

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Южно-Уральский государственный университет» (национальный исследовательский университет)
Высшая школа электроники и компьютерных наук
Кафедра информационно-измерительной техники

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.т.н., профессор

_____ / А.Л. Шестаков /

« ____ » _____ 2019 г.

Алгоритм коррекции динамической погрешности измерительной системы на основе метода прямого поиска

(наименование темы работы (проекта))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ-12.03.01.2019. 322 ПЗ ВКР

(код направления/специальности, год, номер студенческого)

Руководитель, доцент

_____ / Е.В. Юрасова /

« ____ » _____ 2019 г.

Автор

студент группы КЭ – 463

_____ / Е.А. Кошечева /

« ____ » _____ 2019 г.

Нормоконтролер, доцент

_____ / А.С. Волосников /

« ____ » _____ 2019 г.

АННОТАЦИЯ

Кощева Е.А. Алгоритм коррекции динамической погрешности измерительной системы на основе метода прямого поиска. – Челябинск: ЮУрГУ, ВШЭКН, 2019; 78 с. 23 ил., библиогр. список – 28 наим., 1 прил.

В рамках выпускной квалификационной работы был разработан алгоритм, предназначенный для коррекции динамической погрешности. Для этого были изучены существующие способы коррекции динамических погрешностей. В основу алгоритма легла динамическая измерительная система на основе модели датчика.

Алгоритм выявляет оптимальное значение настраиваемого параметра исходя из минимального значения оценки динамической погрешности. В качестве способа оценки минимума функции был выбран метод прямого поиска минимума Фибоначчи.

Для проверки работоспособности алгоритма были проведены программная реализация и цифровое моделирование алгоритма. Разработка программы и моделирование проводились с использованием среды программирования MATLAB 2014. Программа написана на встроенном в среду языке программирования. Пояснительная записка к выпускной квалификационной работе оформлена в текстовом редакторе MS Word 2016.

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>				
<i>Разраб.</i>		Кощева Е.А.			АЛГОРИТМ КОРРЕКЦИИ ДИНАМИЧЕСКОЙ ПОГРЕШНОСТИ ИЗМЕРИТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ МЕТОДА ПРЯМОГО ПОИСКА	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Проз.</i>		Юрасова Е.В.					3	
<i>Н.контр.</i>		Волосников А.С.			ЮУрГУ Кафедра ИНИТ			
<i>Утв.</i>		Лапин А.П.						

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	6
1. АНАЛИТИЧЕСКИЙ ОБЗОР МЕТОДОВ КОРРЕКЦИИ ДИНАМИЧЕСКОЙ ПОГРЕШНОСТИ	8
1.1 Понятие динамических измерений. Основные метрологические динамические характеристики.....	8
1.2 Технология разработки программного обеспечения.....	12
1.3 Современное состояние в области коррекции динамических погрешностей.....	26
Выводы по разделу один	30
2. МЕТОД МОДАЛЬНОГО УПРАВЛЕНИЯ ДИНАМИЧЕСКИМИ ХАРАКТЕРИСТИКАМИ ИЗМЕРИТЕЛЬНЫХ СИСТЕМ	31
2.1 Динамическая модель измерительной системы с дополнительным каналом оценки динамической погрешности.....	31
2.2 Динамическая модель измерительной системы с дополнительным каналом оценки динамической погрешности 2 порядка	42
Выводы по разделу два.....	44
3. ПОДСТРОЙКА ПАРАМЕТРОВ ДИНАМИЧЕСКОЙ ИЗМЕРИТЕЛЬНОЙ СИСТЕМЫ.....	45
3.1 Среда программирования MATLAB	45
3.2 Решение однородных дифференциальных уравнений.....	48
3.3 Выбор метода поиска минимума.....	53
3.4 Подстройка параметров динамической системы с помощью метода Фибоначчи.....	56
Выводы по разделу три.....	58
4. МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКОЙ ИЗМЕРИТЕЛЬНОЙ СИСТЕМЫ.....	59
4.1 Описание программы.....	59
4.2 Результаты моделирования	62

Выводы по разделу четыре	68
ЗАКЛЮЧЕНИЕ.....	69
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	70
ПРИЛОЖЕНИЕ А	73

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						5
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дат.</i>		

ВВЕДЕНИЕ

Измерение – единственный способ получения информации об изучаемом процессе, объекте или состоянии. На данном этапе технологического процесса измерительная техника является неотъемлемой частью промышленности. Использование идеальных приборов, не вносящих искажений в результат измерений, является невозможным. Вследствие чего восстановление сигналов становится все более актуальной проблемой.

Динамическая составляющая полной величины погрешности измерения, обусловленная инерционностью первичного измерительного преобразователя, зачастую оказывает существенно большее влияние на результат измерения, чем статическая. Значение динамической погрешности зачастую очень велико, а потому, даже располагая информацией о нем, сложно оценить величину входного сигнала по выходному. Для восстановления входного сигнала средства измерений и полной динамической характеристики необходима коррекция динамической погрешности преобразования. А потому задача коррекции динамических погрешностей остается актуальной в течении вот уже пятидесяти лет [1]. При это отсутствие значительной априорной информации не позволяет обобщить решение данной задачи, поэтому в большинстве работ рассматриваются только частные случаи, имеющие ограниченную область применения.

Наиболее интересными считаются подходы, имеющие широкую область применения. В настоящие дни существует две научные школы централизованно занимающиеся вопросами коррекции динамической погрешности: Санкт-Петербургская научная школа и научная школа Южно-Уральского государственного университета. Представители первой занимаются разложением неопределенности результатов измерения на отдельные составляющие с целью компенсации некоторых из них. Такой подход позволяет существенно повышает точность измерений, однако содержит довольно сложную математику, применение которой на производстве трудно осуществимо. Представители научной школы

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						6
Изм.	Лист	№ докум.	Подп.	Дат.		

ЮУрГУ все большее внимание уделяют междисциплинарным подходам, в частности, ЮУрГУ рассмотрению возможности применения методов теории автоматического управления для коррекции динамической погрешности. Использование такого подхода в основе алгоритмов, направленных на компенсацию динамической погрешности, существенно повышает точность обработки результатов.

Целью выпускной квалификационной работы является разработка алгоритма уменьшения оценки динамической погрешности средств измерения на основе метода прямого поиска минимума динамической погрешности.

Задачами выпускной квалификационной работы являются проведение аналитического обзора существующих методов погрешности с целью определения достоинств и недостатков существующих методов, осуществление выбора метода поиска минимума погрешности исходя из требований точности и быстродействия, разработка алгоритма коррекции динамической погрешности на основе динамической измерительной системы с модальным управлением, программная реализация разработанного алгоритма, цифровое моделирование динамической измерительной системы и разработанного алгоритма.

						<i>Лист</i>
					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	7
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дат.</i>		

1. АНАЛИТИЧЕСКИЙ ОБЗОР МЕТОДОВ КОРРЕКЦИИ ДИНАМИЧЕСКОЙ ПОГРЕШНОСТИ

1.1 Понятие динамических измерений. Основные метрологические динамические характеристики

Динамические измерения – это измерения изменяющейся по размеру физической величины (ФВ) [2, с. 15]. Измерение относят к динамическим, если динамической погрешностью при данной скорости измерения нельзя пренебречь в рамках решаемой измерительной задачи. Динамической погрешностью называют разность между погрешностью средства измерения в динамическом режиме и статической погрешностью, соответствующей значению величины в данный момент времени.

Инерционные элементы входят в состав любого средства измерения. Наличием таких элементов определяется инерционность всего измерительного устройства. Поэтому мгновенные значения в динамическом режиме выходного сигнала зависят как от мгновенных значений входной величины (входных величин), так и от изменений этой величины (этих величин), то есть от ее (их) производных. Инерционные свойства измерительных устройств определяются динамическими характеристиками.

Динамические характеристики средств измерения определяют зависимость выходного сигнала средств измерения от меняющийся во времени величин, а, именно, параметров входного сигнала внешних влияющих величин. От значений динамических характеристик зависит динамическую погрешность измерения. Другими словами, динамические характеристики позволяют оценить параметры быстродействия средств измерения.

Различают полные и частные динамические характеристики.

С помощью полной динамической характеристики можно однозначно определить изменение выходного сигнала средства измерения при любых

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						8
Изм.	Лист	№ докум.	Подп.	Дат.		

изменения во времени информативных и неинформативных параметров входных сигналов и нагрузки.

Полные динамические характеристики: переходная характеристика, импульсная переходная характеристика, передаточная функция, амплитудно-фазовая характеристика и совокупность амплитудно-частотной и фазово-частотной характеристики.

Переходная характеристика показывает зависимость изменения выходного сигнала $y(t)$ при подаче на него единичного ступенчатого входного измеряемого сигнала $x(t)$. Основные типы переходных характеристик приведены на рисунке 1.

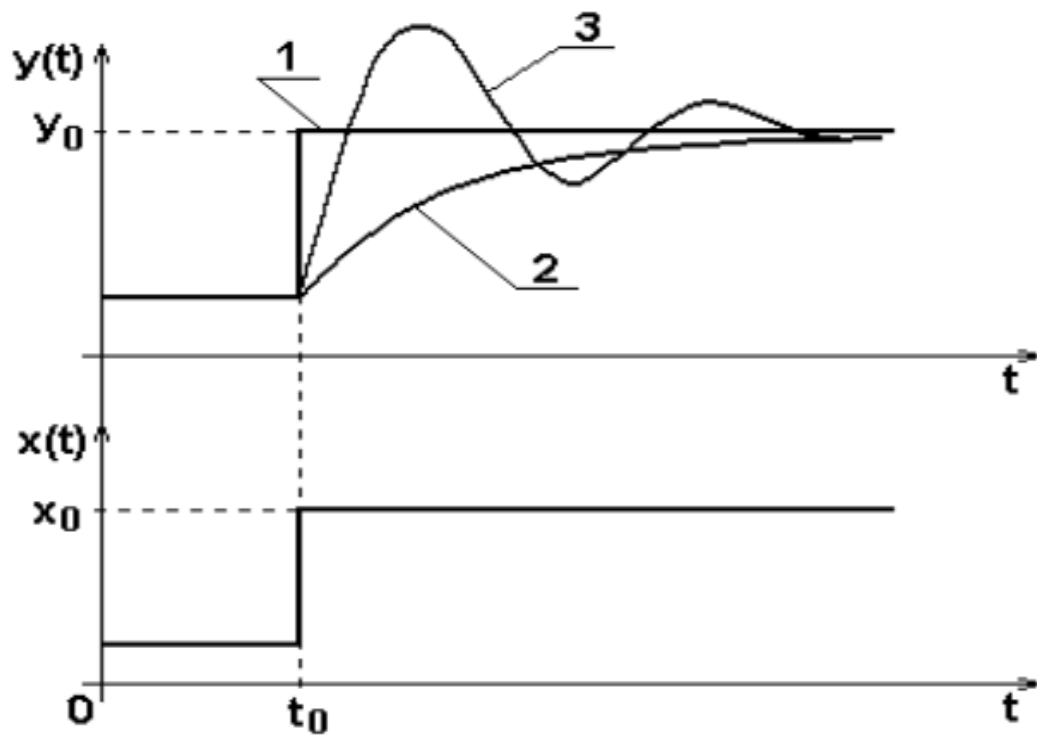


Рисунок 1 – Типы переходных характеристик средств измерения
1 – идеальная (безынерционная) характеристика; 2 – апериодическая характеристика первого порядка; 3 – периодическая (колебательная) характеристика

Импульсной переходной или весовой характеристикой $\omega(t)$ называется функция, которая при нулевых начальных условиях описывает реакцию звена на единичное

импульсное воздействие. Единичным импульсом называется импульс с единичной площадью бесконечно малой длительностью.

Для реализуемости системы в реальном времени необходимо, чтобы отклик на выходе системы возникал не раньше, чем поступающее на вход системы воздействие, вызывающее отклик, то есть $\delta(t) = 0$ и $h(t) = 0$ при $t < 0$.

Передаточная функция (ПФ) $W(p)$ определяется отношением изображений по Лапласу выходного сигнала к входному сигналу при нулевых начальных условиях:

$$W(p) = \frac{Y(p)}{X(p)}, \quad (1)$$

где $Y(p)$ – изображение выходного сигнала по Лапласу;

$X(p)$ – изображение входного сигнала по Лапласу;

p – переменная комплексная величина.

С помощью частотных характеристик описываются передаточные свойства звеньев и систем в режиме установившихся гармонических колебаний, вызванных внешним гармоническим воздействием.

Амплитудно-фазовой характеристикой $W(j\omega)$ называется представление частотного отклика линейной стационарной динамической системы в виде графика в комплексных координатах.

Ее получают формальной заменой оператора p на $j\omega$ в передаточной функции $W(p)$. При этом $j = \sqrt{-1}$ – мнимая единица; $\omega = 2\pi f$ – круговая частота; f – частота рассматриваемых сигналов.

Запишем амплитудно-фазовую характеристику в алгебраической форме:

$$W(j\omega) = G(\omega) + jQ(\omega), \quad (2)$$

где $G(\omega)$ – вещественная часть;

$Q(\omega)$ – мнимая часть.

						Лист
					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	10
Изм.	Лист	№ докум.	Подп.	Дат.		

График амплитудно-фазовой характеристики представлен на рисунке 2.

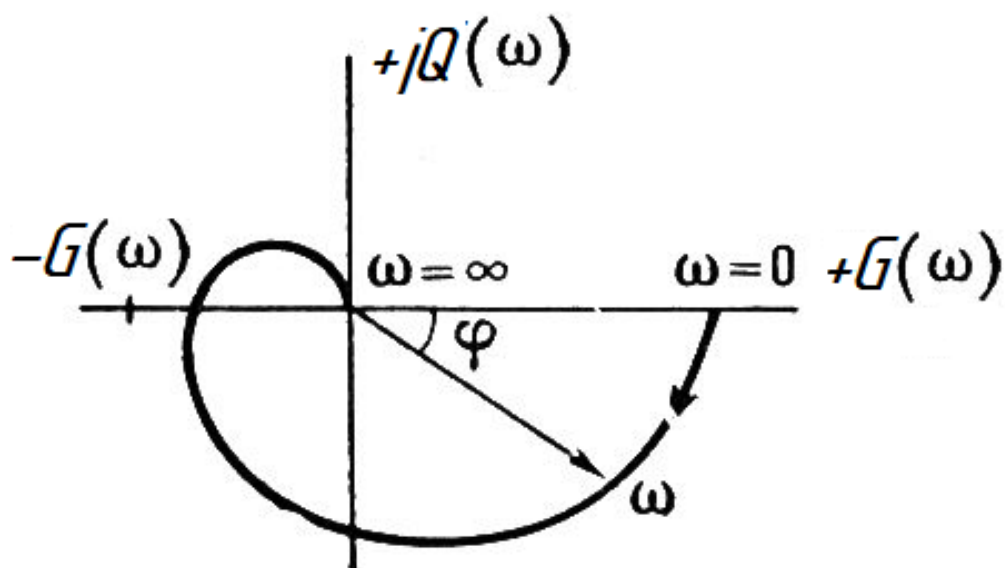


Рисунок 2 – Типичный вид амплитудно-фазовой характеристики СИ

Помимо алгебраической формы записи амплитудно-фазовой характеристики существует также показательная форма записи:

$$W(j\omega) = A(\omega) \cdot e^{j\varphi(\omega)}, \quad (3)$$

где $A(\omega)$ – амплитудно-частотная характеристика;

$\varphi(\omega)$ – фазово-частотная характеристика.

Чаще всего амплитудно-фазовые характеристики используются при анализе устойчивости систем.

Алгебраическая и показательная формы амплитудно-фазовой характеристики записи связаны между собой с помощью соотношений:

$$A(\omega) = \sqrt{[G(\omega)]^2 + [Q(\omega)]^2}, \quad (4)$$

$$\varphi(\omega) = \arctan \frac{Q(\omega)}{G(\omega)}. \quad (5)$$

									Лист
									11
Изм.	Лист	№ докум.	Подп.	Дат.	ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР				

По уравнению (4) строится амплитудно-частотная характеристика, пример которой представлен на рисунке 3, а, а по уравнению (5) – фазово-частотная характеристика (рисунок 3, б).

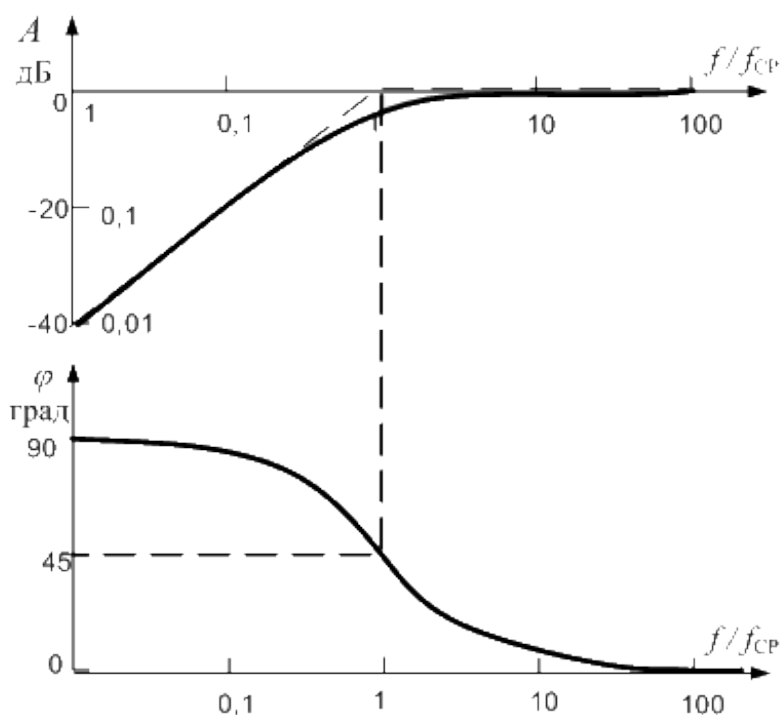


Рисунок 3 – Графики амплитудно- частотной (а) и фазово-частотной (б) характеристик

Частными динамическими характеристиками называют параметры или функционалы полных характеристик, например, время реакции или коэффициент демпфирования.

