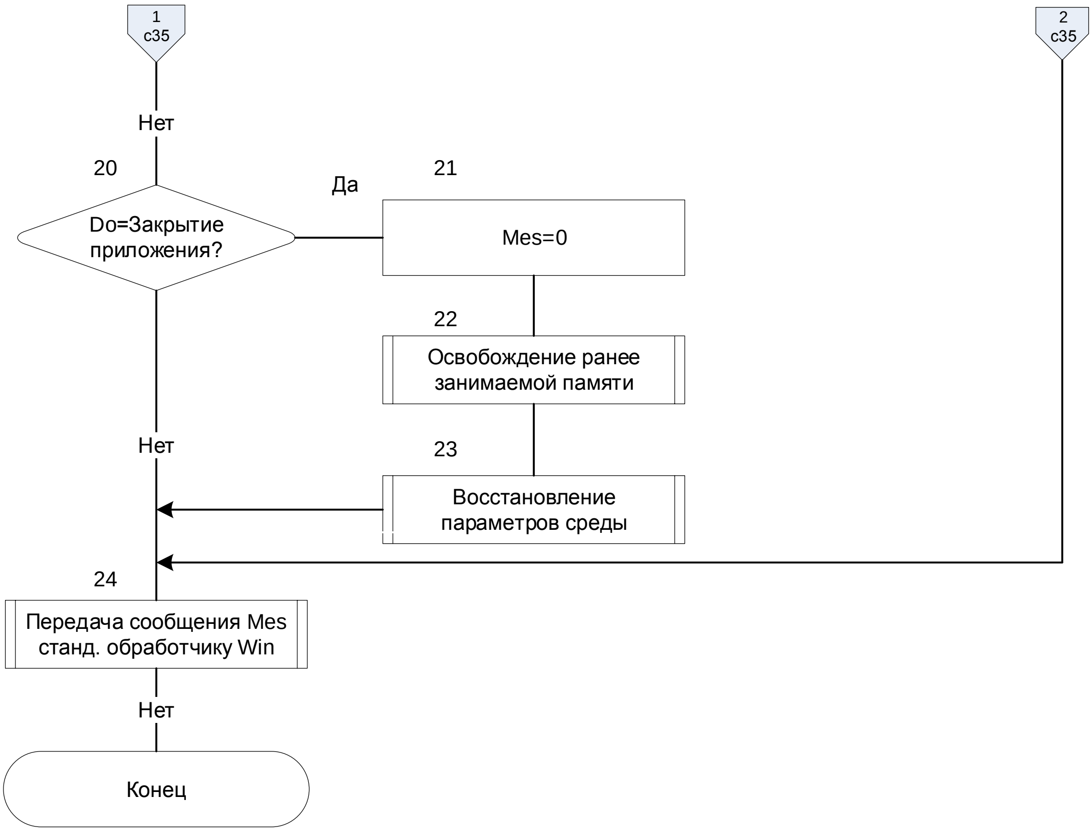

Рисунок 9 – Вспомогательный алгоритм (продолжение).

Рассмотрим главное окно программы, а также дополнительные окна. На Рисунке 10 изображено главное окно программы.

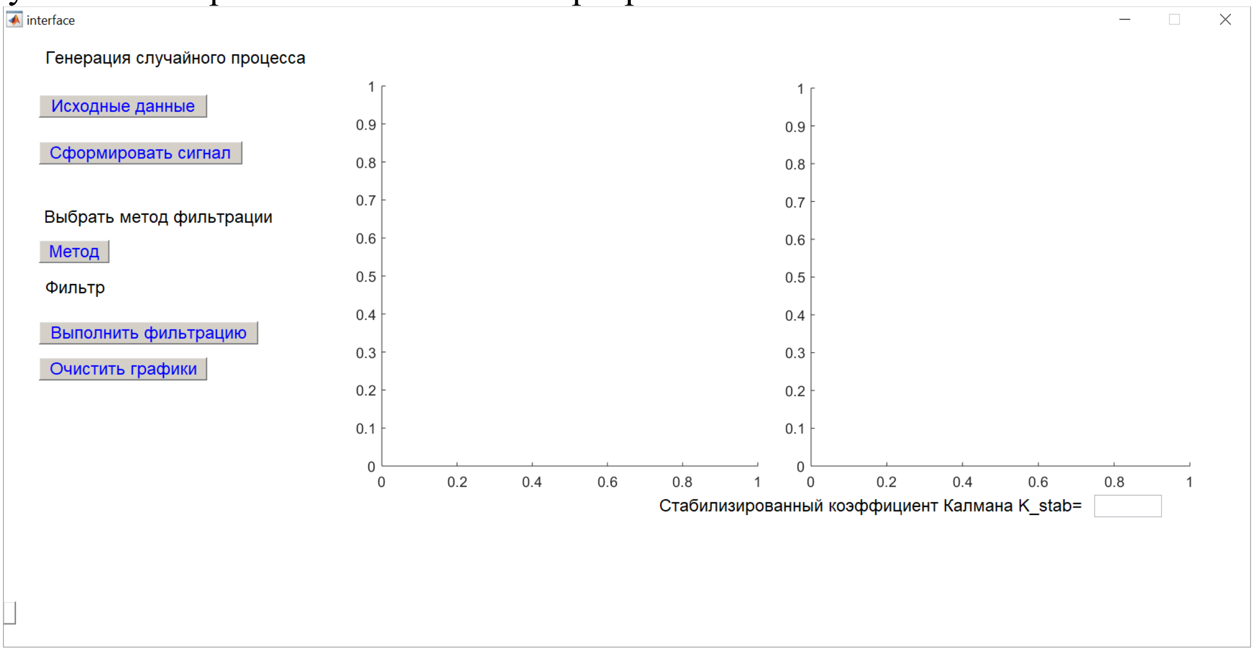

Рисунок 10 – Главное окно программы.

На главном окне расположены следующие кнопки: «Исходные данные», «Сформировать сигнал», «Метод», «Выполнить фильтрацию», «Очистить графики». При нажатии на кнопку «Исходные данные» будет вызвано окно, в котором можно изменить исходные данные (Рисунок 11).

