

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

Факультет электротехнический

Кафедра автоматики

Направление подготовки 27.03.04 «Управление в технических системах»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой

Голощапов С.С.

2017 г.

Программное обеспечение автоматической калибровки датчиков инерциальной информации

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ–27.03.04.2017.219.00 ПЗ ВКР

Автор проекта

студент группы МиЭт-428

подпись / Садыкова Д.А.
ФИО

2017 г.

Руководитель работы

Д.т.н., профессор кафедры автоматки
должность

подпись / Войнов И.В.
ФИО

2017 г.

Нормоконтроль

Старший преподаватель
должность

подпись / Елисеев В.П.
ФИО

2017г.

АННОТАЦИЯ

Садыкова Д. А. Программное обеспечение автоматической калибровки датчиков инерциальной информации. – Миасс: ЮУрГУ, ЭТФ; 2017г., 108 стр. 55 ил., библиографический список – 13 наименований, 1 приложение.

В данной выпускной квалификационной работе проведена разработка программного обеспечения для автоматизированной калибровки чувствительных элементов – акселерометров, способного принимать, обрабатывать и визуализировать данные с блока чувствительных элементов инерциальной навигационной системы (ИНС). Блок ИНС установлен на работе-манипуляторе UR10 с шестью вращательными степенями свободы, обеспечивающим необходимые для выполнения калибровки угловые эволюции. Получение данных от блока ИНС выполняется по беспроводному каналу BlueTooth.

Основной задачей разработанного программного обеспечения является:

- выполнение автоматизированных калибровок чувствительных элементов-акселерометров по гибко формируемым программам калибровки;
- визуализация полученной информации в табличном и графическом видах в различных разрезах.

В рамках данной ВКР также проведен анализ нескольких методов поиска экстремума одномерных и двумерных функций с возможностью их применения для автоматического позиционирования оси чувствительности акселерометра вдоль прямой действия ускорения свободного падения на предстартовых проверках акселерометров перед запуском подвижных объектов.

					27.03.04.2017.219.28.00 ПЗ			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>				
<i>Разраб.</i>	Садыкова				Программное обеспечение автоматической калибровки датчиков инерциальной информации	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Провер.</i>	Войнов						4	108
<i>Реценз</i>								
<i>Н. Контр.</i>	Елисеев							
<i>Утверд.</i>	Голощапов							
						ЮУрГУ Кафедра «Автоматика»		

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
1 АНАЛИЗ ТЕХНИЧЕСКОГО ЗАДАНИЯ	9
2 ДАТЧИКИ ИНЕРЦИАЛЬНОЙ ИНФОРМАЦИИ	10
2.1 Акселерометры	11
2.2 Датчики угловой скорости (ДУС)	16
3 ПОСТРОЕНИЕ НАВИГАЦИОННЫХ СИСТЕМ НА БАЗЕ ДАТЧИКОВ ИНЕРЦИАЛЬНОЙ ИНФОРМАЦИИ	17
3.1 Платформенные навигационные системы	20
3.2 Бесплатформенные навигационные системы (БИНС)	22
4 ТИПЫ ПОГРЕШНОСТЕЙ И ВИДЫ КАЛИБРОВКИ ЧУВСТВИТЕЛЬНЫХ ЭЛЕМЕНТОВ	25
5 ПРОЦЕДУРЫ ПРЕДСТАРТОВОЙ ПРОВЕРКИ ПЛАТФОРМЕННОЙ НАВИГАЦИОННОЙ СИСТЕМЫ, УСТАНОВЛЕННОЙ НА ПОДВИЖНОМ ОБЪЕКТЕ	33
6 МАТЕМАТИЧЕСКИЕ СПОСОБЫ (МЕТОДЫ), ИСПОЛЬЗУЕМЫЕ НА ЭТАПАХ ПРЕДСТАРТОВОЙ ПРОВЕРКИ	36
6.1 Метод золотого сечения	36
6.2 Метод квадратичной интерполяции	36
6.3 Квазиньютоновские методы	37
6.3.1 Квази-ньютоновский метод BFGS (Бройдена – Флетчера – Голдфарба – Шэнно)	37
6.3.2 Метод наискорейшего спуска	39
6.3.3 Метод DFP (Дэвидона – Флетчера – Пауэлла)	40
6.4 Метод Нелдера — Мида	41
7 СТЕНДЫ ДЛЯ КАЛИБРОВКИ И ИСПЫТАНИЙ ЧУВСТВИТЕЛЬНЫХ ЭЛЕМЕНТОВ	42
8 ИСПОЛЬЗОВАНИЕ МНОГОЗВЕННЫХ МАНИПУЛЯТОРОВ В КАЧЕСТВЕ СТЕНДОВ КАЛИБРОВКИ	44
9 БЛОК ИНЕРЦИАЛЬНОЙ НАВИГАЦИОННОЙ СИСТЕМЫ (БЛОК ЧУВСТВИТЕЛЬНЫХ ЭЛЕМЕНТОВ)	57