1.2 Технология разработки программного обеспечения

На данном этапе технического развития во все области производства широко внедряются информационные технологии, что способствует внедрению большого числа программных средств (ПС) различного функционального назначения.

Согласно ГОСТ 19781-90 «Обеспечение систем обработки информации программное. Термины и определения» программное обеспечение (ПО) – это

									Лист
									12
Изм.	Лист	№ докум.	Подп.	Дат.	ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР				

совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ [3].

Объем и сложность используемого ПО постоянно возрастают. Поэтому в мире активно ведутся исследования по усовершенствованию подходов к разработке ПС.

Технология разработки программного обеспечения – это совокупность процессов и методов создания программного продукта; это система инженерных принципов для создания экономичного ПО с заданными характеристиками качества [4, с. 10].

Выделяют шесть этапов разработки программного обеспечения, которые входят в жизненный цикл ПО:

1. Анализ требований, предъявляемых к системе.
2. Определение спецификаций.
3. Проектирование.
4. Кодирование.
5. Тестирование.
6. Эксплуатация и сопровождение.

Соотношение затрат этапов разработки представлено на рисунке 4.

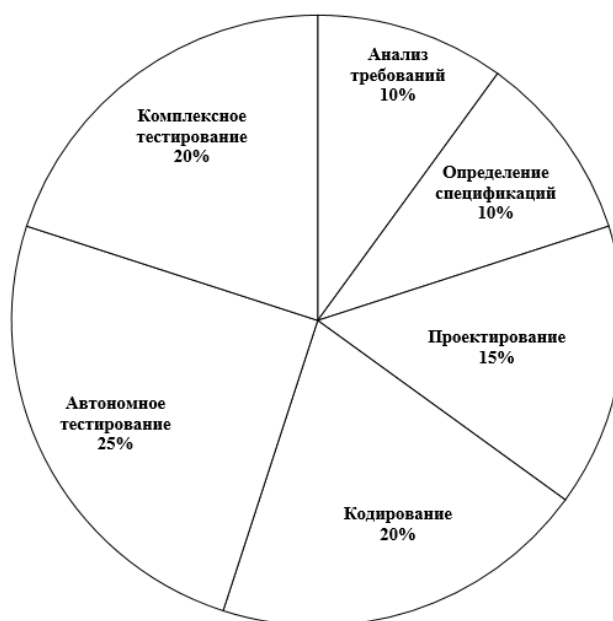


Рисунок 4 – Среднее распределение затрат по этапам разработки

На этапе анализа требований формируется перечень требований, предъявляемых к будущей программе. Этот этап наименее формализован.

Для составления исходных описаний часто используется метод структурного анализа, который заключается в декомпозиции (то есть разделении) исходного объекта на составные части. При этом необходимо различать обязательные требования и те, которые носят рекомендательный характер. Составление развернутого технического задания является завершающей частью этапа анализа требований.

Определение спецификаций характеризуется детальным описанием функций, реализуемых устройством, определением структуры входных и выходных данных, методов и средств их размещения, алгоритмов обработки данных.

Для отражения всех этих вопросов используют функциональные спецификации, содержащие основные характеристики системы. Подробное описание характеристик в спецификации позволяет снизить количество ошибок.

В спецификации вносятся данные для тестирования не только отдельных элементов системы, но и системы в целом. В целом, спецификации определяют функции, которые должна выполнять система, не указывая, каким образом это достигается.

На стадии проектирования формируется общая структура вычислительной системы и разрабатываются алгоритмы ее работы. При этом система разбивается на конечные составные модули. Для каждой части системы должен быть сформирован ряд требований. Если спецификации программы определены корректно, появляется возможность описания процесса обработки информации.

Со времени возникновения самого понятия ПО, появилось множество стратегий по его разработке. На данный момент основными стратегиями разработки ПО можно считать: кодирование с устранением ошибок в процессе; каскадную стратегию; инкрементную стратегию; эволюционную стратегию.

Самой простой для понимания является стратегия кодирования с исправлением ошибок. Суть стратегии заключается в том, что после получения требований к ПО,

											Лист
											14
Изм.	Лист	№ докум.	Подп.	Дат.							

переходят непосредственно к кодированию, затем тестированию и возврату на этап кодирования. Данный цикл повторяется, пока требования к ПО не будут выполнены. Стратегия кодирования с устранением ошибок в процессе появилась на самом первом этапе развития вычислительной техники и имеет ряд серьезных недостатков, к которому следует отнести в первую очередь длительность процесса разработки, высокую стоимость из-за необходимости постоянного тестирования, низкое качество получаемого ПО, риски невыполнения требований к разрабатываемому ПО.

С целью избежать выше обозначенных ошибок во время разработки ПО, активно применяют каскадную, инкрементную и эволюционную стратегии. Опишем каждую из вышеперечисленных стратегий подробнее.

Каскадной стратегией называют стратегию с однократным последовательным прохождением каждого этапа разработки ПО от анализа требований до тестирования, без возврата на предыдущие этапы разработки. К достоинствам данной стратегии следует отнести неизменные требования, указанные в техническом задании (ТЗ), простоту ее применения за счет однократного прохождения этапов разработки, а также контроля проекта, удобство управления проектом, легкость взаимодействия с заказчиком. Недостатками можно считать необходимость четкого формулирования требований к разработке (так как изменять требования в процессе разработки ПО при данной стратегии не допускается), высокие затраты на разработку из-за невозможности вернуться на предыдущие этапы, малое взаимодействие с конечным потребителем ПО (только на этапе тестирования) высокая цена ошибки, при допущении ее на ранних этапах разработки, отсутствие возможности реализации на рынке промежуточных версий ПО.

Каскадную стратегию следует применять при получении четких неизменных требований, указанных в ТЗ, которые гарантированно не изменятся в процессе разработки ПО, в случаях разработки ПО по аналогии с уже разработанными и успешно функционирующими ПО того же типа или разработки новой версии

						ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подп.	Дат.			15

существующей программы, а также для разработки конкретных составных модулей некоторого проекта, использующего другие стратегии разработки.

Инкрементная стратегия представляет собой улучшенную версию каскадной стратегии или каскадную стратегию с обратной связью. Как и в каскадной модели, сначала определяются все требования к разрабатываемому ПО, затем последовательно проходятся все этапы разработки, и на выходе первого цикла получают первую версию ПО, в которой, как правило, реализуются основные или базовые функции, требуемые в техническом задании на разработку. Затем идет доработка программы, расширение ее функционала. Версии программы, получаемые в каждом цикле, получили название инкремента.

Преимуществами инкрементной стратегии можно считать получение на каждом цикле новой версии ПО, что дает возможность реализовывать на рынке несколько версий ПО, имеющих различное количество доступных функций, сокращение денежных затрат на разработку, возможность работы с изменившимися требованиями, уменьшение рисков, гарантированное повышение качества ПО после каждого инкремента, работа с конечными потребителями на этапах тестирования различных версий ПО.

К недостаткам следует отнести повышение сложности планирования работ, влияние изменившихся требований к ПО на уже разработанный функционал, различная сложность работ на разных инкрементах.

На данный момент используется тактика экстремального проектирования – один из вариантов инкрементной стратегии с короткими по длительности циклами и малым приращением функционала ПО на каждом цикле.

Инкрементную модель следует применять при наличии конкретных требований к ПО, часть из которых может быть уточнена в процессе разработки, при разработке проектов средней или высокой сложности с четкими требованиями по ТЗ, при необходимости быстрого внедрения ПО с как минимум базовым функционалом и возможностью обновления версий.

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						16
Изм.	Лист	№ докум.	Подп.	Дат.		

Эволюционная стратегия, как и инкрементная, является многократной по отношению к прохождению этапов разработки. В отличие от инкрементной стратегии, требования к ПО для данной стратегии разработки на первом этапе задаются частично и в дальнейшем уточняются, что приводит к возрастанию длительности каждого цикла и сокращению количества версий. Однако каждая из полученных при помощи эволюционной стратегии версий значительно отличается от другой (наблюдается весомый прирост функционала в каждой версии).

К основным преимуществам эволюционной стратегии можно отнести легкость уточнения и внесения исправлений в требования к разработке ПО, возможность поставлять на рынок промежуточные версии продукта, уменьшение рисков, обширное участие конечных потребителей на этапах тестирования различных версий ПО, начиная с самых ранних, высокое качество получаемого ПО, согласование требований с заказчиками на всех циклах разработки.

К основным недостаткам можно отнести отсутствие информации о необходимом количестве итерации, повышение сложности планирования рабочего процесса и управления проектом, необходимость постоянного участия заказчика в разработке, повышенные требования к системам контроля версий.

Эволюционную стратегию следует применять в случаях, когда нет возможности заранее предугадать все возможные требования к проектируемому ПО, при разработке проектов повышенной сложности или проектов, основанных на совершенно новых технологиях, проектов по разработке новейшего ПО, не имеющего аналогов на рынке.

Очевидно, что от выбора стратегии проектирования зависит качество разрабатываемого ПО. Преимущества одних стратегий над другими могут оказаться недостатками при неверном выборе стратегии разработки. Аналогично, некоторые из вышеуказанных недостатков стратегий разработки могут оказаться преимуществами при разработке определенного вида ПО.

На стадии кодирования алгоритм программы преобразуется в понятный для конечного устройства вид. Использование высокоуровневых языков

программирования и интеллектуализированных сред разработки позволяет значительно упростить данный процесс. Этап кодирования считается наиболее простым этапом разработки, так как освоен лучше, чем другие этапы создания программ, и четко формализован.

Этап тестирования самый затратный в финансовом плане (на него приходится около половины расходов на создание системы). Ошибки при планировании тестирования приводят к существенному увеличению сроков разработки системы и срыву графиков разработки.

Обычно план проведения тестирования составляется на этапе проектирования. В процессе тестирования в качестве входных данных используются случайные данные, которые система могла бы получить на вход (характерные для системы в рабочем состоянии) и отслеживают выходные параметры.

Тестирование делится на автономное, комплексное и системное.

Автономное тестирование: проверка модуля происходит с использованием данных, предоставленных программистом. Происходит имитация программной среды модуля программами управления тестированием (UUT). Вместо настоящих подпрограмм, с которыми взаимодействует данный модуль, в программах управления тестированием используются фиктивные подпрограммы.

Модуль, который был успешно протестирован на автономном уровне, подвергается комплексному тестированию.

Комплексное тестирование: на данном этапе группы программных компонент тестируются совместно, таким образом на выходе получаем практически полностью протестированную систему. Комплексное тестирование позволяет обнаружить ошибки, пропущенные на этапе автономного тестирования. Исправление этих ошибок может составлять до одной четвертой от общих затрат [5, с.16].

Системное (оценочное) тестирование: на конечном этапе проверки проводится тестирование всей системы в целом. Главное требование при этом – независимость проверочных тестов. Когда происходит сравнение нескольких систем (альтернативных разработок) системное тестирование называется сравнительным.

					<i>ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР</i>	<i>Лист</i>
						18
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дат.</i>		

Для определения корректности работы программы вводится ряд критериев, которым должно удовлетворять тестирование [5, с. 20].

С понятием «тестирование» тесно связаны такие понятия, как «верификация» и «испытание».

Испытание системы проводится при помощи тестирования и необходимо для того, чтобы продемонстрировать, что функционирование системы соответствует спецификациям на данную систему.

Верификация – предоставление доказательств того, что функционирование системы удовлетворяет спецификациям на данную систему.

Аттестация – понятие, включающее в себя весь процесс испытания и верификации для создания правильно работающей программы или системы.

Отклонения от нормальной работы системы подразделяют на три вида: сбой системы, выброс, ошибка.

Сбой системы: программа не работает в соответствии с составленными на нее спецификациями.

Выброс: обработка правильными алгоритмами некоторых данных приводит к сбою системы.

Ошибка: дефект в алгоритме программы, вызывающий выброс.

Существуют понятия «надежная» и «правильная» программа. Они не взаимозаменяемы. Система является правильной, если в системе нет ошибок, а ее внутренние данные не содержат выбросов. Система называется надежной, если, несмотря на сбои, она продолжает удовлетворительно функционировать [5, с. 20].

По многим причинам в программу могут вноситься изменения по мере ее эксплуатации. Приведем некоторые из них:

1. Заказчик нечетко сформулировал свои требования к программе или не удовлетворен готовой программой.
2. Обнаружились ошибки, которые были пропущены на стадии тестирования.
3. Потребовалась модификация программы для ее корректной работы при частных условиях функционирования.

									Лист
									19
Изм.	Лист	№ докум.	Подп.	Дат.					

4. Сопровождение многих версий системы (чтобы избежать появления слишком большого числа версий программы или системы, разработчики вносят изменения в определенные промежутки времени – «периоды обновления»).

Рассмотрим основные методы разработки программного обеспечения.

Структурное программирование. Идея структурного программирования заключается в том, что структура программы должна отражать алгоритм решения поставленной задачи. Суть метода сводится к исключению из языков программирования оператора GoTo (оператор безусловного перехода), применению стандартного набора управляющих конструкций и использованию одного из методов нисходящего проектирования. Основатель принципа, Э. Дейкстра считал, что чем меньше операторов GoTo использует программист, тем более он квалифицирован.

По сравнению с интуитивным программированием структурное программирование обладает следующими достоинствами:

- упрощение процесса тестирования;
- повышение эффективности работы программистов;
- упрощение сопровождения за счет повышения ясности ПО;
- снижение времени выполнения и затрат памяти.

Алгоритм формируется из последовательно расположенные модули. Каждый из таких модулей проектируется с единственным входом и выходом.

Существуют три базовые структуры, на основе которых составляются алгоритмы: функциональный блок, конструкция принятия решения, конструкция обобщенного цикла.

Функциональный блок – это отдельный вычислительный оператор или любая последовательность вычислений с единственным входом и единственным выходом [3, с.40]. Последовательность функциональных блоков (так называемая конструкция следования) может быть сведена к одному функциональному блоку (преобразование Бома – Джакопини).

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						20
Изм.	Лист	№ докум.	Подп.	Дат.		

Выбор между двумя альтернативами вычислений в зависимости от выполнения условия (ряда условий) обеспечивается с помощью конструкции принятия двоичного решения.

Конструкция обобщенного цикла представляет собой (базовый случай) цикл с предусловием (пока условие истинно, тело цикла выполняется).

Допускается также использование дополнительных циклических конструкций (цикл с параметром, цикл с постусловием), оператора Case (усложнение конструкции If-Then-Else), подпрограмм с несколькими входами или выходами.

Модульное проектирование – один из первых подходов к разработке структуры программных средств. Оно основано на разбиении алгоритма на структурно-завершенные компоненты. В последствии полученные модули иерархически организуются.

Модульная программа – это программа, в которой любую часть логической структуры можно изменить, не вызывая изменений в ее других частях [6, с. 27].

У модульного проектирования есть ряд достоинств. К ним можно отнести упрощение детализации алгоритмов обработки данных, облегчение чтения и понимания программ.

Модульность структуры ведет к появлению недостатков, таких как увеличение времени выполнения программ и увеличение объема программы.

Нисходящее проектирование заключается разбиении задачи на подзадачи таким образом, чтобы рассмотреть каждую подзадачу отдельно.

На основе функциональных требований к будущему ПО, определяется обобщенная, не детализированная структура программы. Следующим шагом является разработка не детализированных ранее компонент ПО на основе требований к отдельным функциональным. Далее процесс происходит рекурсивно до окончательного уточнения каждой части программы.

При нисходящем проектировании каждый шаг завершается проверкой правильности и корректности внесенных уточнений и их влияния на функционирование всего программного средства.

										Лист
										21
Изм.	Лист	№ докум.	Подп.	Дат.	ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР					

Разработка ПО согласно ГОСТ Р ИСО/МЭК 12207-2010 использует в основе метод нисходящего проектирования.

У нисходящего проектирования применяются две основные стратегии реализации: пошаговое уточнение и анализ сообщений.

При использовании стратегии пошагового уточнения детализация уровней начинается с верхнего, а каждый следующий шаг производит детализацию программных компонент на более низком уровне.

Возможны следующие классические способы реализации пошагового уточнения [5, с. 36]:

- проектирование программного средства с помощью псевдокода и управляющих конструкций структурного программирования;
- использование комментариев для описания обработки данных.

Рассмотрим недостатки и преимущества метода пошагового уточнения.

Преимущества метода пошагового уточнения:

- проектирование корректной структуры программы занимает центральное положение в списке приоритетов;
- проверка корректности ПО может проводиться на всех этапах разработки.

У стратегии пошагового уточнения есть серьезный недостаток: возможность возникновения необходимости во внесении изменений, которые повлекут за собой необходимость пересмотреть решения, принятые на более ранних этапах.

В основе методов восходящего проектирования используется реализация снизу-вверх, то есть реализация сначала функций нижнего уровня, затем формирования на их основе компонент более высокого уровня с последующей реализацией функций более высокого уровня.