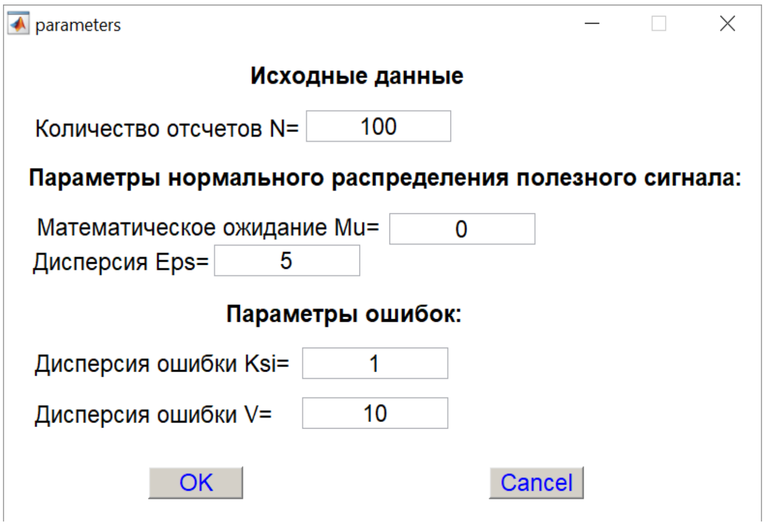

Рисунок 11 – Окно «parameters».

Чтобы сохранить измененные данные и закрыть окно, необходимо нажать на кнопку «ОК». Чтобы просто закрыть окно, не сохраняя данных, необходимо нажать на кнопку «Cancel».

После закрытия окна можно сгенерировать случайный процесс нажатием на кнопку «Сформировать сигнал» (Рисунок 12).

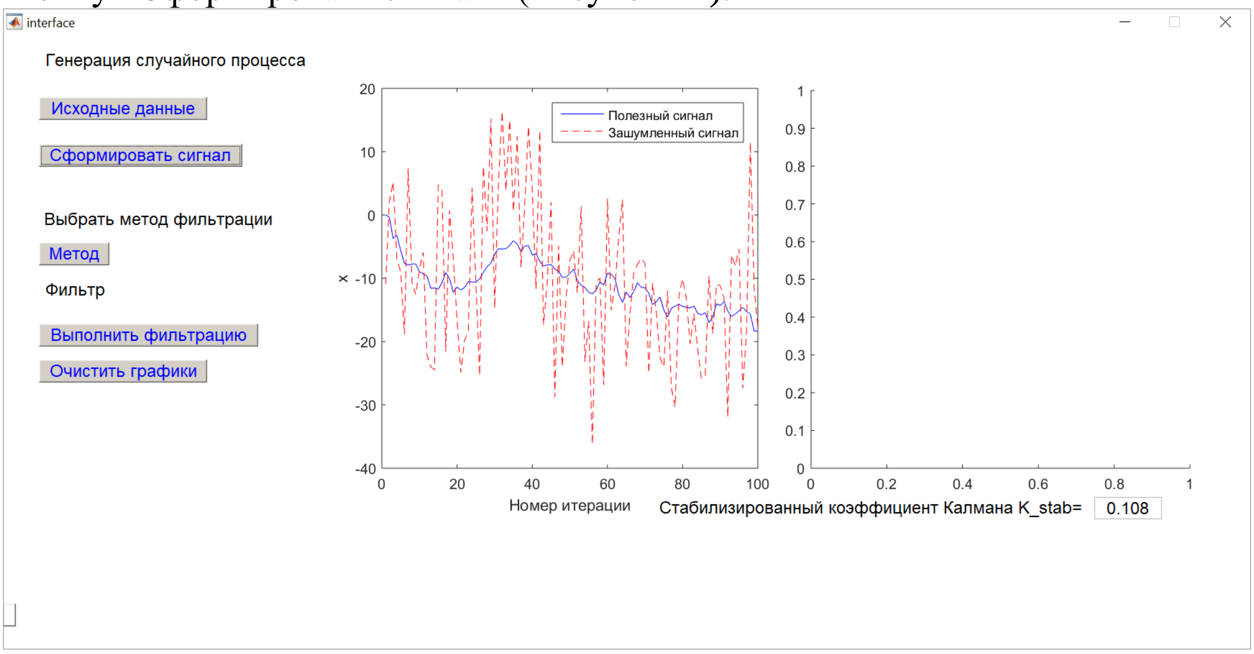

Рисунок 12 – Полученный случайный процесс.

Теперь необходимо найти наилучшее приближение истинного значения. Для этого воспользуемся спроектированным фильтром Калмана. При нажатии на кнопку «Метод» появится окно выбора алгоритма фильтрации (Рисунок 13).

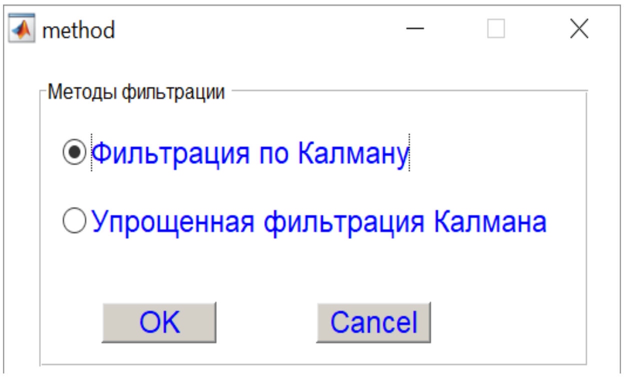

Рисунок 13 – Окно «method».

Выбрав нужный метод, нажимаем на кнопку «ОК», после чего текущее окно будет закрыто.

Далее нужно нажать на кнопку «Выполнить фильтрацию», чтобы провести фильтрацию зашумленного сигнала при помощи алгоритма Калмана, либо при помощи упрощенного алгоритма Калмана (Рисунок 14, Рисунок 15).