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		5

10 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ АВТОМАТИЗИРОВАННОЙ КАЛИБРОВКИ ЧУВСТВИТЕЛЬНЫХ ЭЛЕМЕНТОВ - АКСЕЛЕРОМЕТРОВ	51
10.1 Реализация алгоритмов калибровки чувствительных элементов	64
10.2 Реализация методов поиска экстремума	78
11 ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ.....	85
ЗАКЛЮЧЕНИЕ	92
ЛИТЕРАТУРА.....	92
ПРИЛОЖЕНИЕ А.....	93

ВВЕДЕНИЕ

Создание систем ориентации и навигации нового поколения определяют постоянный рост требований к точности навигационных комплексов. Одним из основных способов повышения точностных характеристик инерциальных навигационных систем является создание и использование адекватной математической модели датчиков инерциальной информации (гироскопов и акселерометров). Под калибровкой понимается процесс определения или уточнения параметров принятой математической модели датчиков. Для проведения испытаний и калибровки необходимо специализированное оборудование и программное обеспечение, которое позволит экспериментально получить требуемые исходные данные для получения числовых параметров коэффициентов математической модели.

Уточнение параметров математической модели, а также уточнение самой математической модели дает возможность повышения точности навигационных систем алгоритмическими способами.

Основными инструментальными факторами, приводящими к ошибкам в инерциальных навигационных системах, являются погрешности акселерометров и датчиков угловых скоростей, обычно представляемые постоянной составляющей, ошибкой масштабных коэффициентов, ошибками неортогональности и перекоса осей блоков чувствительных элементов, температурной нестабильностью и другими причинами.

Современные навигационные системы – сложные аппаратно-программные комплексы с реализацией алгоритмов получения навигационной информации в бортовом вычислительном устройстве.

При этом вычисление навигационной информации можно разделить на 2 типа:

- вычисление навигационной информации с идеальными датчиками – идеальная математическая модель работы навигационной системы (рис. 1);

- вычисление навигационной информации с учетом модели погрешности датчиков (рис. 2).

Рисунок 1 – Блок-схема вычисления навигационной информации
в случае идеальных датчиков

Рисунок 2 - Блок-схема вычисления навигационной информации
с учетом моделей погрешностей датчиков

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		8

1 АНАЛИЗ ТЕХНИЧЕСКОГО ЗАДАНИЯ

Исходя из требований технического задания, необходимо разработать программное обеспечение для автоматизированной калибровки чувствительных элементов – акселерометров, входящих в состав блока ИНС, установленного на шестизвенном роботе-манипуляторе.

Блок ИНС включает в себя комбинированный микромеханический чувствительный элемент ADIS16485, содержащий 3 датчика угловой скорости, 3 датчика линейных ускорений, оси чувствительности которых ориентированы вдоль осей XYZ правой системы координат. Прием измеренной информации выполняется бортовым процессором блока ИНС и, далее, по беспроводному интерфейсу BlueTooth передается на приемную часть системы (персональный компьютер) для обработки данных и их использования в алгоритмах вычисления навигационной информации.

Робот-манипулятор UR10 является шестизвенным высокоманевренным манипулятором, способным работать в нескольких функциональных режимах. В его состав входит электронный интеллектуальный блок управления с несколькими коммуникационными интерфейсами для управления и сбора данных о состоянии манипулятора. В задачах калибровки чувствительных элементов данный робот-манипулятор рассматривается как задатчик тестовых ориентаций и эволюций.

Программное обеспечение должно выполнять сбор, обработку, визуализацию поступающей от блока ИНС данных, управление и прием статусной информации от манипулятора UR10, реализацию алгоритмов калибровки, реализацию алгоритмов поиска экстремума выходного сигнала акселерометра.

Визуализация данных должна выполняться в режиме реального времени в табличном и графическом виде.

Структурная схема разрабатываемого комплекса управления и сбора и обработки и калибровки представлена на рис.3.

									Лист
									9
Изм.	Лист	№ документа	Подпись	Дата	27.03.04.2017.219.28.00 ПЗ				

Рисунок 3 - Структурная схема работы комплекса испытаний ИНС

Структурная схема системы включает в себя электромеханическую часть – манипулятор (поз. 1 рис. 3), блок инерциальных навигационных датчиков (поз. 2 рис. 3), блок сбора данных (поз. 4 рис. 3) , блок управления (поз. 3 рис. 3) и персональный компьютер с разрабатываемым программным обеспечением. В качестве электромеханического блока используется шестизвенный манипулятор UR10. Данные от блока инерциальных навигационных датчиков поступают по беспроводному каналу в блок сбора данных, далее по проводному интерфейсу USB данные передаются на ПК, на котором установлена программа сбора, калибровки и обработки данных.

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		10

2 ДАТЧИКИ ИНЕРЦИАЛЬНОЙ ИНФОРМАЦИИ

Основными приборами систем инерциальной навигации являются акселерометры и датчики угловых скоростей.