Использование данного метода сопряжено с риском начать реализацию с несущественных деталей, что приведет к невозможности реализации полного алгоритма.

Использование одного метода проектирования встречается довольно редко. В основном, если существуют уже готовые разработанные модули.

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						22
Изм.	Лист	№ докум.	Подп.	Дат.		

Зачастую применяется комплекс, который сочетает в себе как методы нисходящего проектирования, так и методы восходящего проектирования.

Существует два основных способа реализации «комбинированного» метода:

1. Предварительное определение и проектирование наиболее важных промежуточных уровней. Дальнейшее проектирование вверх и вниз от этих уровней одновременно восходящим и нисходящим проектированием.

2. Одновременное проектирование как восходящим, так и нисходящим методами проектирования с последующей «встречей» на одном из промежуточных уровней.

Так как фактически проектируется два различных уровня (верхний и нижний) требуется обеспечить согласование интерфейса. Его выполнение производится на окончательном этапе проектирования.

При проектировании методами расширения ядра сначала формируется ядро программы, которое выполняет какую-то основную функцию, после чего функционал ядра постепенно расширяется до тех пор, пока структура алгоритма полностью не сформируется.

Метод структурного программирования JSP Джексона. Наибольшую эффективность достигает в случае высокой степени структуризации данных.

В основе метода лежит принцип разработки структуры программы, зависящей от структуры данных.

Существует набор конструкций, которым определяются структуры данных и программ. Основными конструкциями являются конструкции выбора данных, последовательности, повторения и элементарная конструкция.

Для формирования сложных структур данных на основе комбинации четырех вышеописанных конструкций используется три структуры данных: иерархическая, сетевая и реляционная.

При построении иерархических структур используется понятие подчинения нижнего уровня верхнему.

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подп.	Дат.		23

Сетевая структура данных содержит связи между отдельными компонентами (одного или разных уровней структуры).

Реляционный подход организации данных подразумевает представление данных в виде совокупности таблиц. Сами таблицы при этом имеют иерархическую структуру.

По уровню тестирования методы делятся на компонентное, интеграционное и системное тестирование.

При компонентном тестировании происходит проверка отдельных небольших частей программы. Интеграционное тестирование позволяет выявить наличие ошибок во взаимосвязях между отдельно-взятыми частями программы, каждая из которых проверена компонентным тестированием. Системное тестирование позволяет оценить комплексно работу готовой программы, а также удобство взаимодействия конечного пользователя с программой.

Метрологическая аттестация ПО – признание метрологической службой правомерности, а также законности применения данного программного обеспечения, основанное на определении того, в какой степени ПО влияет на метрологические характеристики средств измерений и измерительных систем, в которых применяется данное ПО.

Метрологическая аттестация необходима для определения оценок характеристик составляющих погрешности результатов обработки в конкретных условиях применения этого алгоритма. Аттестация ПО СИ должна устанавливать, соответствует ли ПО метрологическим требованиям.

Определить влияние ПО на метрологические характеристики средств измерений можно двумя способами.

Первый заключается в рассмотрении СИ с ПО как единого объекта для метрологического анализа. При втором подходе определяются метрологические характеристики ПО отдельно. Определение МХ ПО независимо от средства измерения проводится в соответствии с МИ 2174-91 [7].

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						24
Изм.	Лист	№ докум.	Подп.	Дат.		

Обязательными процедурами методики аттестации алгоритма считаются:

1. Определение основных характеристики оцениваемых алгоритмов. Для алгоритмов, применяемых при обработке результатов прямых многократных измерений, примерами характеристик могут являться границы систематических погрешностей; их среднеквадратическое отклонение (СКО).

2. Определение модели используемых исходных данных, которые поступают на обработку. Примерами таких данных могут служить независимые случайные величины (НСВ), которые имеют гауссовские распределения и для которых отсутствует информация об их математическом ожидании и дисперсии; НСВ, имеющие равномерное распределение.

3. На некоторых, заранее выбранных типовых моделях проводится оценка значений характеристик аттестуемых алгоритмов или программ.

4. Составление свидетельства (документа об аттестации) для разработанного алгоритма.

Результаты метрологической аттестации не завершаются выдачей свидетельства о прохождении аттестации. Поскольку программа реализована для средства измерений или измерительной системы, результаты аттестации находятся в разделах технической документации по обработке данных для некоторых видов измерения или в методиках выполнения измерения.

Помимо метрологических характеристик в документации может указываться также типовые модели исходных данных, методы и способы оценки выбранных характеристик и зависимости метрологических характеристик от параметров модели. Данная информация позволяет подтвердить результаты аттестации с помощью повторных испытаний.

Вышеуказанные зависимости MX от параметров модели можно представить в аналитическом виде, в табличной форме, в комбинированном виде (часть параметров описывается аналитически, а другая часть при помощи таблиц), в графическом виде.

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						25
Изм.	Лист	№ докум.	Подп.	Дат.		

1.3 Современное состояние в области коррекции динамических погрешностей

В статье [8] исследована адаптивная измерительная система, синтезированная на основе метода коррекции динамической погрешности для реальных шумов системы, по критерию минимизации полной динамической погрешности обусловленной инерционностью датчика, а также шумами и помехами, присутствующими на его выходе. Синтез алгоритма произведен на основе градиентного метода поиска минимума сигнала оценки динамической погрешности. В статье представлена математическая модель динамической самонастраивающейся измерительной системы, синтезированной на основе модели датчика первого порядка (рисунок 5).

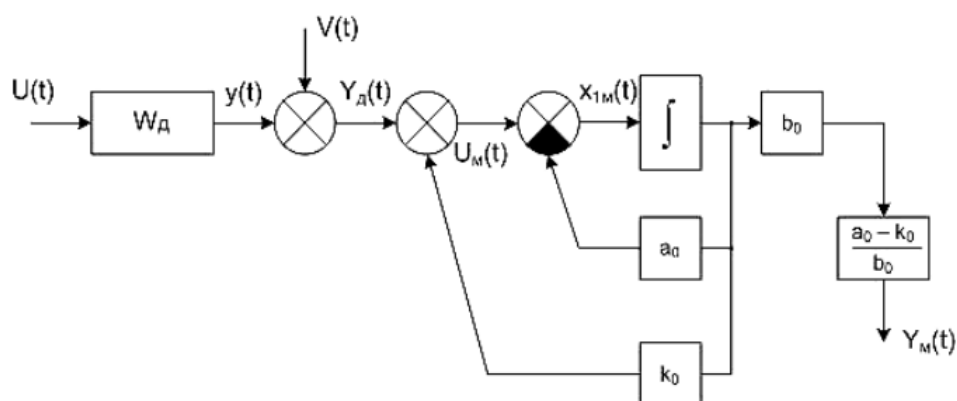


Рисунок 5 – Структурная схема самонастраивающейся измерительной системы

Представлены результаты суперкомпьютерного моделирования измерительной системы первого порядка, выполненного на основе суперкомпьютерного центра ЮУрГУ. Сделан вывод, что глубокая математическая обработка результатов измерения, выполняемая после завершения эксперимента, позволяет снизить оценку динамической погрешности измерений на 84%.

Статья [9] рассматривает способы решения проблемы коррекции динамической погрешности датчиков, которые входят в системы мониторинга и управления в реальном времени. Предлагается ряд рекомендаций по синтезу распределенных

информационных измерительных систем с наименьшей динамической погрешностью.

Рассмотрен компенсационный снижения двух составляющих динамической погрешности: погрешности, связанной с затратами времени на реализацию вычислительных процессов, и погрешности цифрового фильтра.

Для снижения динамической погрешности систем используются аппаратные методы, архитектурные методы и численные методы цифровой обработки информации датчиков.

К архитектурным методам относят распараллеливание решения задач и параллельный сбор информации с датчиков, к аппаратным методам – применение более быстродействующих АЦП, вычислителей и менее инерционных датчиков, к математическим метода цифровой обработки информации – методом экстраполяции переменной.

Многошаговая экстраполяция 1-ого типа на основе полинома Лагранжа по отфильтрованным значения $U_{иф}$ определяется выражениями:

$$U_{1r(i+k)}^* = \sum_{j=0}^r C_j U_{(i+(j-r)k)\phi}, \quad (6)$$

$$C_j = (-1)^{r-j} \frac{(r+1)!}{j!(r+1-j)!}. \quad (7)$$

где k – количество шагов экстраполяции,

C_j – коэффициенты в формуле экстраполяции,

r – порядок формулы экстраполяции.

Компенсация погрешности путем применения экстраполяционного метода позволяет скорректировать искаженный сигнал путем снижения динамической погрешности в 2–3 раза. Использование нескольких подходов минимизации погрешности улучшает качество работы систем мониторинга.

В статье [10] также рассматривается проблема коррекции погрешности динамических измерений. Погрешность динамического измерения определяется

									Лист
									27
Изм.	Лист	№ докум.	Подп.	Дат.					

двумя составляющими: первая, обусловленная динамическими свойствами (инерционностью) датчика, и вторая, обусловленная присутствием аддитивного шума на выходе датчика. Уменьшение динамической погрешности заключается в одновременном корректировании этих двух составляющих. Разработана структура измерительной системы с оценкой погрешности динамического измерения. Коррекция динамической погрешности осуществляется путем одновременного восстановления и фильтрации входного измеряемого сигнала датчика. Предложена структура специального фильтра с предварительной коррекцией передаточной функции датчика к виду удобному для дальнейшей обработки измеренного сигнала. Дальнейшая обработка сигнала заключается в итеративном применении восстанавливающего КИХ-фильтра с оценкой динамической погрешности. Проведено компьютерное моделирование предложенной измерительной системы. В ходе моделирования рассмотрен датчик второго порядка. Получены оптимальные значения порядка восстанавливающего фильтра для входных сигналов различного вида в присутствии аддитивного случайного шума на выходе датчика. Продемонстрировано уменьшение оценки динамической погрешности на основе предложенной структуры измерительной системы (рисунок б).

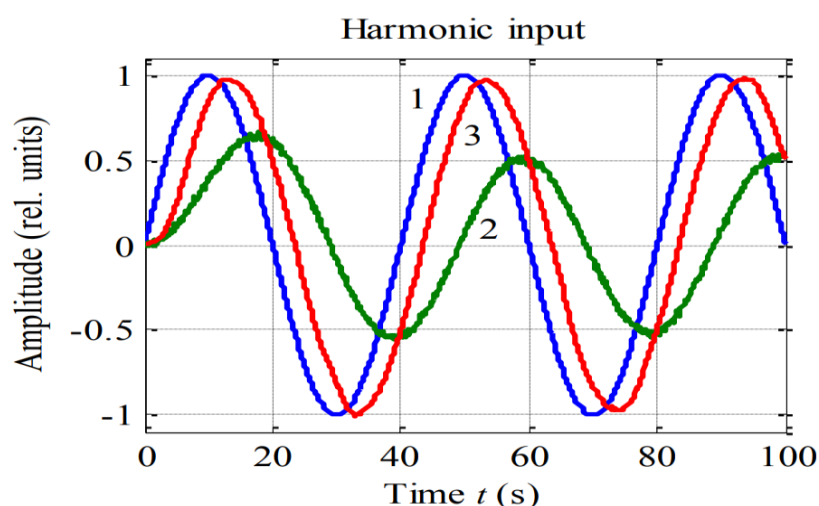


Рисунок б – Графики сигналов измерительной системы

1 – входной сигнал датчика, 2 – выходной сигнал датчика,

3 – восстановленный сигнал

									Лист
									28
Изм.	Лист	№ докум.	Подп.	Дат.					

Учет источников динамической погрешности измерения координатно-измерительной машины и их взаимного влияниями затрудняет нахождение модели погрешности динамических измерений методом идентификации параметров. В статье [11] используется оценка ядра для анализа сложной взаимосвязи между значениями динамических погрешностей измерения и трехмерными координатами и параметрами прямого компьютерного управления, включая скорость позиционирования, расстояние приближения и скорость контакта. В модели учитываются возможные динамические погрешности измерений. Результаты исследований показывают, что за счет использования регрессионной модели оценки ядра описываемый метод более точно и эффективно прогнозирует динамические погрешности измерений, чем обычная регрессия по методу наименьших квадратов и модель регрессии на основе частичных наименьших квадратов.

В статье [12] описывается метод прогнозирования погрешностей динамических измерений с оптимизацией опорных векторов алгоритмом «firefly».

Точность прогнозирования динамических погрешности при измерениях оказывает существенное влияние на точность и стабильность датчиков. Чтобы решить проблему низкой точности модели при классическом способе прогнозирования погрешностей динамических измерений, применяется метод опорных векторов. В этом исследовании выбор параметров для метода опорных векторов производится при помощи алгоритма «firefly». Это позволяет избежать локального минимального значения динамической погрешности, которое появляется в классическом методе оптимизации параметров модели. Для моделирования используют данные о динамической погрешности двух датчиков.

Для оценки характеристик моделей используются среднеквадратическая ошибка предсказания и средняя приведенная ошибка предсказания. Эти значения также сравниваются с данными, полученными в результате поиска в сетке опорных векторов и оптимизации методом роя частиц в методе опорных векторов.

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						29
Изм.	Лист	№ докум.	Подп.	Дат.		

Результаты экспериментов показывают, что модель опорных векторов, основанная на алгоритме «firefly», более точно прогнозирует динамические погрешности измерений, чем другие рассмотренные модели.

В рамках научного направления динамических измерений научной школы ЮУрГУ широко представлены исследования и разработки моделей динамических измерительных систем, основанных на методе модального управления [13], методах скользящих режимов [14, 15], параметрической адаптации систем [16, 17], а также методом нейросетевого управления [18, 19].

Выводы по разделу один

В процессе написания первой главы были проанализированы математические модели измерительных систем, их свойства, этапы их разработки, изучена технология разработки программного обеспечения, описан жизненный цикл ПО, стратегии его разработки, описаны виды аттестации ПО и их особенности, проведен обзор существующих методов коррекции динамической погрешности результатов измерений. При проведении аналитического обзора были рассмотрены следующие методы коррекции динамической погрешности:

1. Применение структуры измерительной системы на основе КИХ-фильтра.
2. Экстраполяционный метод.
3. Метод прогнозирования погрешностей с оптимизацией опорных векторов алгоритмом «firefly».
4. Метод прогнозирования погрешностей на основе метода оценки ядра.
5. Методы, представленные научной школой ЮУрГУ, в том числе с использованием суперкомпьютера «Торнадо».

Можно сделать вывод, что последние 5 лет разработка методов и алгоритмов коррекции динамической погрешности продолжает оставаться актуальной во всем мире.

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						30
Изм.	Лист	№ докум.	Подп.	Дат.		

2. МЕТОД МОДАЛЬНОГО УПРАВЛЕНИЯ ДИНАМИЧЕСКИМИ ХАРАКТЕРИСТИКАМИ ИЗМЕРИТЕЛЬНЫХ СИСТЕМ

2.1 Динамическая модель измерительной системы с дополнительным каналом оценки динамической погрешности

Одной из перспективных тенденций в развитии современной информационно-измерительной техники является ее интеллектуализация [20], позволяющая выполнять сложные измерительные процедуры специальными аппаратными средствами, и построение измерительных преобразователей, способных к индивидуализации алгоритмов обработки, в том числе и путем изменения своей структуры и параметров на основе накопленной априорной и получаемой измерительной информации [21].

Передаточная функция датчика в общем виде:

$$W_D(p) = \frac{\prod_{i_1=1}^l (T_{i_1}^2 p^2 + 2\xi_{i_1} T_{i_1} p + 1) \prod_{j_1=1}^q (T_{j_1} p + 1)}{\prod_{i=1}^r (T_i^2 p^2 + 2\xi_i T_i p + 1) \prod_{j=1}^s (T_j p + 1)}, \quad (8)$$

где $T_{i_1}, T_i, T_{j_1}, T_j$ – постоянные времени;

ξ_{i_1}, ξ_i – коэффициенты демпфирования;

p – комплексная переменная.

Уравнение датчика представим в форме линейного дифференциального уравнения с постоянными коэффициентами:

$$\begin{aligned} \frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \dots + a_0 y(t) = \\ = b_m \frac{d^m U(t)}{dt^m} + b_{m-1} \frac{d^{m-1} U(t)}{dt^{m-1}} + \dots + b_0 U(t), \end{aligned} \quad (9)$$

где $y(t)$ – выходной сигнал датчика;

						Лист
					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	31
Изм.	Лист	№ докум.	Подп.	Дат.		

$U(t)$ – входной измеряемый сигнал;

$a_{n-1}, \dots, a_1, a_0, b_m, b_1, b_0$ – постоянные коэффициенты ($m \leq n$).

Для уменьшения динамической погрешности измерения посредством настройки параметров измерительной системы (ИС) необходимо достоверно оценить динамическую погрешность непосредственно в процессе измерения. В данном случае следует рассматривать динамическую модель измерительной системы с дополнительным каналом оценки погрешности, которая будет учитывать приведенный к выходу датчика шум (рисунок 7).