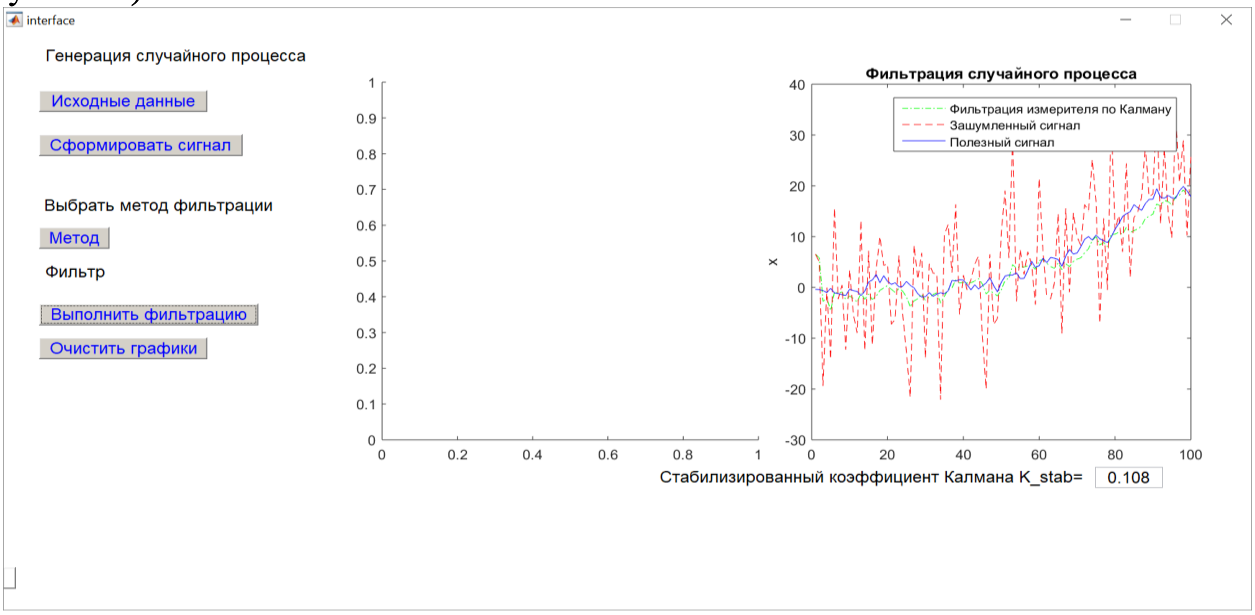

Рисунок 14 – Фильтрация случайного процесса по алгоритму Калмана.

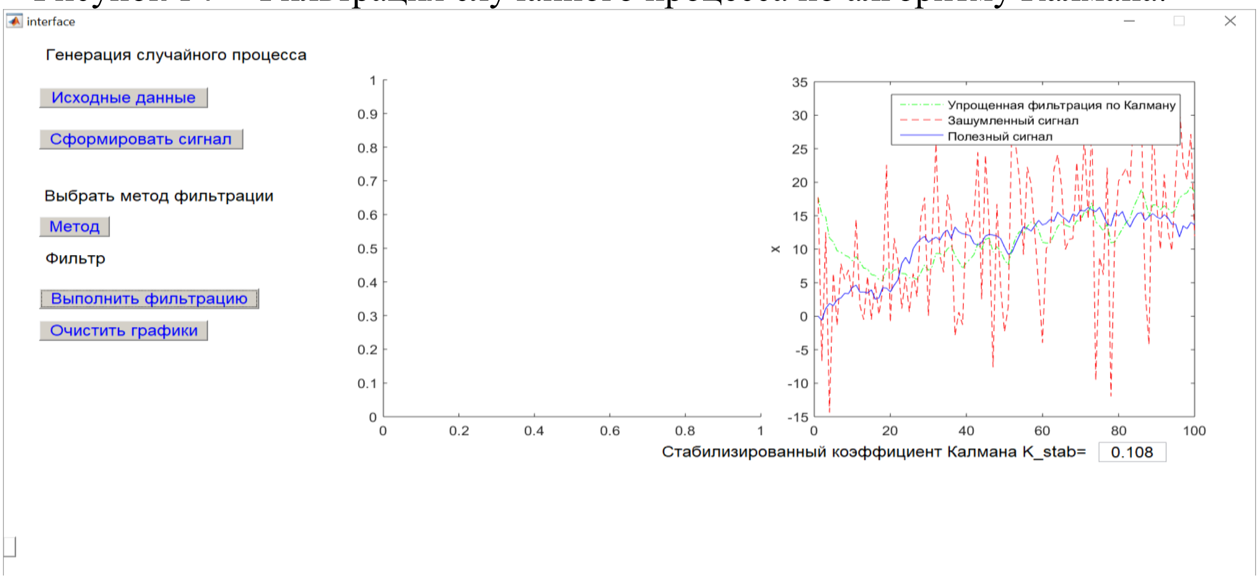

Рисунок 15 – Фильтрация случайного процесса по упрощенному алгоритму Калмана.

Чтобы очистить графики, необходимо нажать на соответствующую кнопку «Очистить графики».

3.5 Алгоритм работы программы

Алгоритм работы программы изображен на Рисунке 16.

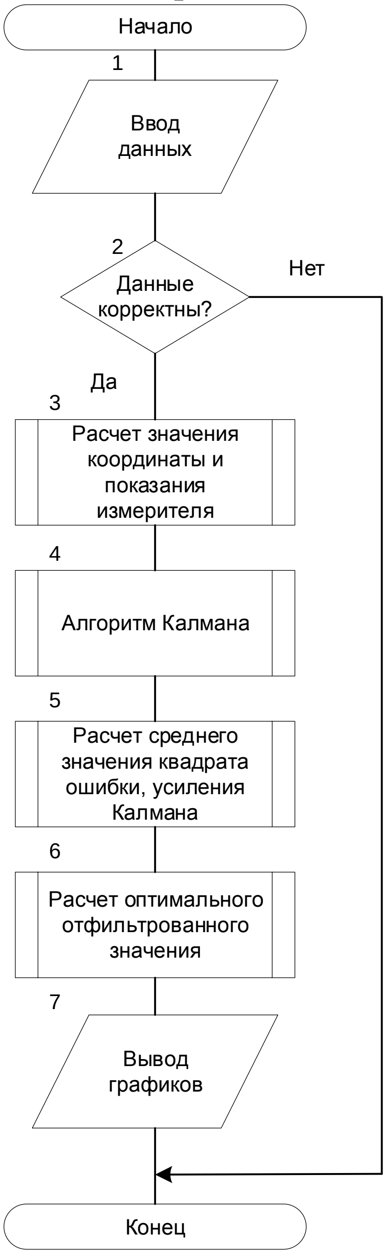

Рисунок 16 – Алгоритм работы программы.

3.6 Описание функций

Программа состоит из одного главного модуля interface.m. Главный модуль содержит функции, каждая из которых решает свою подзадачу.

Далее опишем каждую функцию.