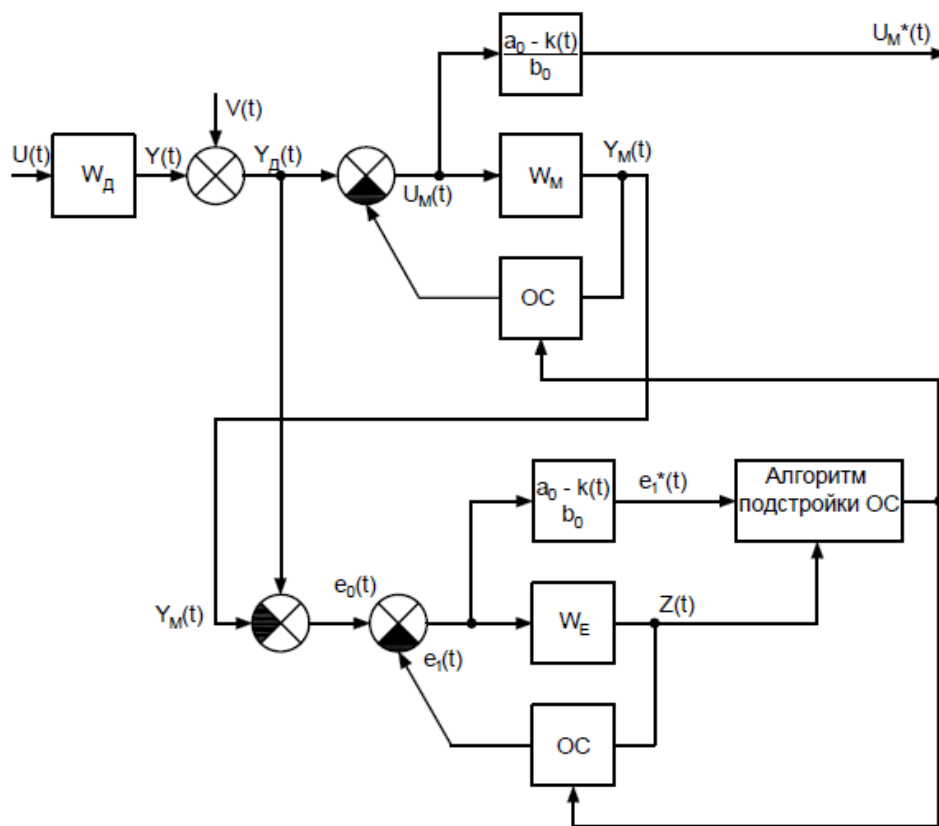


Рисунок 7 – Обобщенная структурная схема динамической измерительной системы

В качестве основы такой модели будем использовать базовую динамическую модель измерительной системы на основе модели датчика. Структура измерительной системы включает в себя полную динамическую модель датчика, который обеспечивает изменение входного сигнала вплоть до аналого-цифрового

преобразования [21, с. 24]. К выходу датчика приведен сигнал шума. Сигнал с выхода датчика поступает на вход модели, аналогичной модели датчика, охваченной обратными связями с изменяемыми коэффициентами k_0, k_1, \dots, k_{n-1} .

Структурная схема ИС в форме пространства состояний, где $x_{1M}(t), x_{2M}(t), \dots, x_{nM}(t)$ – координаты состояния модели датчика, представлена на рисунке 8.

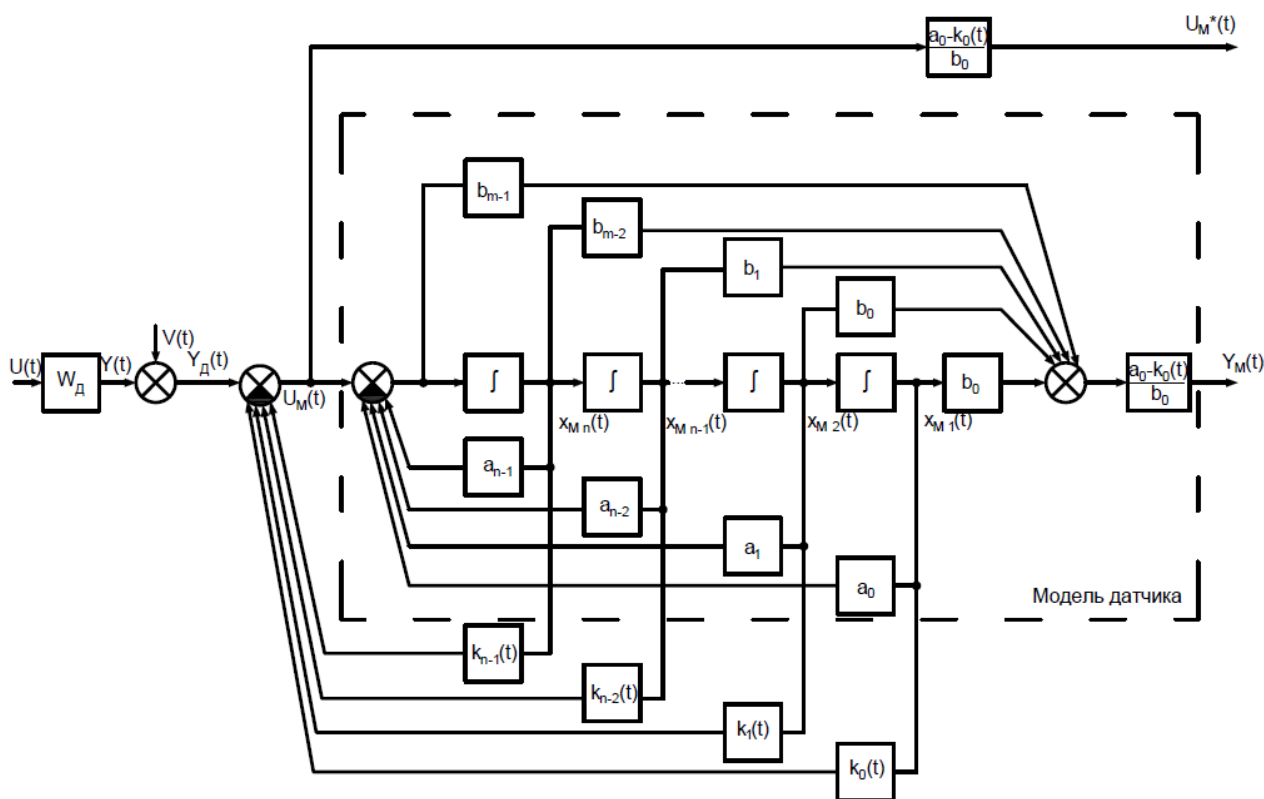


Рисунок 8 – Структурная схема измерительной системы

Настройка коэффициентов k_0, k_1, \dots, k_{n-1} обратных связей ИС проводится исходя из соображений близости выходных сигналов датчика и его модели для данного уровня шума. Критерием близости служит динамическая погрешность ИС.

Запишем передаточную функцию датчика в виде отношения полиномов:

$$W_D(p) = \frac{y(p)}{U(p)} = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0}{p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0}, \quad (11)$$

где $y(p)$ – изображение выходного сигнала датчика;

$U(p)$ – изображение выходного сигнала датчика.

Запишем передаточную функцию модели датчика:

$$W_M(p) = W_D(p) = \frac{y_M(p)}{U_M(p)}, \quad (12)$$

где $y_M(p)$ – изображение Лапласа выходного сигнала модели датчика;

$U_M(p)$ – изображение Лапласа входного сигнала модели датчика.

При наличии реального шума преобразование Лапласа сигнала на выходе датчика примет следующий вид:

$$y_D(p) = y(p) + V(p), \quad (13)$$

где $V(p)$ – изображение высокочастотного шума на выходе датчика.

Изображения координат состояния модели $x_{1M}(p)$, $x_{2M}(p)$, ..., $x_{nM}(p)$ связаны соотношениями:

$$\begin{aligned} x_{2M}(p) &= p x_{1M}(p), \\ x_{3M}(p) &= p^2 x_{2M}(p), \\ &\vdots \\ x_{nM}(p) &= p^{n-1} x_{2M}(p). \end{aligned} \quad (14)$$

Тогда

$$\begin{aligned} p x_{nM}(p) &= p^n x_{1M}(p) = \\ &= U_M(p) - a_{n-1} p^{n-1} x_{1M}(p) - \dots - a_1 p x_{1M}(p) - a_0 x_{1M}(p). \end{aligned} \quad (15)$$

Выразим значение $x_{1M}(p)$:

$$x_{1M}(p) = \frac{U_M(p)}{p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0}. \quad (16)$$

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						34
Изм.	Лист	№ докум.	Подп.	Дат.		

Исходя из структурной схемы, представленной на рисунке 8, изображение выходного сигнала модели датчика будет определяться следующим выражением:

$$y_M(p) = b_0 x_{1M}(p) + b_1 x_{2M}(p) + \dots + b_{m-1} x_{mM}(p) + b_m x_{mM}^{(1)}(p). \quad (17)$$

С учетом соотношений координат состояния данное выражение приобретет следующий вид:

$$y_M(p) = (b_0 + b_1 p + \dots + b_{m-1} p^{m-1} + b_m p^m) x_{1M}(p). \quad (18)$$

Подставим в выражение значение $x_{1M}(p)$:

$$\frac{y_M(p)}{U_M(p)} = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0}{p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0} = W_D(p). \quad (19)$$

Изображение входного сигнала модели датчика, исходя из структурной схемы, будет определяться:

$$U_M(p) = y(p) + V(p) + k_{n-1} x_{nM}(p) + \dots + k_1 x_{2M}(p) + k_0 x_{1M}(p). \quad (20)$$

С учетом уравнений передаточной функции датчика, сигнала на выходе датчика и соотношений координат состояния модели представим уравнение входного сигнала модели датчика в следующем виде:

$$U_M(p) = W_D(p)U(p) + V(p) + (k_{n-1} p^{n-1} + \dots + k_1 p + k_0) x_{1M}(p). \quad (21)$$

Подставим в выражение значение $x_{1M}(p)$:

$$U_M(p) = W_D(p)U(p) + V(p) + \frac{k_{n-1} p^{n-1} + \dots + k_1 p + k_0}{p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0} U_M(p). \quad (22)$$

Тогда изображение выходного сигнала измерительной системы будет определяться как:

$$U^*(p) = W_{uc} U(p). \quad (29)$$

Изображение выходного сигнала модели датчика примет вид:

$$y_M(p) = W_d(p)U_M(p) = [y(p) + V(p)] \cdot W_{uc}(p). \quad (30)$$

С учетом того, что изображение сигнала с выхода датчика определяется как:

$$y_D(p) = W_d(p)U(p) + V(p), \quad (31)$$

сформируем изображение сигнала погрешности в виде:

$$\begin{aligned} e_0(p) &= y_D(p) - y_M(p) \left(\frac{a_0 - K(t)}{b_0} \right) = \\ &= (U(p) - U^*(p))W_d(p) + V(p) = e_{ис}(p)W_d(p) + V(p), \end{aligned} \quad (32)$$

где $e_{ис} = U(p) - U^*(p)$ – погрешность измерительной системы.

Формирование сигнала $e_0(p)$ по выражению (32) дает оценку погрешности измерительной системы, которая отличается от истинной оценки динамической погрешности $e_{ис}$ точно также, как выходной сигнал датчика относительно входного. На вход канала оценки динамической погрешности измерительной системы подается сигнал аналогичный сигналу на входе корректирующего устройства, что дает возможность скорректировать оценку погрешности аналогично корректированию сигнала самого датчика.

С учетом (23) выражение (32) можно представить в виде суммы динамической и шумовой составляющих погрешности:

$$e_0(p) = U(p)W_d(p)(1 - W_{uc}(p)) + V(p)(1 - W_{uc}(p)). \quad (33)$$

									Лист
									38
Изм.	Лист	№ докум.	Подп.	Дат.	ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР				

Учитывая, что

$$p^n x_{1e}(p) = e_1(p) - a_{n-1} p^{n-1} x_{1e} - a_1 p x_{1e} - a_0 x_{1e}, \quad (37)$$

координата состояния x_{1e} примет вид:

$$x_{1e} = \frac{e_1(p)}{p^n + a_{n-1} p^{n-1} - a_1 p - a_0}. \quad (38)$$

Подставляя в выражение (36) выражение (38), получим:

$$e_1(p) = \frac{W_{uc}^1(p)}{W_d(p)} e_0. \quad (39)$$

Запишем преобразование Лапласа для выходного сигнала канала оценки динамической погрешности с учетом (39) и (32):

$$e_1^*(p) = e_{uc}(p) W_{uc}(p) + V(p) \frac{W_{uc}(p)}{W_d(p)}. \quad (40)$$

Анализируя выражения (40), видим, что изображение сигнала на выходе канала оценки динамической погрешности усиливается и искажается аналогично изображению выходного сигнала измерительной системы, поскольку проходит через ПФ $W_{uc}(p)$. Изображение сигнала шума, присутствующего в ИС, проходит через составляющую передаточной функции $\frac{W_{uc}(p)}{W_d(p)}$, аналогичную ПФ по приведенной шумовой составляющей измерительной системы. То есть, сигнал шума в канале оценки динамической погрешности усиливается и искажается в той же мере, что и шум в канале измерения системы.

При условии, что измерительная система будет обеспечивать малую динамическую погрешность $W_{uc}(p) \approx 1$, сигнал в канале оценки динамической

						Лист
					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	40
Изм.	Лист	№ докум.	Подп.	Дат.		

В ИС, со структурной схемой (см. рисунок 8) для получения ПФ (23) также достаточно модели свободных движений, что дает возможность упростить модель, исключив из нее связи с коэффициентами b_0, b_1, \dots, b_m . Однако для правильного формирования сигнала ошибки ИС эти связи должны присутствовать.

ПФ канала оценки динамической по приведенной шумовой составляющей, определенная в отсутствие полезного входного сигнала имеет вид:

$$W_{e_{\text{шум}}}(p) = \left. \frac{e_1^*(p)}{e_{uc}(p)} \right|_{U(p)=0} = \frac{V(p) \frac{W_{uc}(p)}{W_D(p)} (1 - W_{uc}(p))}{V(p) \frac{W_{uc}(p)}{W_D(p)}} = 1 - W_{uc}(p). \quad (44)$$

Приведенная структурная схема, отражает все существенные связи при реализации ИС в аналоговом виде и в виде программы цифровой обработки сигнала датчика.

2.2 Динамическая модель измерительной системы с дополнительным каналом оценки динамической погрешности 2 порядка

Рассмотрим математическое обоснование применения алгоритма к системе второго порядка. Структурная схема системы 2 порядка представлена на рисунке 10.

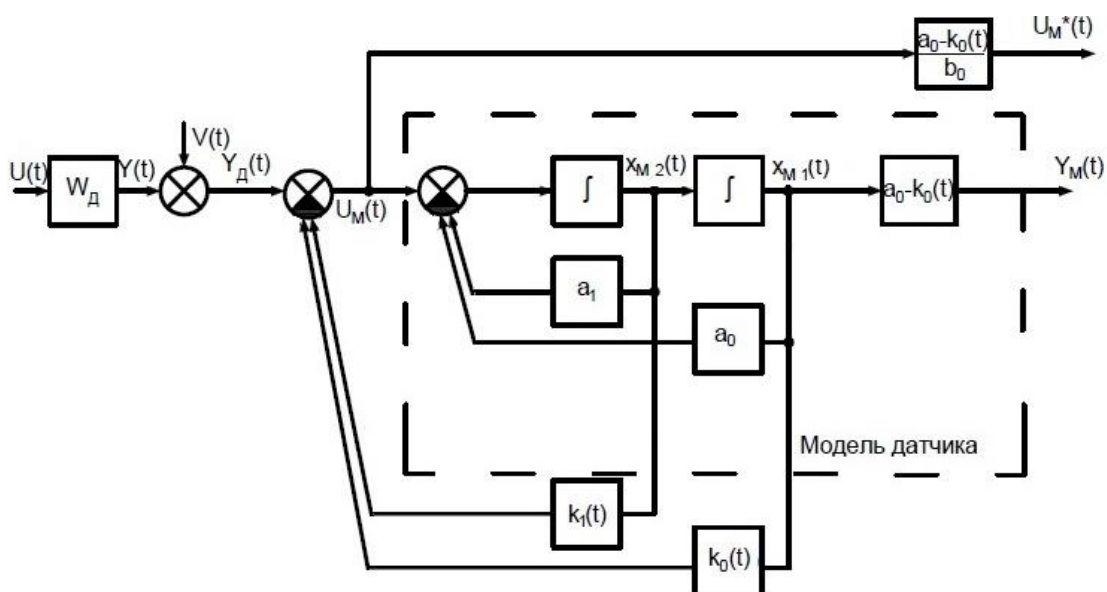


Рисунок 10 – Структурная схема системы 2 порядка

Передаточная функция датчика:

$$W_d(p) = \frac{1}{T^2 p^2 + 2\xi T p + 1}, \quad (45)$$

где T – постоянная времени датчика;

ξ – коэффициент демпфирования.

Передаточная функция датчика может быть представлена в виде:

$$W_d(p) = \frac{1}{p^2 + a_1 p + a_0}, \quad (46)$$

где $a_0 = \frac{1}{T^2}$;

$$a_1 = \frac{2\xi}{T}.$$

В соответствии со структурной схемой на рисунке 8 уравнения датчика имеют вид:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= U(t) - a_1 x_2(t) - a_0 x_1(t), \\ y_d(t) &= a_0 x_1(t) + V(t), \end{aligned} \quad (47)$$

где $x_1(t)$, $x_2(t)$ – координаты состояния датчика; y_d – выходной сигнал датчика.

Вводя координаты состояния модели датчика в корректирующем устройстве $x_3(t) = x_{1M}(t)$, $x_4(t) = x_{2M}(t)$, уравнения модели можно представить в виде:

$$\begin{aligned} \dot{x}_3(t) &= x_4(t), \\ \dot{x}_4(t) &= y_d(t) + (k_1 - a_1)x_4(t) + (k_0 - a_0)x_3(t), \end{aligned} \quad (48)$$

где k_0 , k_1 – настраиваемые параметры измерительной системы.

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						43
Изм.	Лист	№ докум.	Подп.	Дат.		

Выходной сигнал измерительной системы:

$$U^*(t) = U_M(t) \frac{a_0 - k_0}{b_0}. \quad (49)$$

Канал оценки динамической погрешности имеет в своей основе ту же модель датчика, определяемую передаточной функцией (46) с коэффициентами a_0 и a_1 . Грубая оценка погрешности формируется по выражению:

$$e_0(t) = y_d(t) - y_M(t). \quad (50)$$

Вводя координаты состояния канала оценки динамической погрешности $x_5(t) = x_{1e}(t)$, $x_6(t) = x_{2e}(t)$, уравнения канала можно представить в виде:

$$\begin{aligned} \dot{x}_5(t) &= x_6(t), \\ \dot{x}_6(t) &= e_0(t) + (k_1 - a_1)x_6(t) + (k_0 - a_0)x_5(t), \end{aligned} \quad (51)$$

Выходной сигнал канала оценки динамической погрешности измерительной системы:

$$e^* = e_1(t) \frac{a_0 - k_0}{b_0}. \quad (52)$$

Выводы по разделу два

В процессе написания главы была описана динамическая модель ИС на основе модели датчика с дополнительным каналом оценки погрешности, учитывающая приведенный к выходу датчика шум. Представлены структурные схемы ИС, дополнительного канала оценки погрешности, в которых отражены все связи ИС, необходимые для создания программной реализации алгоритма подстройки параметров ИС. Описан частный случай динамической ИС с ПФ 2-го порядка. Представлены структурная схема датчика 2-го порядка, уравнения координат состояния ИС 2-го порядка, сигнал с выхода канала оценки.

									Лист
									44
Изм.	Лист	№ докум.	Подп.	Дат.					

3. ПОДСТРОЙКА ПАРАМЕТРОВ ДИНАМИЧЕСКОЙ ИЗМЕРИТЕЛЬНОЙ СИСТЕМЫ

3.1 Среда программирования MATLAB

Среда программирования MATLAB создавалась в качестве языка программирования высокого уровня для технических вычислений. В основе среды MATLAB лежат алгоритмы матричных операций.

Данная система обладает открытой архитектурой, что позволяет пользователям получать полный доступ к кодам на простом языке программирования MATLAB, синтаксис которого схож с синтаксисом языков группы Си. Среда и язык программирования удобны для проведения научно-технических расчетов и создания средств построения, идентификации, анализа и моделирования систем.

В программный пакет MATLAB встроено большое число функций, предназначенных для решения различных задач. Использование встроенных функций, предназначенных для анализа, охватывают различные области математики, такие как:

- операции над матрицами;
- преобразование и интерполяция многочленов;
- статистические анализ и обработка данных;
- решение дифференциальных уравнений;
- операции над разреженными матрицами;
- решение целочисленных арифметических задач.

Система MATLAB предоставляет возможность использования объектно-ориентированного подхода для разработки алгоритмов программ. MATLAB является полностью интегрированной средой разработки с встроенными отладчиком и профайлером. Язык программирования поддерживает как целые и вещественные типы переменных, так и строковые и булевы. Более того, тип переменной определяется автоматически. Поскольку в основе MATLAB лежит

									Лист
									45
Изм.	Лист	№ докум.	Подп.	Дат.					

матричный способ представления данных, любая переменная хранится в памяти системы, как матрица соответствующей размерности. MATLAB поддерживает все базовые арифметические операции. Определена операция возведения в степен. Также имеется возможность поэлементных умножения, деления и возведения в степень. Имеется возможность выполнения операций над комплексными числами с помощью зарезервированного имени мнимой единицы i или j . При работе с комплексными числами можно воспользоваться некоторыми специальными функциями, такими как взятие действительной и мнимой части, определение комплексно-сопряженного числа, определение модуля и аргумента комплексного числа.

Также MATLAB содержит ряд встроенных математических функций: определение натурального и десятичного логарифма, вычисление логарифма с основанием 2, тригонометрические функции, округление, вычисление модуля, генерация псевдослучайных чисел с различными законами распределения. Почти все элементарные функции допускают возможность их использования для выполнения операций над комплексными числами. Список всех элементарных функций можно получить путем ввода в командную строку запроса `elfun`.

Поскольку любое число в MATLAB представляет с собой матрицу, состоящую из одного элемента, применение элементарных операций и функций возможно и для массивов. Помимо этого, для матриц существуют некоторые специальные функции, такие как определение размера массива, выбор элемента массива, заполнение элементов массива нулями, единицами, псевдослучайными числами, транспонирование и нахождение обратной матрицы.

Большинство полноценных программ подразумевает использование условных операторов и циклов. Условный оператор представляет собой некоторый пункт выбора возможного направления движения в зависимости от выполнения условий. При этом программа может как выполнять одно или несколько действий, так и не выполнять вообще. Составные условия в MATLAB реализуются с помощью логических операторов И, НЕ, ИЛИ. В случае необходимости использования

вложенных друг в друга условных оператора if, условия в которых проверяются относительно одной переменной, разумнее использовать оператор switch. Этот оператор выполняет действие в зависимости от того, какое условие выполнилось.

В системе MATLAB существует три типа циклов: с предусловием, с постусловием и с параметром. В первом случае сначала задается условие входа в цикл, которое проверяется перед каждой итерацией. Во втором – сначала выполняются операции тела цикла, а потом проверяется условие для повторного входа. Цикл с параметром используется в случае известного числа шагов.

Визуализация данных в MATLAB очень разнообразна. Использованием внутреннего синтаксиса MATLAB позволяет выводить данные в виде двумерных и трехмерных графиков в декартовых и полярных координатах. MATLAB позволяет управлять цветовой палитрой, создавать анимацию результатов, снабжать графики цветовыми картами. Также есть возможность наложения графиков друг на друга с целью их сравнения.

Файлы программ в системе MATLAB (m-файлы) бывают двух типов: файл-функции и файл-программы (Script-файлы). Функции, как правило, имеют входные и выходные аргументы, выполняют одну операцию и вызываются другой программой или пользователем из командной строки. Файл-программы, в свою очередь, представляют собой последовательность программ, которую можно вызвать самостоятельно. Любые m-файлы имеют ряд особенностей синтаксиса. Так операторы отделяются друг от друга точкой с запятой, каждый оператор записывается с новой строки, комментарии записываются после знака «%», в программах отсутствует оператор, указывающий на окончание программы, но существует оператор, указывающий на окончание условных операторов и циклов. Существует возможность общения с пользователем посредством ввода данных в программу.

В MATLAB существует понятие области видимости переменных. Все переменные видимы только в пределах функций или программ, в которых они были определены. Переменную можно сделать видимой в пределах другой функции с

помощью ключевого слова `global`. Оно говорит, что переменная глобальная, то есть уже была объявлена ранее в другом файле, и ее объявление не требуется в данной функции. Однако необходимо предварительно объявить глобальную переменную в другом файле и не допускать повторения имен глобальной переменной в нескольких.

3.2 Решение однородных дифференциальных уравнений

Существует большое количество методов для решения ОДУ и систем ОДУ. Методы решения ОДУ подразделяются на аналитические, приближенные и численные.

Аналитические методы удобны тем, что позволяют получать решения оду при помощи элементарных или специальных функций, а также тем, что реализованы в большинстве математических и прикладных пакетов. Главным минусом аналитических методов можно считать ограниченную область применения данной группы методов, в которую главным образом входят линейные ОДУ и ОДУ с постоянными коэффициентами. При решении практических задач аналитические методы малоэффективны.

Более эффективными можно считать приближенные методы решения ОДУ. В основе данной группы методов лежат методы упрощения исходных ОДУ, такие как линеаризация, разложения в ряды. Область применения подобных методов также ограничена.

Наиболее часто для решения ОДУ применяются численные методы. Они представляют собой мощный и универсальный инструмент, применение которого возможно даже, когда традиционные методы малоэффективны.

Для численного решения задачи Коши для ОДУ MATLAB предоставляет различные функции, называемые также решателями или солверами (анг. *solver*). При выборе необходимого солвера, следует учитывать, что решатели ОДУ основаны на различных методах (метод Рунге-Кутты, метод Адамса-Башворт-Мултона, метод Розенброка, метод трапеций с интерполяцией) и имеют различную

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						48
Изм.	Лист	№ докум.	Подп.	Дат.		

точность вычисления. Поэтому необходимо выбирать солвер с учетом типа решаемой задачи и необходимой скорости и точности вычислений.

Рассмотрим основные солверы ОДУ [27, с. 419]. При невысоких требованиях к точности решения задачи можно применять солвер ode23. В его основу заложены явные методы Рунге-Кутты 2-го и 4-го порядков в следующих модификациях: Богацки и Шампина. Методы считаются одношаговыми, поскольку при их использовании для вычисления значения в некоторой точке требуется информация только о значении в предыдущей рассчитанной точке. Плюсами решателя можно считать его высокую скорость при соблюдении условий его применения: данный солвер рекомендуется применять для решения систем ОДУ умеренной жесткости.

Также хорошие результаты позволяет получить солвер ode45. В его основу заложены явные методы Рунге-Кутты 4-го и 5-го порядков в следующих модификациях: Дорманда и Принца. Методы, аналогично методам, применяемым в ode23, являются одношаговыми. Решать часто применяется для грубой оценки решения и показывает хорошие результаты при применении солвера для нежестких систем.

При невысоких требованиях к точности решения задачи для жестких систем ОДУ можно применять солвер ode23s. В его основу заложен явный метод с модифицированной формулой Розенброка 2-го порядка. Метод, аналогично методам, применяемым в ode23, является одношаговым. Плюсами решателя можно считать его высокую скорость при соблюдении условий его применения.

В случаях, когда солвер ode45 не позволяет найти решение системы ОДУ, можно использовать ode15s. Этот решатель основан на адаптивном методе, порядок которого является переменным. По умолчанию порядок метода – 5, однако он изменяется от 1 до 5. Метод, заложенный в основу решателя, использует численное «дифференцирование назад». Метод является многошаговым, то есть для нахождения решения в некоторой точке требуется информация не только о значении в предыдущей рассчитанной точке, но и на нескольких предыдущих «шагах». Основная область применения данного солвера – жесткие системы ОДУ.

									Лист
									49
Изм.	Лист	№ докум.	Подп.	Дат.					

Для ОДУ, описывающих системы колебательного вида, выходной сигнал которых можно считать почти колебательным, лучше всего подходит солвер ode23t. В основе этого солвера лежит метод трапеций с интерполяцией. Этот метод является неявным. Одной из особенностей неявных методов является их хорошая применимость к жестким системам. При решении систем умеренной жесткости солвер позволяет получить высокоточное решение.

Решатель, применяемый в случаях, когда требуется высокоточное решение систем ОДУ, – ode113, использующий многошаговый адаптивный метод Адамса–Башворта–Мултона переменного порядка.

Решатель ode23tb включает в себя совокупность различных методов: в начале решения используется неявный метод Рунге–Кутта и далее применяется метод «дифференцирования назад» 2-го порядка. Решатель ode23tb в некоторых случаях может быть эффективнее, чем ode15s, несмотря на относительно низкую точность метода.

Решатель bvp4c применяется для полной формы систем уравнений Коши, то есть для двухточечных краевых задач.

Для ОДУ в частных производных, описывающих системы параболического или эллиптического вида, лучше всего подходит солвер pdepe. Поскольку программа необходима для проверки возможности использования алгоритма коррекции, обеспечение высокой точности решения ОДУ было бы неоправданной тратой ресурсов. Поэтому был выбран решатель ode23 с наиболее быстрой сходимостью.

Как уже было указано выше, численные методы решения задачи Коши для ОДУ подразделяются на одношаговые и многошаговые методы. Методы Рунге–Кутта относятся к одношаговым методам и являются одними из самых часто используемых методов численного решения задачи Коши. Основой данной группы методов служит аппроксимация исследуемой функции $y(x)$ на каждом «шаге» h . Аппроксимация производится путем разложения этой функции в ряд Тейлора с некоторой точностью во всех точках i вблизи шага h .

									Лист
									50
Изм.	Лист	№ докум.	Подп.	Дат.					

Различные методы, входящие в группу методов Рунге-Кутты, получали путем отбрасывания разного количества членов ряд Тейлора. Чем меньше количество членов отбрасывается, тем меньше погрешность интегрирования. Существуют методы Рунге-Кутты разной степени точности в зависимости от максимальной учитываемой степени ряда Тейлора. Рассмотрим методы Рунге-Кутты второго и четвертого порядков.

Второй порядок характеризуется следующим соотношением:

$$y(x_0 + h) = y[(1 - \mathcal{L}) \cdot f_0 + \mathcal{L} \cdot f(x_0 + \gamma \cdot h, y_0 + \gamma \cdot f_0 \cdot h)] + O(h^3), \quad (53)$$

где $0 < \mathcal{L} \leq 1$ – свободный параметр;

$$f_0 = f(x_0, y_0);$$

$$\gamma = (2\mathcal{L})^{-1}.$$

Метод Рунге-Кутты второго порядка обладает локальной погрешностью вычисления третьего порядка и глобальной – второго. При этом обеспечивается сходимость решения уравнений с погрешностью $O(h^2)$.

Чаще всего в расчетах используется значение параметра \mathcal{L} равное 0,5 или 1.

На рисунке 11 представлена геометрическая интерпретация.

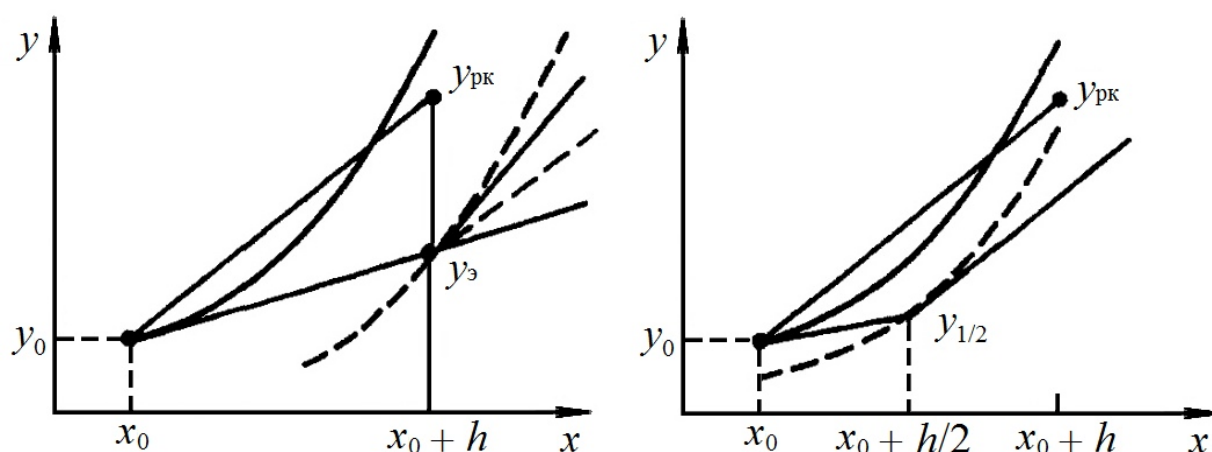


Рисунок 11 – Геометрическая интерпретация для $\mathcal{L} = 0,5$ (слева) и $\mathcal{L} = 1$ (справа)

Для параметра \mathcal{L} наиболее часто используются значения 0,5 и 1. При $\mathcal{L} = 0,5$ уравнение 6.14 выглядит следующим образом:

$$y(x_0 + h) = y_0 + h[\cdot f_0 + f(x_0 + h, y_0 + f_0 \cdot h)] / 2. \quad (54)$$

Первым этапом данного метода является вычисление приближенного значения ОДУ в точке $x_0 + h$. Вычисление производится по формуле Эйлера $y_3 = y_0 + hf_0$. На втором этапе используется уже найденная точка f с координатами $(x_0 + h, y_3)$. В ней определяется наклон интегральной кривой.

Затем определяется средний наклон интегральной прямой в найденной точке $f(x_0 + h, y_3)$, после чего определяется уточненное значение $y_{рк} = y(x_0 + h)$. Алгоритмы данного вида – приближенное вычисление при помощи формул низких порядков, после которого производится уточнение, основанное на полученной на предыдущем шаге информации, – принято называть алгоритмами «прогноз-коррекции».

Метод Рунге-Кутты 2-го порядка основан на отбрасывании всех членов ряда Тейлора, которые включают в себя степени h свыше второй. Аналогично, метод Рунге-Кутты 4-го порядка учитывает члены ряда Тейлора, содержащие степени h по четвертую включительно.

Существует несколько модификаций метода Рунге-Кутты для 4-го порядка, однако классическим, то есть наиболее часто применяемым при решении практических задач, вариантом метода считается метод, основанный на формуле следующего вида:

$$y(x_0 + h) = y_0 + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} + O(h^5), \quad (56)$$

где $k_1 = h f(x_0, y_0)$;

$k_2 = h f(x_0 + h/2, y_0 + k_1/2)$;

						Лист
						52
Изм.	Лист	№ докум.	Подп.	Дат.		

$$k_3 = h f\left(x_0 + h/2, y_0 + k_2/2\right);$$

$$k_4 = h f(x_0 + h, y_0 + k_3).$$

Применение этого метода подразумевает, что на каждом шаге h производится вычисление значения правой части ОДУ в четырех точках.