1) function interface – основная функция. Здесь открывается главное окно, записываются указатели на объекты основного окна, инициализируется структура данных par основного окна, а также связываю с событиями Callback кнопок, расположенных на основном окне, подфункции.
2) function btn_method_Callback(src, evt, handles) – подфункция обработки события Callback кнопки «Метод». Здесь происходит открытие окна «method», записываются указатели на объекты окна «method» в соответствующую структуру, происходит получение структуры данных par основного окна, записываются в строку ввода значения параметров, выбирается один из представленных методов. Кроме того, здесь связываются с событием Callback кнопок «OK» и «Cancel» подфункции.
3) function btn_ok_Callback(src,evt,handles,handles_method) – подфункция обработки события Callback кнопки «OK» окна «method». в зависимости от включенного переключателя заполняем поле method структуры par, а также удаляется диалоговое окно «method».
4) function btn_cancel_Callback(src,evt,handles_method) – подфункция обработки события Callback кнопки «Cancel» окна «method». Здесь происходит закрытие диалогового окна «method».
5) function btn_sgnl_Callback(src,evt,handles) – подфункция обработки события Callback кнопки «Сформировать сигнал». Здесь происходит расчет значений исходного и зашумленного сигналов, после чего строится график этих сигналов.
6) function btn_fltr_Callback(src, evt, handles) – подфункция обработки события Callback кнопки «Выполнить фильтрацию». Здесь рассчитываются, в зависимости от выбранного метода, значения среднеквадратичной ошибки, усиление Калмана и оптимальное отфильтрованное значение. Затем после расчета коэффициентов по алгоритму Калмана строятся графики полученных отфильтрованных значений.
7) function pm_par_Callback(src, evt, handles) – подфункция обработки события Callback кнопки «Исходные данные». Здесь происходит открытие окна «parameters», записываются указатели на объекты окна «parameters» в соответствующую структуру, происходит получение структуры данных par основного окна, записываются в строку ввода значения параметров, а также с событиями Callback кнопок «OK» и «Cancel» связывают соответствующие подфункции.
8) function btn_parOK_Callback(src, evt, handles, handles_par) – подфункция обработки события Callback кнопки «OK» окна «parameters». Здесь в строковые переменные записываются содержимое строк ввода окна «parameters», а также удаляется диалоговое окно «parameters».

9) function btn_parCancel_Callback(src, evt, handles_par) – подфункция обработки события Callback кнопки «Cancel» окна «parameters». Здесь происходит закрытие диалогового окна «parameters».
10) function btn_clr_Callback(src, evt, handles) – подфункция обработки события Callback кнопки «Очистить графики». Здесь происходит очистка окон, на которых выводятся графики.

3.7 Используемые технические средства

Необходимые программно-аппаратные ресурсы представлены в таблице 1.

Составляющие	Требования
Программное обеспечение	Matlab 8.0 и выше
Оперативная память, Мб	100
Пространство на жестком диске, Кб	33
Рабочая частота ЦПУ, МГц	600

Таблица 1 – Программно-аппаратная поддержка работы программы.

Выводы по разделу

В данном разделе выполнена разработка интерфейса программы, а также программная реализация алгоритма Калмана. Представлены алгоритмы работы программы и приведено описание интерфейса программы. В результате проведенной работы был получен пользовательский интерфейс программы при помощи GUI Matlab. Данный интерфейс позволяет задавать параметры исходная сигнала, количество шагов итерации и параметры ошибок модели и измерителя, по которым можно генерировать случайный процесс, а затем проводить его фильтрацию с целью получения исходного сигнала. Также, реализована возможность вывода графиков исходного сигнала, зашумленного и отфильтрованного сигналов для того, чтобы наглядно можно было увидеть результат работы фильтра Калмана.

4. ТЕСТИРОВАНИЕ РАБОТЫ ПРОГРАММЫ

4.1 Сравнение двух методов фильтрации

Сформируем случайный процесс, который является суммой полезного сигнала (сгенерированного по нормально закону за нулевым математическим ожиданием и дисперсией равной пяти) и шума (сгенерированного по нормально закону за нулевым математическим ожиданием и дисперсией равной 10). Количество итераций возьмем не больше ста (а именно сто).

На главном окне программы нажмем на кнопку «Исходные данные» и укажем описанные данные (Рисунок 17).

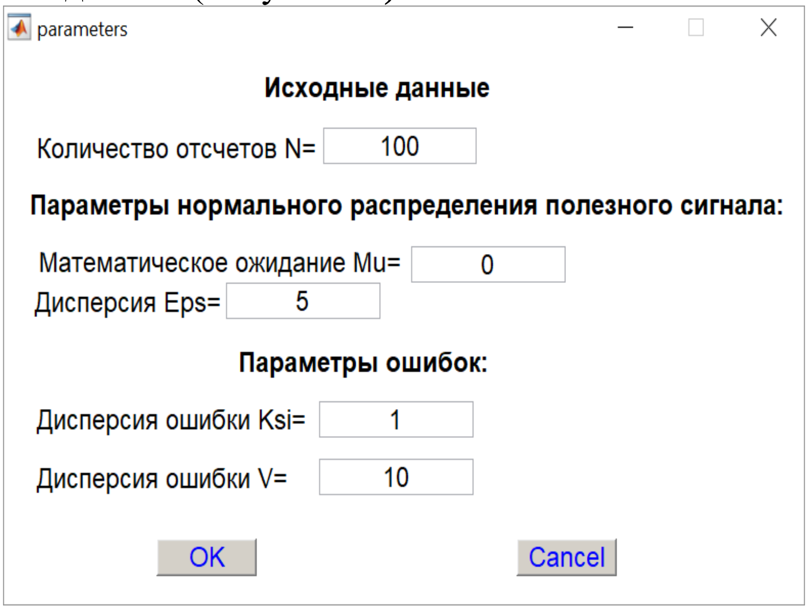

Рисунок 17 – Исходные данные.

Далее построим случайный процесс по имеющимся исходным данным и покажем его на графике (Рисунок 18).