Метод Рунге-Кутты четвертого порядка обладает локальной погрешностью вычисления пятого порядка и глобальной – четвертого.

Формулу, на основе которой реализуется метод Рунге-Кутты четвертого порядка для программной реализации как правило преобразуют к следующему виду:

$$y_i(x_0 + h) = y_{i0} + \frac{q_{i1} + 2q_{i2} + 2q_{i3} + q_{i4}}{3} + O(h^5), \quad (57)$$

где $q_{i1} = h/2 f_i(x_0, y_{i0})$;

$$q_{i2} = h/2 f_i\left(x_0 + h/2, y_0 + q_{i1}/2\right);$$

$$q_{i3} = h/2 f_i\left(x_0 + h/2, y_0 + q_{i2}/2\right);$$

$$q_{i4} = h/2 f_i(x_0 + h, y_0 + q_{i3}).$$

$i = 1, 2, \dots, n$ – номер уравнения в системе ОДУ из n уравнений

3.3 Выбор метода поиска минимума

Задачи поиска максимума (целевая функция вида f) или минимума (целевая функция вида $-f$) функции одной либо нескольких переменных получили название задач оптимизации.

Для одномерной оптимизации могут применяться следующие методы: градиентные (спуска, проекции градиента, субградиента, условного градиента, сопряженных направлений, наискорейшего спуска, сопряженных градиентов), возможных направлений, линеаризации, случайного поиска, Ньютона,

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						53
Изм.	Лист	№ докум.	Подп.	Дат.		

Стеффенсена, покоординатного спуска, оврагов, методы функций (штрафных, барьерных, модифицированных Лагранжа, нагруженных), прямого поиска.

Рассмотрим особенности некоторых из них.

Метод парабол. Все вариации метода парабол требуют внесения поправок к алгоритму. Так, например, во время вычислений необходимо проверять, что алгоритм движется к минимуму. В обратном случае придется сделать достаточно большой шаг в обратном направлении.

Чем тщательнее продуманы поправки, тем выше скорость программы на основе такого алгоритма. У метода есть серьезный недостаток: при наличии нескольких локальных минимумов возникает вероятность сходимости к минимуму, не являющемуся глобальным. Найденные минимумы нельзя удалять, поэтому необходимо сделать несколько приближений на разных участках, а затем, сравнив найденные минимумы, выбрать наименьший. Точность метода зависит от способа деления функции на участки.

Еще одним малозатратным методом является покоординатный спуск. Его достоинствами можно считать простоту программирования метода и легкость в использовании. К недостаткам метода относятся в первую очередь низкая скорость нахождения минимума и плохая сходимость при наличии оврагов. Метод чаще всего применяется для нахождения грубого приближения искомого минимума.

Метод оврагов определяет лишь приблизительное значение минимума и требует уточнения. Данный метод удобен для нахождения минимума сложных функций (в том числе нескольких переменных). Метод требует индивидуального подбора овражного шага для каждой функции, а также внесения поправок в ходе расчета.

Метод случайного поиска подходит как для функций единственной переменной, как и для многомерной оптимизации сложных функций, зависящих от десяти – двадцати переменных. Несомненным преимуществом метода является то, что при применении данного метода к функциям одной переменной значительно сокращается количество необходимых случайных точек. Однако необходимо

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						54
Изм.	Лист	№ докум.	Подп.	Дат.		

тщательно выбирать границы исследуемой области, чтобы не потерять локальные минимумы и не сильно усложнить решение задачи оптимизации.

Методами спуска называются итерационные, то есть приближенные методы, применяемые для решения задачи минимизации функций одной или нескольких переменных. Эта группа методов имеет обширную область применения, которая ограничивается следующей особенностью: определяемый через данные методы экстремум функции будет являться локальным. Таким образом лучше всего эти методы работают для исследования функций, локальный минимум которых является и глобальным.

При решении задачи оптимизации в применении к поиску минимума одним из малозатратных считается метод золотого сечения (из группы методов прямого поиска). К достоинствам этого метода следует в первую очередь отнести хорошую сходимость даже при использовании метода на недифференцируемых функциях. Сходимость метода является линейной. При наличии у функции на некотором отрезке нескольких минимумов данный метод обеспечивает сходимость к одному из них, хотя и нет гарантий, что к глобальному.

Золотое сечение часто используют в общественных и технических областях. Это связано с невозможностью дифференцирования некоторых исследуемых функций, а также с высокими затратами на проведение экспериментов для вычислений функций.

К недостаткам метода следует отнести ограниченность применения только детерминированными задачами, а также невысокую скорость метода.

Наибольшей скоростью сходимости для класса непрерывных функций обладает метод поиска минимума Фибоначчи, поскольку обладает переменным коэффициентом сжатия интервала неопределенности. Еще одним достоинством можно считать задание точности метода заранее, от которой зависит число итераций. Недостатком метода можно считать потерю устойчивости при большом числе итераций.

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						55
Изм.	Лист	№ докум.	Подп.	Дат.		

Для определения минимума оценки динамической погрешности будем использовать метод прямого поиска минимума Фибоначчи.

3.4 Подстройка параметров динамической системы с помощью метода Фибоначчи

Коэффициент подстройки k_0 оказывает большее влияние на полосу пропускания динамической измерительной системы, чем коэффициент подстройки k_1 .

Поэтому свяжем настраиваемые параметры передаточной функции (24) с коэффициентом подстройки k_0 и минимизируем функцию оценки динамической погрешности как функцию одной переменной.

Тогда структурная схема самонастраивающейся модели примет следующий вид (рисунок 12).

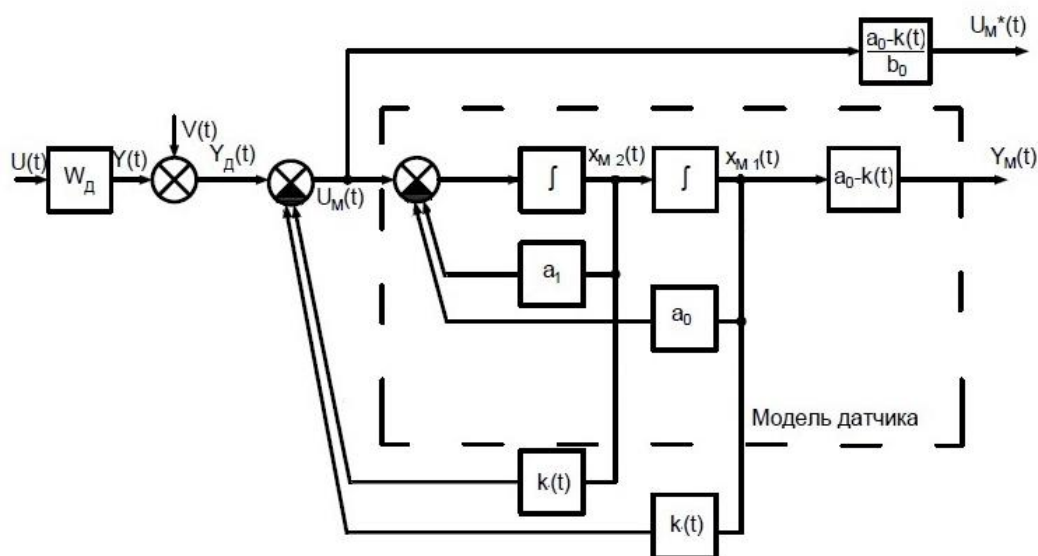


Рисунок 12 – Структурная схема самонастраивающейся модели динамической измерительной системы

Метод Фибоначчи был выбран для определения параметра k при котором обеспечивалась бы минимальная оценка динамической погрешности из-за высокой скорости сходимости и возможности заранее задать точность нахождения минимума.

В явном виде последовательность Фибоначчи определяется формулой Бине:

$$F_n = \frac{\varphi^n - (-\varphi)^{-n}}{2\varphi - 1}, \quad (58)$$

где $\varphi = \frac{1+\sqrt{5}}{2}$ – золотое сечение.

Процедура поиска с использованием чисел Фибоначчи требует два вычисления функции на первом шаге, а на каждом последующем – только по одному [25, с.53]. В данном методе изначально определяется число итераций n . Также сокращение интервала неопределенности имеет коэффициент сжатия, который на некотором шаге l равен F_{n-l} / F_{n-l+1} .

Конечный интервал неопределенности tol (точности определения минимума) задается заранее и связан с числом итераций n (в неявном виде) следующим соотношением:

$$F_n = \frac{k_1 - k_0}{tol}. \quad (59)$$

Первые два вычисления проводятся в точках:

$$\begin{aligned} G_1 &= k_0 + \frac{F_{n-2}}{F_s} (k_1 - k_0) \\ G_2 &= k_0 + \frac{F_{n-1}}{F_s} (k_1 - k_0) \end{aligned} \quad (60)$$

расположенных симметрично относительно середины отрезка $[k_0, k_1]$. Если $f(G_1) < f(G_2)$, то новым отрезком локализации минимума является $[k_0, G_2]$, в случае $f(G_1) \geq f(G_2)$ – $[G_1, k_1]$. Каждая следующая точка выбирается симметрично относительно середины полученного отрезка к лежащей внутри этого отрезка точке уже проведенного вычисления (G_1 или G_2). При этом любая внутренняя точка делит

									Лист
									57
Изм.	Лист	№ докум.	Подп.	Дат.	ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР				

отрезок локализации в отношении, двух последовательных чисел Фибоначчи [26, с. 40].

После $(n-2)$ -го шага точка проведенного вычисления оказывается в середине отрезка локализации. Точка последнего, n -го, шага выбирается на расстоянии α от середины этого отрезка, где α – фиксированное малое положительное число (константа различимости). Константа различимости выбирается заранее существенно меньше заданной точности.

Выводы по разделу три

Глава содержит описание программной среды MATLAB, встроенного в нее языка программирования, основных функций и инструментов, необходимых программирования и моделирования. Описаны способы решения ОДУ с помощью встроенных в MATLAB солверов. Рассмотрен алгоритм нахождения решения задачи Коши методом Рунге-Кутты 2 и 4 степени. Произведен сравнительный анализ методов поиска минимума с целью выбора наиболее оптимального для данной задачи. Приведено описание метода прямого поиска минимума Фибоначчи.

									Лист
									58
Изм.	Лист	№ докум.	Подп.	Дат.	ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР				

4. МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКОЙ ИЗМЕРИТЕЛЬНОЙ СИСТЕМЫ

4.1 Описание программы

Опишем программу в соответствии с ГОСТ 19.402-78 «ЕСПД. Описание программы» [28].

4.1.1 Общие сведения

Наименование программы: «Алгоритм коррекции динамической погрешности».

Программное обеспечение, необходимое для функционирования программы: программная среда MATLAB. Язык программирования, на котором написана программа: язык программирования MATLAB.

4.1.2 Функциональное назначение

Программа предназначена для анализа возможности коррекции динамических погрешностей выходных сигналов датчика 2 порядка на основе метода прямого поиска Фибоначчи. Обработка происходит после выполнения измерений (апостериорная).

4.1.3 Описание логической структуры

Структура алгоритма коррекции динамической погрешности представлен на рисунках 13–15.

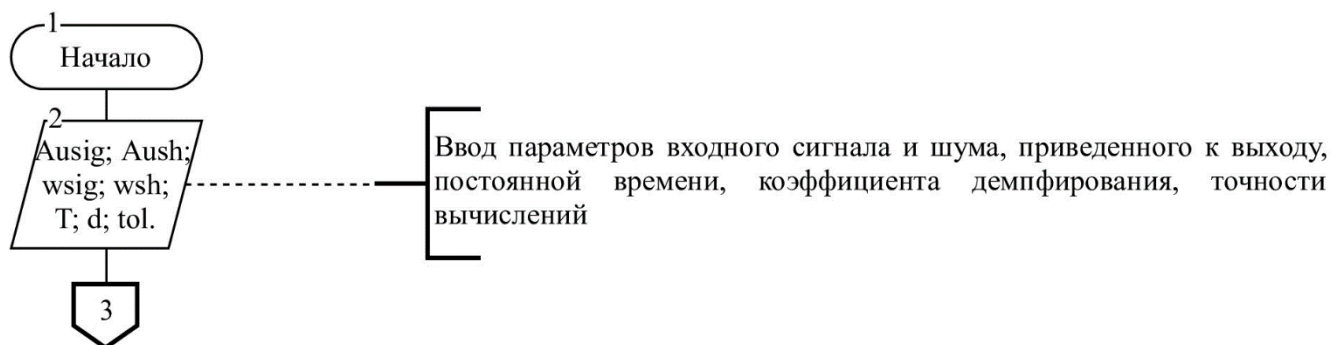


Рисунок 13 – Структура алгоритма коррекции динамической погрешности

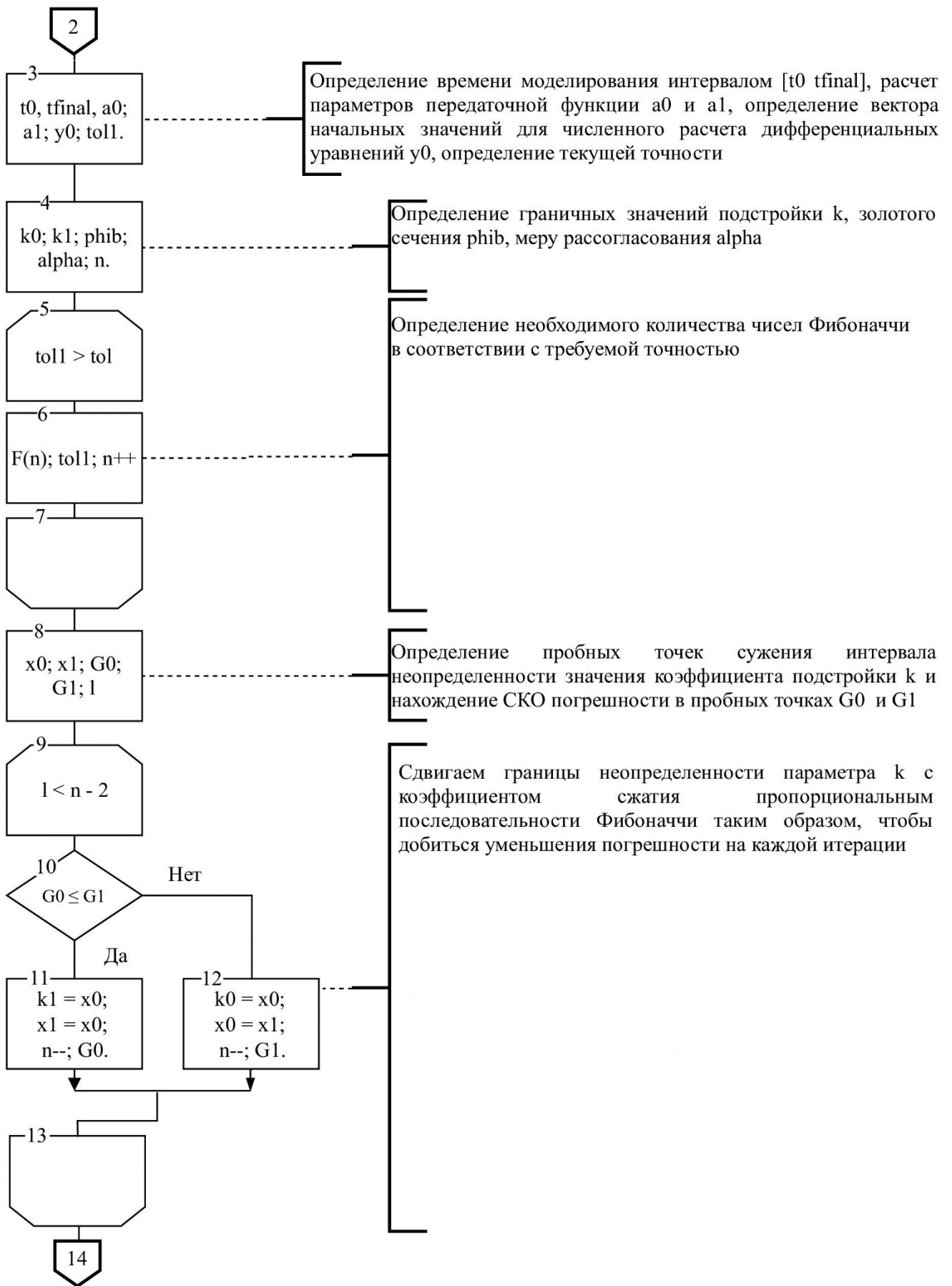


Рисунок 14 – Структура алгоритма коррекции динамической погрешности

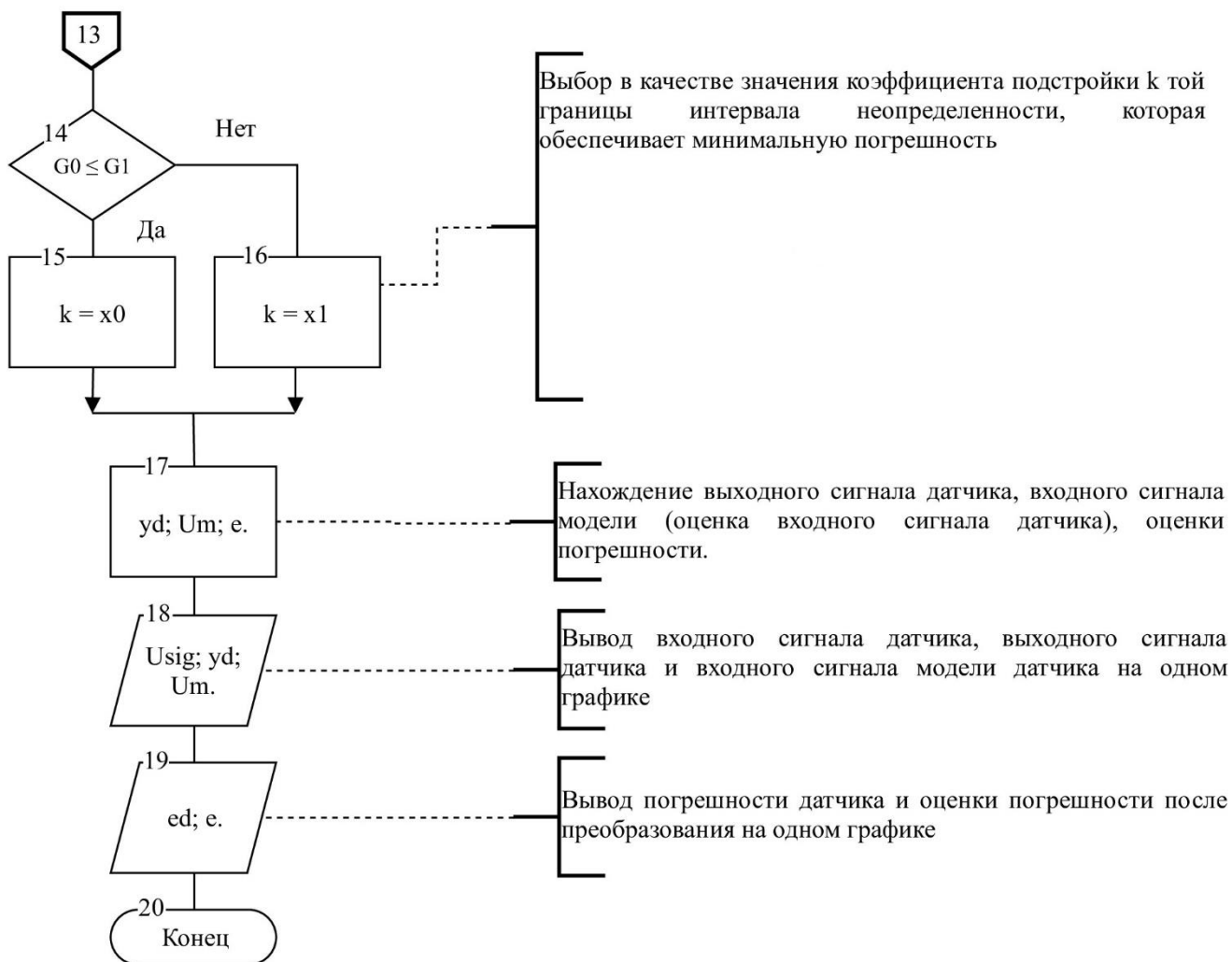


Рисунок 15 – Структура алгоритма коррекции динамической погрешности

Листинг программы приведен в приложении А.

4.1.4 Используемые технические средства

Программа эксплуатируется на персональном компьютере (ПК). Режим работы – в форме диалога оператора и файл-программы.

Для работы используется экран дисплея, клавиатура и манипулятор типа «мышь».

Входные данные вводятся пользователем, выходные данные выводятся в виде графиков.

Требования к ПК:

- любая версия ОС Windows;
- свободное дисковое пространство не менее 1 ГБ;
- оперативная память не менее 1 ГБ (рекомендовано 2 ГБ).

4.1.5 Входные данные

Входной сигнал датчика в виде дискретных отсчетов, входное шумовое воздействие в виде дискретных отсчетов, точность вычислений, параметры датчика (коэффициент демпфирования, постоянная времени).

4.1.6 Выходные данные

График входного и выходного сигналов датчика в относительных единицах от времени в секундах, график восстановленного сигнала в относительных единицах от времени в секундах, график абсолютной погрешности от времени в секундах.

4.2 Результаты моделирования

Проведем цифровое моделирование измерительной системы второго порядка с подстройкой по динамической погрешности коэффициентов обратных связей модели датчика в среде MATLAB для двух видов входных воздействий.

В качестве входных информативного и шумового воздействий будем использовать одночастотные гармонические сигналы.

Для наглядности будем использовать одинаковые параметры датчика при проведении моделирований с различными входными сигналами: коэффициент демпфирования 0,7 и постоянную времени 0,01.

На вход датчика подадим гармонический сигнал вида: $U_{sig}(t) = 1 \cdot \sin(20 \cdot t)$. На выход датчика подадим шумовое воздействие в виде гармонического сигнала с меньшей амплитудой и высокой частотой: $V(t) = 0,02 \cdot \sin(500 \cdot t)$.

									Лист
									62
Изм.	Лист	№ докум.	Подп.	Дат.					

На рисунке 17 приведем сигналы с входа и выхода датчика.

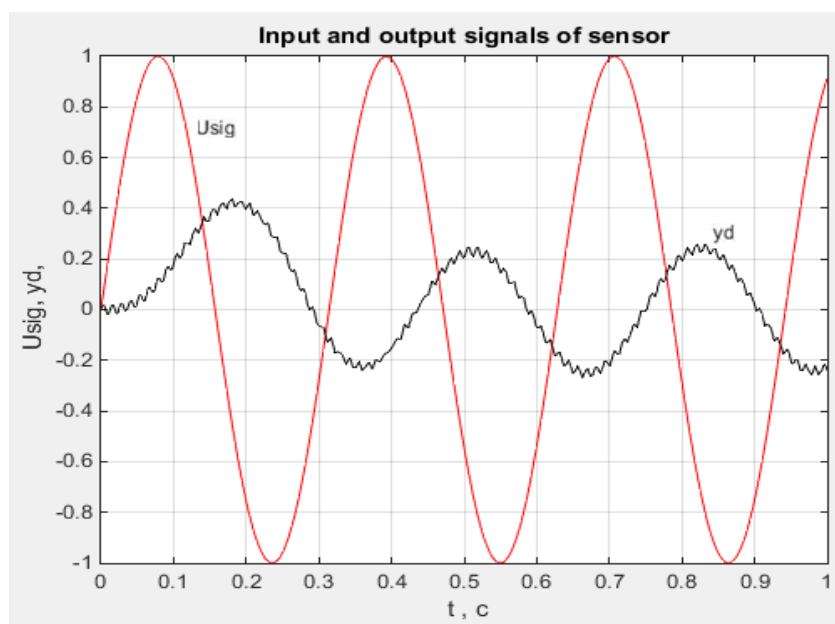


Рисунок 16 – Входной и выходной сигналы датчика

На графике заметно, что выходной сигнал датчика сдвинут по фазе относительно входного примерно на 0,1 с.

Определим разницу между входным и выходным сигналами датчика, то есть его погрешность. График погрешности датчика представлен на рисунке 17.

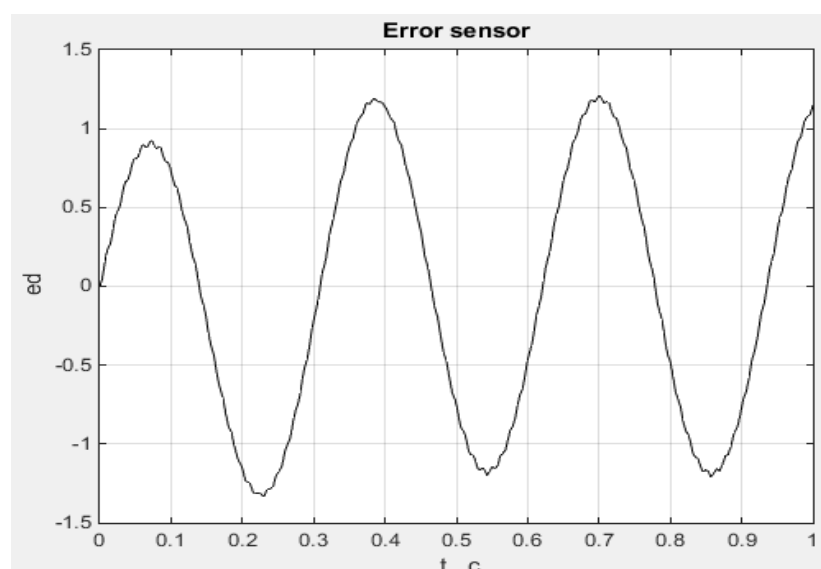


Рисунок 17 – Погрешность датчика

Изм.	Лист	№ докум.	Подп.	Дат.

Из графика видно, что погрешность датчика по амплитуде сравнима с входным сигналом датчика. Подадим сигнал с выхода датчика на вход модели датчика, охваченной обратными связями с коэффициентами подстройки $K(t)$. Алгоритм подстройки параметров $K(t)$, в котором применяется прямого поиска минимума динамической погрешности, описанный в третьей главе, добивается равенства выходных сигналов с выхода датчика и выхода модели датчика. Восстановленный сигнал представим на рисунке 18.

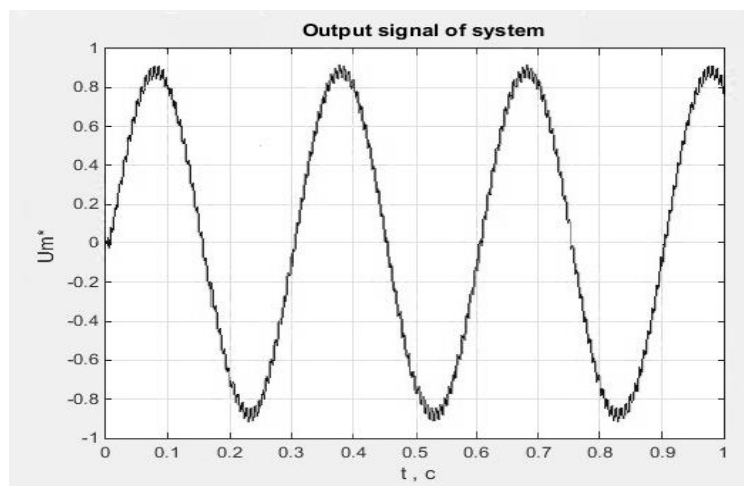


Рисунок 18 – Восстановленный сигнал

Согласно выражению (50) в качестве оценки погрешности измерительной системы можно использовать разницу между выходными сигналами датчика и модели. На рисунке 19 представим изменение графики динамической погрешности до и после применения алгоритма.

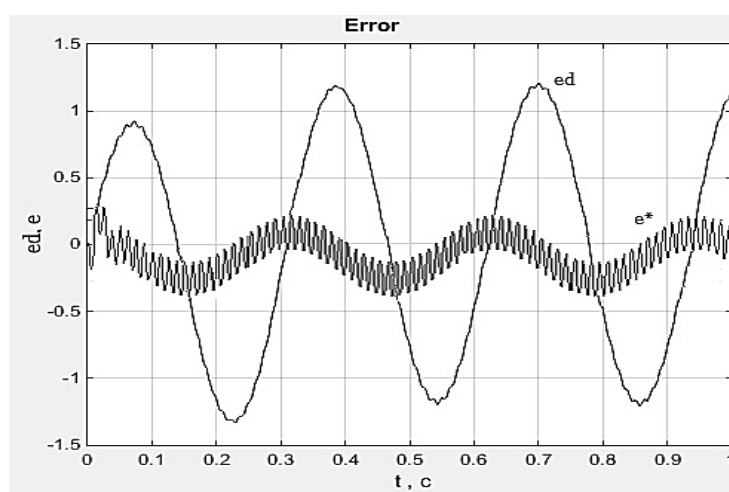


Рисунок 19 – Оценка погрешности датчика и измерительной системы

Из графика видно, что происходит снижение динамической погрешности в 5 раз.

Для наглядности представим отдельно график погрешности измерительной системы после применения алгоритма коррекции в большем масштабе (рисунок 19).

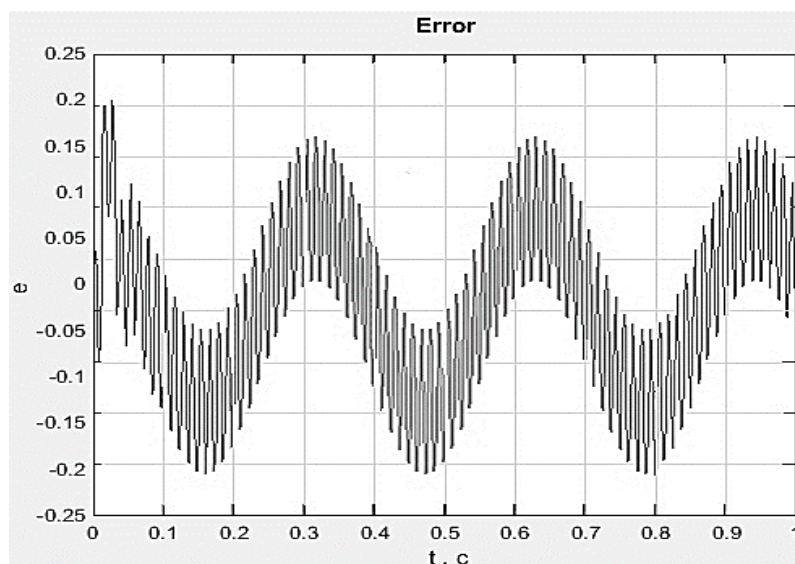


Рисунок 19 – Сигнал погрешности измерительной системы

На вход датчика подадим гармонический сигнал вида: $U_{sig}(t) = 0,8 \cdot \sin(10 \cdot t)$. На выход датчика подадим шумовое воздействие в виде гармонического сигнала с меньшей амплитудой и высокой частотой: $V(t) = 0,01 \cdot \sin(400 \cdot t)$. На рисунке 21 приведем сигналы с входа и выхода датчика.

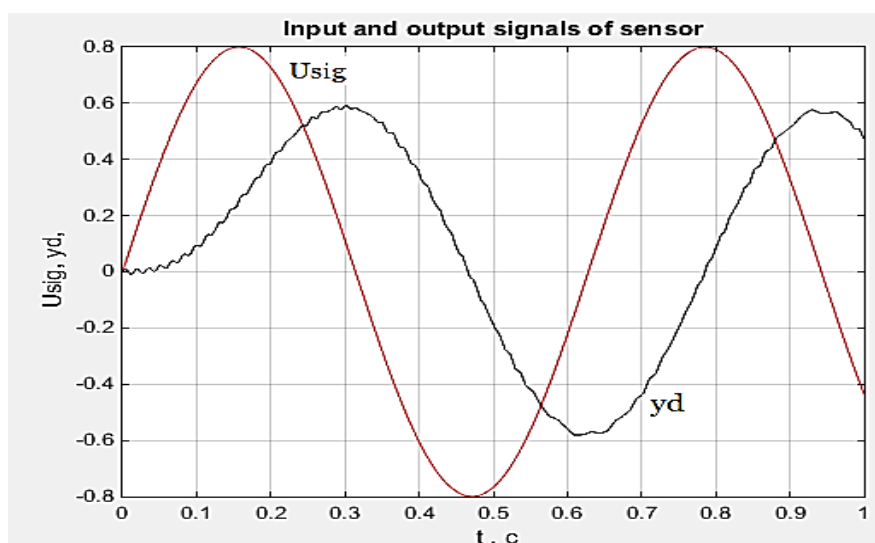


Рисунок 21 – Входной и выходной сигналы датчика

Изм.	Лист	№ докум.	Подп.	Дат.

Из графика видно, что сигнал на выходе датчика отстает по фазе от входного сигнала датчика на 0,1 с.

Определим разницу между входным и выходным сигналами датчика, то есть его погрешность. График погрешности датчика представлен на рисунке 22.

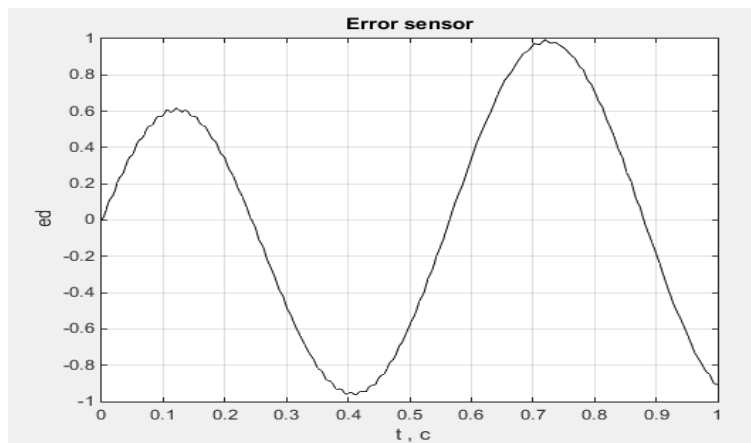


Рисунок 22 – Погрешность датчика

Из графика видно, что погрешность датчика по амплитуде сравнима с входным сигналом датчика. Подадим сигнал с выхода датчика на вход модели датчика, охваченной обратными связями с коэффициентами подстройки $K(t)$. Алгоритм подстройки параметров $K(t)$, в котором применяется прямого поиска минимума динамической погрешности, описанный в третьей главе, добивается равенства выходных сигналов с выхода датчика и выхода модели датчика. Восстановленный сигнал представим на рисунке 23.