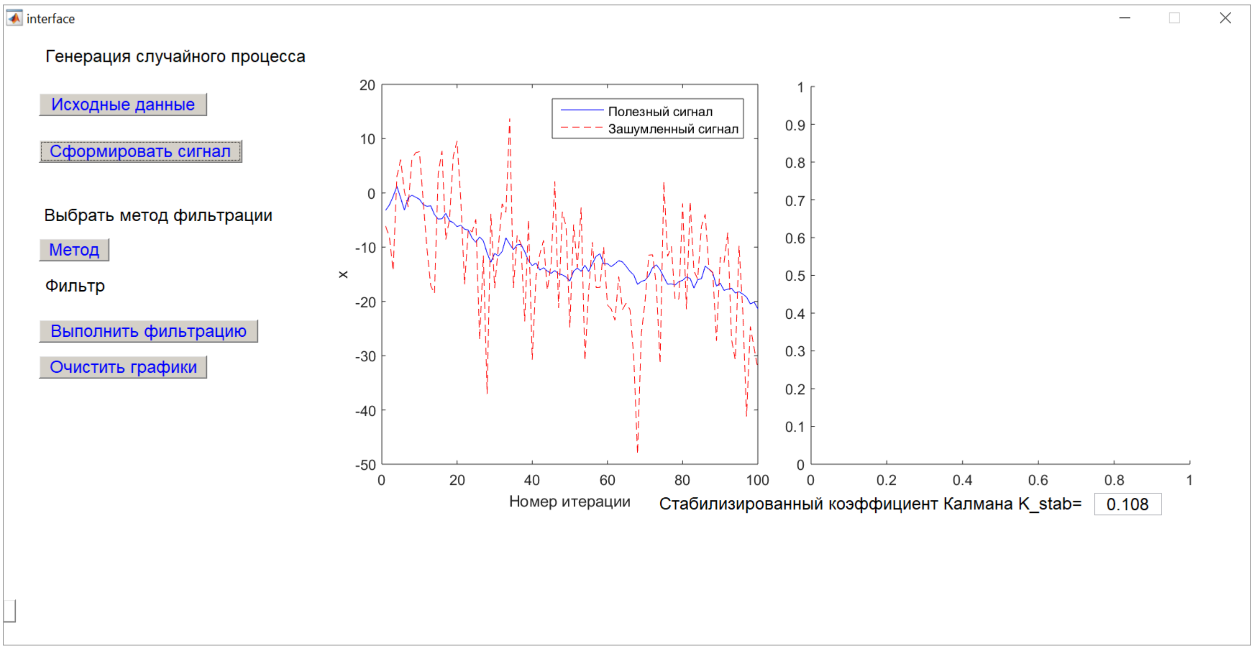

Рисунок 18 – График построенного процесса.

На графике синим цветом обозначен исходный сигнал, а красным – полученный зашумленный сигнал.

Теперь нажмем на кнопку «Метод» и выберем один из двух представленных вариантов алгоритма Калмана (Рисунок 19).

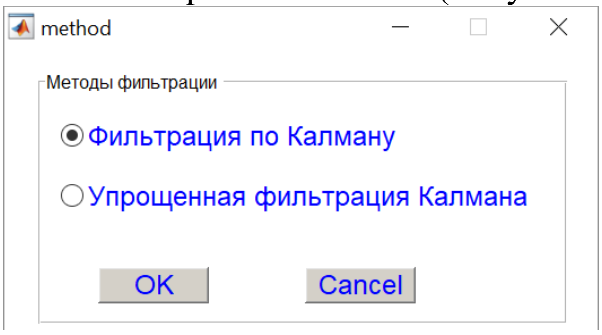

Рисунок 19 – Выбор метода фильтрации.

Сначала рассмотрим вариант описанного алгоритма Калмана, когда физическая составляющая исследуемой модели известна и построим график процесса на выходе построенного фильтра (Рисунок 20).

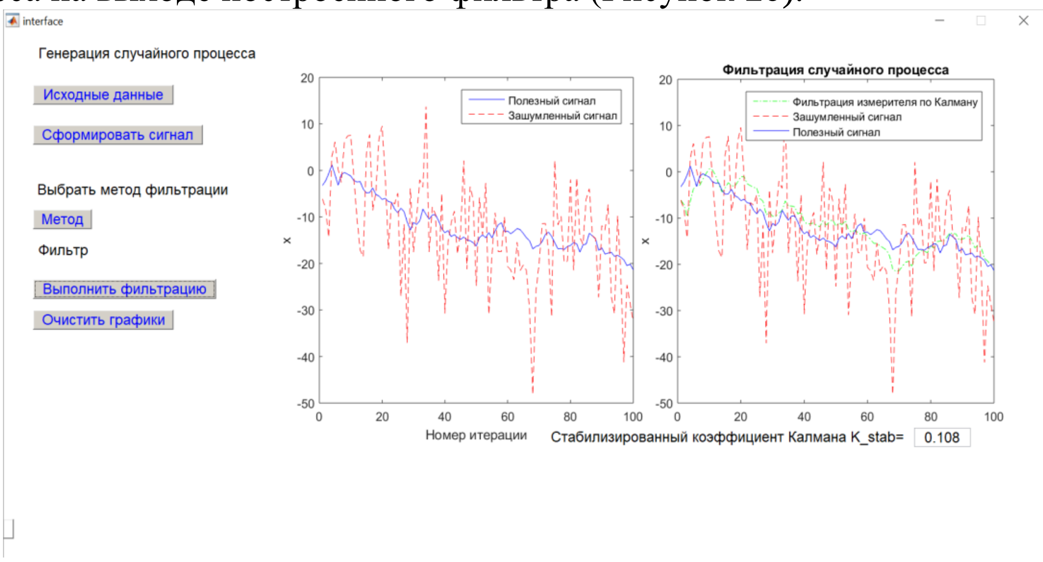

Рисунок 20 – Процесс на выходе фильтра.

Зеленым цветом обозначен отфильтрованный сигнал.

Далее выберем метод, когда коэффициент Калмана считается константой (Рисунок 21).

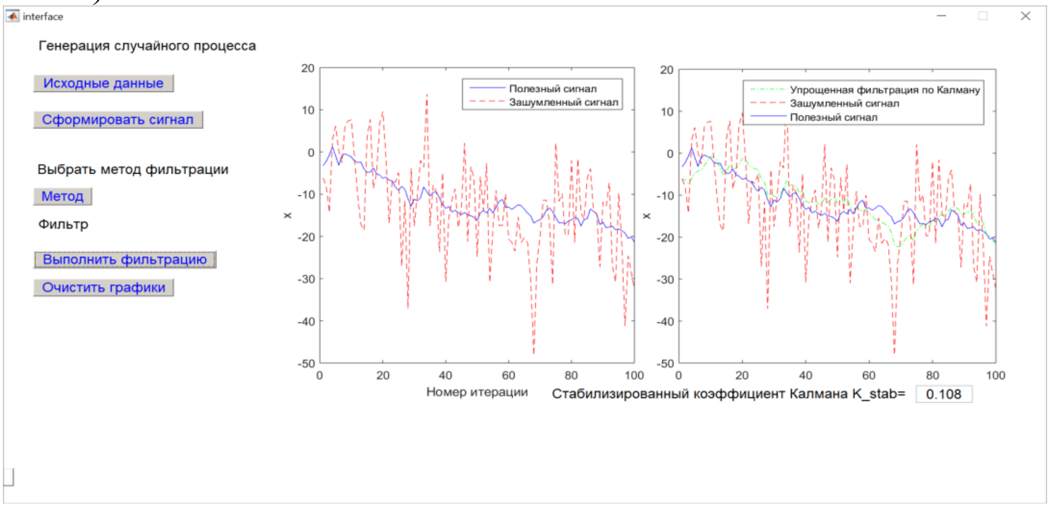

Рисунок 21 – Процесс на выходе фильтра 2.

Выводы по разделу

В данном разделе рассмотрели пример работы программы для фильтрации случайного сигнала двумя методами: Алгоритмом фильтрации по Калману и Упрощенным алгоритмом по Калману.

По графикам видно, что оба метода имеют примерно одинаковые значения. Небольшое отличие имеется лишь на первых итерациях, пока не стабилизировался коэффициент Калмана. Таким образом, можно сделать вывод о том, что спроектированный фильтр достаточно корректно фильтрует зашумленный сигнал как одним методом, так и вторым.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы были рассмотрены методы моделирования случайных процессов: метод формирующего фильтра, метод скользящего суммирования. Было рассмотрено понятие авторегрессии скользящей средней. Были рассмотрены понятия адаптивной обработки сигналов и фильтра Калмана. Можно сделать вывод, что фильтр Калмана является достаточно простым в реализации, но в то же время он является достаточно эффективным, что можно наблюдать во время экспериментальной проверки работы фильтра. Эффективность фильтра заключается в том, что в нем есть возможность задавать априорную информацию (то есть ту, которая была получена ранее рассматриваемого момента времени) о характере системы, связи переменных и на основании этого строить более точную оценку. При этом, даже в простейшем случае (без ввода априорной информации) фильтр Калмана дает достаточно точные результаты.

Кроме того, был рассмотрен алгоритм построения фильтра Калмана в случае, когда физическая составляющая исследуемой модели известна и в случае, когда нет информации о физической составляющей. Спроектирован фильтр Калмана для двух методов: для фильтрации по Калмана и для упрощенной фильтрации по Калману, и спроектирован интерфейс программы.

В результате тестирования разработанная программы показала результаты оптимального отфильтрованного значения достаточно близкие к результатам истинного значения, как методом фильтрации по Калману, так и упрощенным методом фильтрации по Калману. Оптимальность же данного решения достигается, когда ошибки измерителя и модели являются нормальными случайными величинами. В данной задаче фильтра Калмана является квазиоптимальным фильтром, то есть физически конструктивно реализуемым фильтром, параметры которого обеспечивают максимальное отношение сигнал/шум на выходе для данного класса фильтров. При описании упрощенного алгоритма Калмана использовался фильтр Калмана для модели первого порядка.

В дальнейшем планируется разработать решение для усложненной задачи фильтрации сигналов методом формирующего фильтра, а именно разработать алгоритм фильтрации сигнала на фоне небелого шума, когда ошибка не является гауссовой случайной величиной. Решение данной задачи, также, будет проводится при помощи фильтра Калмана, поэтому основная цель – это разработать алгоритм фильтра Калмана для фильтрации сигнала на фоне небелого шума.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лазарев Ю. Моделирование процессов и технических систем в MATLAB, Учебный курс. – СПб.: Питер; Киев: Издательская группа BHV, 2005. – 512 с.
2. Васильев К.К., Служивый М.Н. Математическое моделирование систем связи: учебное пособие. – Ульяновск : УлГТУ, 2008. – 170 с.
3. Теория сигналов и линейных систем. Случайные процессы и функции, https://bourabai.ru/signals/ts171.htm.
4. Джонстон Дж. Эконометрические методы /Пер. с англ. – М.: Статистика, 1980. – 259 с.
5. Доугерти К. Введение в эконометрику /Пер. с англ. – М.: Инфра-М, 1997. – 310 с.
6. Кречетов А.Д., Пашкевич В.П. Оценка параметров радиосигналов. Методические указания к выполнению работ. Государственный комитет РФ по высшему образования. Санкт-Петербургская Государственная Академия аэрокосмического приборостроения, 1996. – 37 с.

7.Осадчий И.С. Метод оценки параметров распределения гауссовского шума для задач обнаружения импульсного сигнала, "ЖУРНАЛ РАДИОЭЛЕКТРОНИКИ" N 4, 2015, Научно-производственное объединение «Лептон», 2015.
8.Хуанг Т.С. Быстрые алгоритмы в цифровой обработке изображений. М.: Радио и связь, 1984. 224 с.
9.Осадчий И.С. Классификация аппаратных структур и сравнительный анализ быстродействующих медианных фильтров // Известия вузов. Электроника, № 1, 2011. С. 57-63.
10.Переверзев А.Л. Аппаратная реализация одномерного медианного фильтра с модульной архитектурой // Известия вузов. Электроника, № 1, 2008. С. 68-73.
11.Кнут Д.Э. Искусство Программирования. Сортировка и Поиск. 2-е изд. Т. 3. М.: Вильямс, 2007. 824 с.
12.Rousseuw P.J., Croux C. Alternatives to the Median Absolute Deviation // Journal of the American Statistical Association, Vol. 88, No. 424, December 1993. pp. 1273-1283.
13.Чабунин И.С. Моделирования случайного микропрофиля дорожной поверхности методом формирующего фильтра, Известия МГТУ «МАМИ» №1(15), 2013, т1, 218-224 с.
14.Султанов А.Х., Багманов В.Х., Мешков И.К., Мешкова А.Г., Ишмияров А.А. Компенсация межсимвольных искажений на основе формирующих фильтров в телекоммуникационных системах, Вестник «УГАТУ» №5(58), 2013, 128-134 с.
15.Прохладин Г.Н. Пособие по дисциплине Моделирование систем и процессов, ч. 1, 2011.
16.Айвазян С.А. Прикладная статистика. Основы эконометрики. Том 2. — М.: Юнити-Дана, 2001. — 432 с.