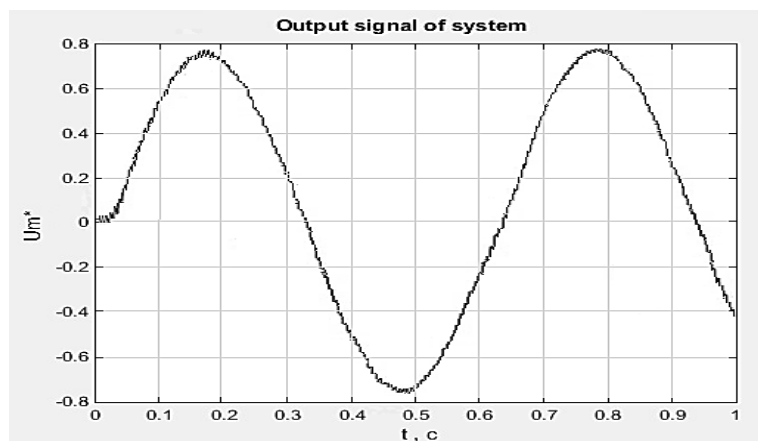


Рисунок 23 – Восстановленный сигнал

Согласно выражению (50) в качестве оценки погрешности измерительной системы можно использовать разницу между выходными сигналами датчика и модели. На рисунке 22 представим изменение графики динамической погрешности до и после применения алгоритма.

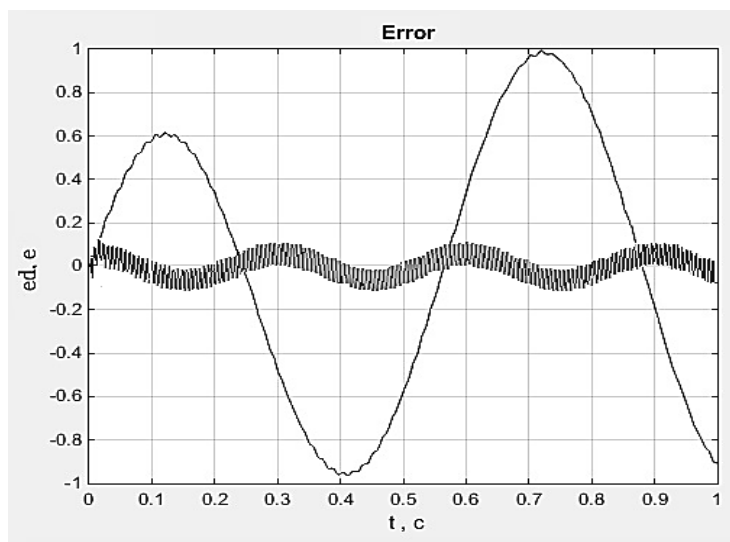


Рисунок 22 – Оценка погрешности датчика и измерительной системы

Из графика видно, что происходит снижение динамической погрешности в 5 раз. Для наглядности представим график погрешности измерительной системы после применения алгоритма коррекции (рисунок 23).

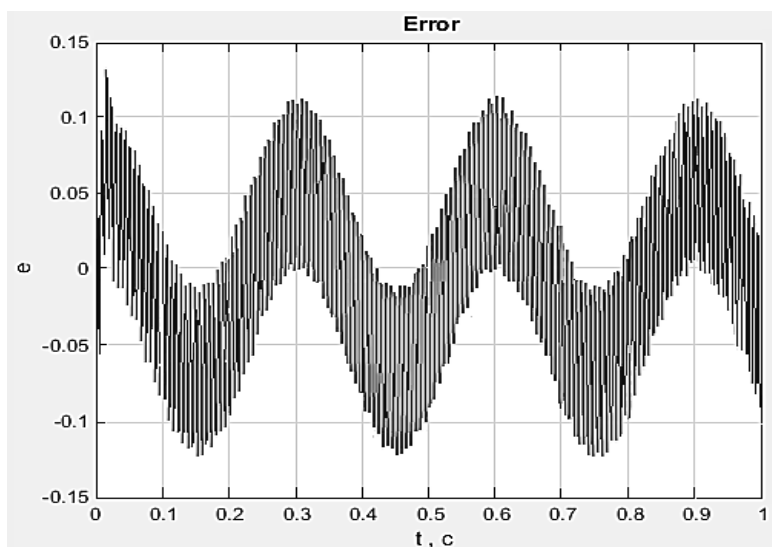


Рисунок 23 – Сигнал погрешности измерительной системы после применения алгоритма коррекции

Выводы по разделу четыре

Глава содержит описание программы в соответствии с ГОСТ 19.402-78, в том числе назначение программы, ее структуру, входные и выходные данные и системные требования, необходимые для работы программы. Также проведено цифровое моделирование с двумя наборами входных сигналов. На основе результатов цифрового моделирования можно сделать вывод о возможности применения алгоритма для коррекции динамической погрешности.

					<i>ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР</i>	<i>Лист</i>
						68
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дат.</i>		

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была приведена общая теория коррекции динамической погрешности. Представлено математическое описание динамических ИС с модельным управлением. Доказана возможность снижения динамической погрешности посредством изменения настраиваемых параметров ОС модели датчика.

Произведено описание ИС для первичного преобразователя второго порядка. Обоснована замена всех настраиваемых параметров ОС на параметр k для сведения задачи многомерной оптимизации к одномерной. Выбран метод Фибоначчи в качестве метода поиска минимума динамической погрешности ИС от настраиваемого параметра ОС на основе требований по быстродействию алгоритма.

Написана программа, предназначенная для проверки работоспособности алгоритма коррекции динамической погрешности для ИС второго порядка в среде MATLAB на встроенном языке программирования.

Реализовано цифровое моделирование измерительной системы с передаточной функцией второго порядка. Оно показало возможность снижения динамической погрешности с помощью алгоритма на основе метода Фибоначчи в 5 раз. На основе результатов цифрового моделирования можно сделать вывод об эффективности применения алгоритма коррекции динамической погрешности на основе метода Фибоначчи.

						Лист
					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	69
Изм.	Лист	№ докум.	Подп.	Дат.		

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Полуэктов, Р.А. Методы коррекции динамических погрешностей / Р.А. Полуэктов, Г.Н. Солопченко // Автометрия. – 1971. – № 5., Грановский, В.А. Динамические измерения: Основы метрологического обеспечения / В.А Грановсктй. – Л.: Энергоатомиздат, 1984. – 224 с.
2. Грановский, В.А. Динамические измерения. Основы метрологического обеспечения / В.А. Грановский. – Ленинград: Энергоатомиздат, 1984. – 314 с.
3. ГОСТ 19781-90 «Обеспечение систем обработки информации программное. Термины и определения».
4. Бахтизин, В.В. Технология разработки программного обеспечения: учеб. пособие / В. В. Бахтизин, Л. А. Глухова. – Минск: БГУИР, 2010. – 267 с.
5. Калайда, В.Т. Технология разработки программного обеспечения: Учебное пособие. / В.Т. Калайда, В.В. Романенко. – Томск: Томский межвузовский центр дистанционного образования, 2007. – 257 с.
6. Зиглер, К. Методы проектирования программных средств / К. Зиглер. – М.: Мир, 1985. – 328 с.
7. МИ 2174-91 ГСИ. «Аттестация алгоритмов и программ обработки данных при измерениях. Основные положения».
8. Кожемякин, С.Д. Реализация алгоритма коррекции динамической погрешности самонастраивающейся измерительной системы на базе суперкомпьютерного центра ЮУрГУ / С.Д.Кожемякин // Пром-инжиниринг: труды II международной научно-технической конференции. – Челябинск: Издательский центр ЮУрГУ, 2016. – С. 469–474.
9. Кузнецов, Ф.И. Минимизация динамических погрешностей в датчиковых системах мониторинга и управления в реальном масштабе времени / Ф.И. Кузнецов // Известия ЮФУ. Серия «Технические науки». – 2014. – С. 63–69.

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						70
Изм.	Лист	№ докум.	Подп.	Дат.		

10. Volosnikov, A.S Dynamic Measurement Error Evaluation and Minimization based on FIR-filter / A.S. Volosnikov, E.V. Yurasova // Proceedings – 2018 Global Smart Industry Conference, GloSIC 2018. – 2018.

11. Mei, Z. Analysis of CMM Dynamic Measurement Error Based on Kernel Estimation / Z. Mei // 32nd youth academic annual conference of chinese association of automation. – New York: publisher IEEE, 2017. – P.1162–1166.

12. Jiang, ML. Dynamic measurement errors prediction for sensors based on firefly algorithm optimize support vector machine / L. Jiang, DD. Jiang, JP. Xiong, JG. Shen, SH. Ahmed, JY. Luo, HB. Song // Sustainable cities and society. – 2017. – V. 35. – P. 250–256.

13. Волосников, А.С. Нейросетевая динамическая измерительная система с последовательной обработкой информации / А.С. Волосников // Успехи современного естествознания. – 2005. – № 2. – С. 27-28.

14. Шестаков, А.Л. Модальный синтез измерительного преобразователя / А.Л. Шестаков // Известия Российской академии наук. Теория и системы управления. – 1995. – № 4. – С. 67–75.

15. Бизяев, М.Н. Динамические измерения в скользящем режиме с применением дополнительной фильтрации выходного сигнала / М.Н. Бизяев // Известия Челябинского научного центра. – 2004. №4 (26). – С. 128–133.

16. Бизяев, М.Н. Восстановление динамически искаженных сигналов испытательно-измерительных систем методом скользящих режимов / М.Н. Бизяев, А.Л. Шестаков // Известия Российской академии наук.– 2006. № 6. –119 с.

17. Солдаткина, Е.В. Алгоритмы адаптации параметров измерительной системы к минимуму оценки динамической погрешности: дис. ... канд. техн. наук / Е.В. Солдаткина; ЮУрГУ. – Челябинск, 2000. – 161 с.

18. Юрасова, Е.В. Измерительная система динамических параметров с моделью первичного измерительного преобразователя для контроля входных параметров электроустановок / Е.В. Юрасова // Технические науки – от теории к практике: сб. ст. по матер. XX междунар. науч.-практ. конф. – Новосибирск: СибАК, 2013.

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						71
Изм.	Лист	№ докум.	Подп.	Дат.		

19. Волосников, А.С. Нейросетевые модели и алгоритмы восстановления сигналов динамических измерительных систем: дис. ... канд. техн. Наук / А.С. Волосников; ЮУрГУ. – Челябинск, 2006. – 137 с.

20. Волосников, А.С. Нейросетевая динамическая измерительная система с последовательной обработкой информации / А.С. Волосников // Успехи современного естествознания. – 2005. – № 2. – С. 27–28.

21. Сычев, Е.И. Проблемы технических измерений / Е.И. Сычев // Измерительная техника. – 1995. – № 44. – С. 15–17.

22. Солдаткина, Е.В. Алгоритмы адаптации параметров измерительной системы к минимуму оценки динамической погрешности: дис.... канд. техн. наук / Е.В. Солдаткина. – Челябинск, 2000. – 164 с.

23. Шестаков, А.Л. Методы теории автоматического управления в динамических измерениях: монография / А.Л. Шестаков; Челябинск: Изд. центр ЮУрГУ, 2013. – 256 с.

24. Таранчук, В.Б. Основные функции систем компьютерной алгебры / В.Б. Таранчук. – Минск: БГУ, 2013. – 259 с.

25. Попова, Т.М. Методы одномерной оптимизации / Т.М. Попова. – Хабаровск: Изд-во Тихоокеан. гос. ун-та, 2011. – 216 с.

26. Шипилов, С.А. Методы безусловной одномерной оптимизации / С.А. Шипилов. – Новокузнецк: НФИ КемГУ, 2001. – 240 с.

27. Дьяконов, В.П. MATLAB. Полный самоучитель / В.П. Дьяконов. – М.: ДМК Пресс, 2012. – 768 с.

28. ГОСТ 19.402-78 «ЕСПД. Описание программы».

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						72
Изм.	Лист	№ докум.	Подп.	Дат.		

ПРИЛОЖЕНИЕ А

Скрипт-файлы программы

```
1. % file ka3_e.m
2. global a0 k a1 Ausig Wusig Aush Wush
3. %
4. disp('enter input amplitude, input frequency')
5. Ausig = input('Ausig = '); %Input amplitude
6. Wusig = input('Wusig = '); %Input frequency
7.
8. disp('enter noise amplitude, noise frequency')
9. Aush = input('Aush = '); %Noise amplitude
10. Wush = input('Wush = '); %Noise frequency
11.
12. disp('enter parametrs of sensor: damping ratio, time constant')
13. d = input('d = '); %Damping ratio
14. T = input('T = '); %Time constant
15.
16. disp('enter tolerance')
17. tol = input('tol = ');
18.
19. disp('You can take a tea break')
20.
21. % Simulation time
22. t0 = 0;
23. tfinal = 1;
24. ts = [t0 tfinal];
25.
26. %Coefficients of transfer function of the sensor
```

						Лист
					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	73
Изм.	Лист	№ докум.	Подп.	Дат.		


```

27. a0 = 1/T^2;
28. a1 = 2*d/T;
29.
30. %real tolerance
31. tol1 = 1;
32.
33. %initial value vector
34. y0=[0 0 0 0 0 0];
35.
36. %Adjustable parameters of the measuring system
37. k = -100000;
38. k1 = 0;
39.
40. %Golden ratio
41. phib = (1+sqrt(5))/2;
42. alpha = tol/100;
43.
44. %Calculating Fibonacci numbers
45. n = 0;
46. while (tol1>tol)
47. n=n+1;
48. F(n) = (phib^n - (phib)^n) / (2*phib-1);
49. tol1 = (k1-k0)/F(n);
50. end;
51.
52.
53. %Initial point calculation
54. x0 = k0 + F(n-2) * (k1-k0) / F(n);
55. x1 = k0 + F(n-1) * (k1-k0)/F(n);

```

						Лист
					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	74
Изм.	Лист	№ докум.	Подп.	Дат.		

```

56. l = 1;
57.
58. k = x0;
59. [t,y]=ode45('vdp3', ts, y0, tol);
60. e0 = yd - ym*(k0-a0);
61. e1 = e0 +k* y(:,6)+k* y(:,5);
62. G0 = std(e1);%Finding the error function of the boundary point x0
63.
64. k = x1;
65. [t,y]=ode45('vdp3', ts, y0, tol);
66. e0 = yd - ym*(k0-a0);
67. e1 = e0 +k* y(:,6)+k* y(:,5);
68. G1 = std(e1);%Finding the error function of the boundary point x1
69.
70. while l < (n - 2)
71. %pereopredelenie granichnyh toчек
72. if (G0>G1)
73. k0 = x0;
74. x0 = x1;
75. x1 = k0 + F(n-1-1)*(k1-k0) / F(n-1);
76. k = x0;
77. [t,y]=ode45('vdp3', ts, y0, tol);
78. e0 = yd - ym*(k-a0);
79. e1 = e0 +k* y(:,6)+k* y(:,5);
80. G0 = std(e1);%Finding the error function of the boundary point x0
81. k = x1;
82. [t,y]=ode45('vdp3', ts, y0, tol);
83. e0 = yd - ym*(k-a0);
84. e1 = e0 +k* y(:,6)+k* y(:,5);

```

						Лист
					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	75
Изм.	Лист	№ докум.	Подп.	Дат.		

```

85. G1 = std(e1);%Finding the error function of the boundary point x1
86. else
87. k1 = x1;
88. x1 = x0;
89. x0 = k0 + F(n-1-2)*(k1-k0) / F(n-1);
90. k = x0;
91. [t,y]=ode45('vdp3', ts, y0, tol);
92. e0 = yd - ym*(k-a0);
93. e1 = e0 +k* y(:,6)+k* y(:,5);
94. G0 = std(e1);%Finding the error function of the boundary point x0
95. k = x1;
96. [t,y]=ode45('vdp3', ts, y0, tol);
97. e0 = yd - ym*(k-a0);
98. e1 = e0 +k* y(:,6)+k* y(:,5);
99. G1 = std(e1);%Finding the error function of the boundary point x1
100. end;
101. l = l + 1;
102. end;
103.
104.
105. k = x0;
106. [t,y]=ode45('vdp3', ts, y0, tol);
107. e0 = yd - ym*(k-a0);
108. e1 = e0 +k* y(:,6)+k* y(:,5);
109. G0 = std(e1);%Finding the error function of the boundary point x0
110. k = x1;
111. [t,y]=ode45('vdp3', ts, y0, tol);
112. e0 = yd - ym*(k-a0);
113. e1 = e0 +k* y(:,6)+k* y(:,5);

```

					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	Лист
						76
Изм.	Лист	№ докум.	Подп.	Дат.		

```

114. G1 = std(e1);%Finding the error function of the boundary point x1
115. if G0>G1
116. k = x1;
117. else
118. k = G1;
119. end;
120.
121. [t,y]=ode45('vdp3', ts, y0, tol);
122. Usig = Ausig *sin(Wusig*t); %Input signal
123. Ush = Aush *sin(Wush*t); %Noise
124. yd=a0*y(:,1)+Ush; %Sensor output signal
125. ym=y(:,3)*(a0-k); %Model output signal
126. Um = yd+k*y(:,3)+k*y(:,4);
127. e0 = yd - ym*(k-a0);
128. e1 = e0 +k* y(:,6)+k* y(:,5);
129.
130. plot (t, Usig, 'g')
131. grid
132. title('Input and output signals of sensor, input signal of model')
133. ylabel('Usig, yd, Um')
134. xlabel(' t , c')
135. hold on
136. plot (t, yd, 'b')
137. plot (t, Um, 'r')
138. hold off
139. pause
140.
141. plot (t, Usig-yd, 'g')
142. grid

```

						Лист
					ЮУрГУ – 12.03.01. 2019. 322 ПЗ ВКР	77
Изм.	Лист	№ докум.	Подп.	Дат.		