17. Сергиенко А. Б. Алгоритмы адаптивной фильтрации: особенности реализации в MATLAB / А. Б. Сергиенко // Математика в приложениях. - 2003. - № 1 (1).
18. Уидроу Б., Стирнз С. Д. Адаптивная обработка сигналов. — М.: Радио и связь, 1989. — 440 с.
19. Haykin S. Adaptive Filter Theory, 4th edition.— Prentice Hall, 2002.— 936 p.
20. George E.P. Box, Gwilym M. Jenkins, Gregory C. Reinsel Time Series Analysis: Forecasting and Control (Third ed.). Prentice-Hall, 1994.
21. Неймарк Ю.И., Коган Н.Я., Савельев В.П. Динамические модели теории управления. – М.: Наука, 1985. – 400 с.
22. Greg Welch, Gary Bishop, An Introduction to the Kalman Filter, SIGGRAPH 2001 Course.
23. M.S.Grewal, A.P. Andrews, Kalman Filtering — Theory and Practice Using MATLAB, Wiley, 2001.
24. Браммер К., Зиффлинг Г. Фильтр Калмана-Бьюси . – М.: Наука, 1982. – 200 с.
25. Design and use Kalman filters in MATLAB and Simulink, https://ch.mathworks.com/discovery/kalman-filter.html.
26. Durbin, J., and Koopman, S.J. «Time Series Analysis by State Space Methods». Oxford: Oxford University Press, 2001.
27. Durbin, J., and Koopman, S.J. «A simple and efficient simulation smoother for state space time series analysis» Biometrika vol. 89, issue 3, 2002.
28. Harvey, A.C., and Jaeger, A. “Detrending, stylised facts and the business cycle.” Journal of Applied Econometrics (8), 1993.
29. Harvey, A.C. «Forecasting, Structural Time Series Models and the Kalman Filter». Cambridge: Cambridge University Press, 1989.

ПРИЛОЖЕНИЕ ТЕКСТ ПРОГРАММЫ

```
function interface
% основная функция
% открываем основное окно interface
H = open('interface.fig');
% указатели на объекты основного окна interface записываем в
структуру handles
handles=guihandles(H);
% инициализируем структуру данных par основного окна interface
par.n = 100;
par.mu = 0;
par.sigEps = 5;
par.sigKsi = 1;
par.sigV = 10;
par.method = 'filter';
% сохраняем структуру данных par основного окна interface
guidata(handles.win_main,par)
% с событием Callback кнопки Метод связываем подфункцию
btn_dtm_Callback
set(handles.btn_method, 'Callback', {@btn_method_Callback,
handles})
% с событием Callback кнопки Исходные данные связываем
подфункцию btn_dtm_Callback
```

```
set(handles.btn_par, 'Callback', {@btn_par_Callback, handles})
% с событием Callback кнопки Дискрет связываем подфункцию
btn_dtm_Callback
set(handles.btn_dtm,'Callback',{@btn_dtm_Callback,handles})
% с событием Callback кнопки Сформировать сигнал связываем
подфункцию btn_sgnl_Callback
set(handles.btn_sgnl,'Callback',{@btn_sgnl_Callback,handles})
% с событием Callback кнопки Фильтр сигнал связываем подфункцию
btn_fltr_Callback
set(handles.btn_fltr, 'Callback', {@btn_fltr_Callback, handles})
% с событием Callback кнопки Очистить графики связываем
подфункцию btn_dtm_Callback
set(handles.btn_clr, 'Callback', {@btn_clr_Callback, handles})

function btn_method_Callback(src, evt, handles)
% подфункция обработки события Callback кнопки Метод
% открываем диалоговое окно dtm
h = open('method.fig');
% указатели на объекты окна method записываем в структуру
handles_ method
handles_method = guihandles(h);
% получаем структуру данных par основного окна interface
par = guidata(handles.win_main);
% включаем нужный переключатель в окне method
if isequal(par.method, 'filter')
    set(handles_method.rb_kalm, 'Value', 1)
else
    set(handles_method.rb_ekalm, 'Value', 1)
end
% с событием Callback кнопки OK окна method связываем подфункцию
btn_methodOK_Callback
set(handles_method.btn_ok,'Callback',{@btn_ok_Callback,handles,h
andles_method})
% с событием Callback кнопки Cancel окна method связываем
подфункцию btn_methodCancelCallback
set(handles_method.btn_cancel,'Callback',{@btn_cancel_Callback,h
andles_method})


function btn_ok_Callback(src, evt, handles, handles_method)
par = guidata(handles.win_main);
% в зависимости от включенного переключателя заполняем поле
method структуры par
if get(handles_method.rb_kalm, 'Value')
    % включен переключатель 1
    par.method='filter';
else
    % включен переключатель 2
    par.method='filter1';
end
% сохраняем структуру данных par основного окна interface
guidata(handles.win_main, par)
% удаляем диалоговое окно method
```

```
delete(handles_method.win_method)

function btn_cancel_Callback(src, evt, handles_method)
% подфункция обработки события Callback кнопки Cancel окна
method
% удаляем диалоговое окно method
delete(handles_method.win_method)

function btn_par_Callback(src, evt, handles)
h = open('parameters.fig');
handles_par = guihandles(h);
par = guidata(handles.win_main);
set(handles_par.edt_N,'String',num2str(par.n))
set(handles_par.edt_mu,'String',num2str(par.mu))
set(handles_par.edt_eps,'String',num2str(par.sigEps))
set(handles_par.edt_Ksi,'String',num2str(par.sigKsi))
set(handles_par.edt_V,'String',num2str(par.sigV))
% с событием Callback кнопки OK окна parameters связываем
подфункцию btn_parOK_Callback
set(handles_par.btn_parOK,'Callback',{@btn_parOK_Callback,
handles, handles_par})
% с событием Callback кнопки Cancel окна parameters связываем
подфункцию btn_parCancel_Callback
set(handles_par.btn_parCancel,'Callback',{@btn_parCancel_Callbac
k, handles_par})

function btn_parOK_Callback(src, evt, handles, handles_par)
% записываем в строковую переменную str содержимое строки ввода
окна
% parameters
str=get(handles_par.edt_N,'String');
str1=get(handles_par.edt_mu,'String');
str2=get(handles_par.edt_eps,'String');
str3=get(handles_par.edt_Ksi,'String');
str4=get(handles_par.edt_V,'String');

% получаем структуру данных par основного окна interface
par=guidata(handles.win_main);
% преобразуем str в число и записываем в поле  структуры par
par.n=str2num(str);
par.mu=str2num(str1);
par.sigEps=str2num(str2);
par.sigKsi=str2num(str3);
par.sigV=str2num(str4);
% сохраняем структуру данных par основного окна interface
guidata(handles.win_main,par)
% удаляем диалоговое окно dtm
delete(handles_par.win_par)

function btn_parCancel_Callback(src, evt, handles_par)
delete(handles_par.win_par)

function btn_dtm_Callback(src,evt,handles)
```

```
% подфункция обработки события Callback кнопки Дискрет
% открываем диалоговое окно dtm
h=open('dtm.fig');
% указатели на объекты окна tolerance записываем в структуру
handles_ dtm
handles_dtm=guihandles(h);
% получаем структуру данных par основного окна interface
par=guidata(handles.win_main)
% записываем в строку ввода окна dtm значение точности, которое
хранится в поле dtm структуры par
set(handles_dtm.edt_dtm,'String',num2str(par.dtm))
set(handles_dtm.edt_w0,'String',num2str(par.w0))
set(handles_dtm.edt_e,'String',num2str(par.e))
set(handles_dtm.edt_a,'String',num2str(par.A))
set(handles_dtm.edt_t0,'String',num2str(par.T0))
% с событием Callback кнопки OK окна dtm связываем подфункцию
btn_dtmOK_Callback
set(handles_dtm.btn_dtmOk,'Callback',{@btn_dtmOk_Callback,handle
s,handles_dtm})
% с событием Callback кнопки Cancel окна dtm связываем
подфункцию btn_dtmCancel_Callback
set(handles_dtm.btn_dtmCancel,'Callback',{@btn_dtmCancel_Callbac
k,handles_dtm})

function btn_dtmOk_Callback(src,evt,handles,handles_dtm)
% подфункция обработки события Callback кнопки OK окна dtm

% записываем в строковую переменную str содержимое строки ввода
окна dtm
str=get(handles_dtm.edt_dtm,'String');
str1=get(handles_dtm.edt_w0,'String');
str2=get(handles_dtm.edt_e,'String');
str3=get(handles_dtm.edt_a,'String');
% получаем структуру данных par основного окна interface
par=guidata(handles.win_main);
% преобразуем str в число и записываем в поле dtm структуры par
par.dtm=str2num(str);
par.w0=str2num(str1);
par.e=str2num(str2);
par.A=str2num(str3);
% сохраняем структуру данных par основного окна interface
guidata(handles.win_main,par)
% удаляем диалоговое окно dtm
delete(handles_dtm.win_dtm)

function btn_dtmCancel_Callback(src,evt,handles_dtm)
% подфункция обработки события Callback кнопки Cancel окна dtm

% удаляем диалоговое окно tolerance
delete(handles_dtm.win_dtm)

function btn_sgnl_Callback(src, evt, handles)
global x;
```

```
global y;
% подфункция обработки события Callback кнопки Сформировать
сигнал
axes(handles.axes1);
par = guidata(handles.win_main);
N = par.n; %Число отсчетов
sigmaV = par.sigV; %Дисперсия ошибки v
sigmaKsi = par.sigKsi; %Дисперсия ошибки ?
sigmaEps = par.sigEps;
mu = par.mu;
n = 1:N;
x = n;
x(1) = normrnd(mu, sigmaEps);
y(1) = x(1)+normrnd(0, sigmaV);
for t=1:(N-1)
    x(t+1) = x(t)+normrnd(0, sigmaKsi);
    y(t+1) = x(t+1)+normrnd(0, sigmaV);
end
plot(n, x, 'b', n, y, 'r--')
legend('Полезный сигнал', 'Зашумленный сигнал')
xlabel('Номер итерации')
ylabel('x')

function btn_fltr_Callback(src, evt, handles)
% подфункция обработки события Callback кнопки Выполнить
фильтрацию
global x;
global y;
global a; %Для стабилизированного коэф.Калмана
axes(handles.axes2);
par = guidata(handles.win_main);
N = par.n; %Число отсчетов
sigmaV = par.sigV; %Дисперсия ошибки v
sigmaKsi = par.sigKsi; %Дисперсия ошибки
sigmaEps = par.sigEps;
mu = par.mu;
n = 1:N;
%x = n;

%x(1) = normrnd(mu, sigmaEps);
%y(1) = x(1)+normrnd(0, sigmaV);
%for t=1:(N-1)
 %   x(t+1) = x(t)+normrnd(0, sigmaKsi);
  %  y(t+1) = x(t+1)+normrnd(0, sigmaV);
%end
if isequal(par.method, 'filter')
    xOpt(1) = y(1);%y(1);
    eOpt(1) = sigmaV;
    for t=1:(N-1)
        eOpt(t+1) =
sqrt((sigmaV^2)*(eOpt(t)^2+sigmaKsi^2)/(sigmaV^2+eOpt(t)^2+sigma
Ksi^2));
        K(t+1) = (eOpt(t+1))^2/sigmaV^2;
```

```
        xOpt(t+1) = xOpt(t)*(1-K(t+1))+K(t+1)*y(t+1)
        a = mean(K); %Среднее значение коэффициента Калмана

set(handles.edt_Kstab,'String',num2str(eval(sprintf('%.3f',mean(
K)))));
    end
    plot(n, xOpt, 'g-.', n, y, 'r--', n, x, 'b')
    legend('Фильтрация измерителя по Калману', 'Зашумленный
сигнал', 'Полезный сигнал')
    title('Фильтрация случайного процесса')
    xlabel('Номер итерации')
    ylabel('x')
    %a = par.meanK;
else
    xOpt1(1) = y(1);%y(1);
    eOpt(1) = sigmaV;
    Kstab = a;%0.108; %Стабилизированный коэффициент Калмана
    for t=1:(N-1)
        %eOpt(t+1) =
sqrt((sigmaV^2)*(eOpt(t)^2+sigmaKsi^2)/(sigmaV^2+eOpt(t)^2+sigma
Ksi^2))
        %K(t+1) = (eOpt(t+1))^2/sigmaEps^2
        xOpt1(t+1) = xOpt1(t)*(1-Kstab)+Kstab*y(t+1)
        %mean(K)
    end
    plot(n, xOpt1, 'g-.', n, y, 'r--', n, x, 'b')
    legend('Упрощенная фильтрация по Калману', 'Зашумленный
сигнал', 'Полезный сигнал')
    xlabel('Номер итерации')
    ylabel('x')
end

function btn_clr_Callback(src, evt, handles)
axes(handles.axes1)
cla('reset');
axes(handles.axes2)
cla('reset');
```