Акселерометр – прибор измеряющий проекцию кажущегося ускорения (разности между истинным ускорением объекта и гравитационным ускорением) на собственную ось чувствительности.

Датчик угловых скоростей (ДУС) предназначен для измерения проекции угловой скорости объекта на ось чувствительности. При этом выходным сигналом ДУС в зависимости от типа может быть электрическим напряжением или ток.

Ускорение является векторной величиной, которая имеет не только численное значение, но и направление. Следовательно, система датчиков, определяющая ускорение, должна измерять и его величину, и его направление.

2.1 Акселерометры

Акселерометр может применяться как для измерения проекций абсолютного линейного ускорения, так и для косвенных измерений проекции гравитационного ускорения. Последнее свойство используется для создания инклинометров. Акселерометры входят в состав инерциальных навигационных систем, где полученные с их помощью измерения интегрируют, получая инерциальную скорость и координаты объекта в стартовой системе координат.

Зная ускорение тела в каждый момент времени, можно рассчитать траекторию его движения. При движении объекта в трехмерном пространстве для вычисления координат необходимо иметь три взаимно ортогональные проекции полного ускорения. Математическая модель, описывающая получение координат, даже в случае идеальных датчиков, достаточно сложна, т.к. необходимо учитывать и повороты объекта в пространстве, что приводит к переориентации осей чувствительности акселерометров.

									Лист
									11
Изм.	Лист	№ документа	Подпись	Дата	27.03.04.2017.219.28.00 ПЗ				

В случае неидеальных датчиков необходимо учитывать как углы ориентации подвижного объекта (углы ориентации осей чувствительности акселерометров) так и собственные погрешности чувствительных элементов (например: дрейф нулевого смещения из-за повышение температуры и другие причины).

Акселерометры применяются не только в летательных аппаратах, но и во многих других объектах. Малогабаритные акселерометры часто встраиваются в мобильные устройства (в частности, в телефоны) и применяются в качестве шагомеров, датчиков для определения положения в пространстве, автоматического поворота дисплея и других целей. В устройствах управления игровых приставок акселерометр совместно с гироскопом используются для управления в играх без использования кнопок - путем поворотов в пространстве, встряхиваний и т. д. Акселерометры используют в жестких дисках для активации механизма защиты от повреждений, полученных в результате ударов, встрясок и падений. Акселерометр реагирует на внезапное изменение положения устройства и паркует головки жесткого диска, что позволяет предотвратить повреждение диска и потерю данных. Такая технология защиты используется в основном в ноутбуках, нетбуках и на внешних накопителях.

Все акселерометры используются для измерения проекции, связанной с кажущимся ускорением. С его помощью определяется разница абсолютного ускорения с ускорением свободного падения того или иного объекта.

Акселерометры классифицируются по следующим характеристиками и критериям:

- физический принцип работы (маятниковые акселерометры, микромеханические акселерометры, струнные акселерометры);
- диапазон измеряемых линейных ускорений;
- количество осей чувствительности;
- тип выходного сигнала (аналоговое напряжение или ток, цифровой интерфейс);

										Лист
										12
Изм.	Лист	№ документа	Подпись	Дата	27.03.04.2017.219.28.00 ПЗ					

В пьезоэлектрических акселерометрах применяются материалы, формирующие электрический потенциал при приложении давления или микроперемещениях. Это явление известно как пьезоэлектрический эффект.

Пьезоэлектрические акселерометры обладают следующими преимуществами:

- широким рабочим диапазоном частот;
- линейностью амплитудной характеристики (АХ) в широком динамическом диапазоне;
- высокой стойкостью к внешним воздействиям;
- активным характером преобразования, не требующим применения источника питания;
- отсутствием движущихся частей, гарантирующим высокую надежность и долговечность;
- возможностью реализации малогабаритного исполнения;
- технологичностью в производстве.

Метрологические и эксплуатационные характеристики акселерометров, прежде всего, зависят от свойств используемого пьезоматериала (ПМ) и конструктивных особенностей электромеханического преобразователя, а также датчика в целом.

Пьезоэлектрические акселерометры, кроме электромеханического преобразователя, могут содержать встроенные электронные устройства, например, согласующий усилитель (СУ), схему первичной обработки сигнала и др.[7]

Пьезорезистивный акселерометр (тензометрический акселерометр) изменяет электрическое сопротивление материала при приложенном механическом давлении. Датчики деформации пьезорезистивных акселерометров изменяют электрическое сопротивление пропорционально приложенному механическому напряжению. Монолитный датчик акселерометра включает в себя встроенные механические ограничители и обладает очень высокой прочностью при очень хорошем соотношении сигнал/шум. Акселерометры этого типа идеально

подходят для измерения перемещения, низкочастотной вибрации и ударного воздействия и предназначены для испытаний на столкновение с препятствием, на езду по трудным дорогам, а также для биодинамических измерений и тому подобных приложений. [8]

Интегральные акселерометры представляют собой датчики линейного ускорения и в этом качестве широко используются для измерения углов наклона тел, сил инерции, ударных нагрузок и вибрации. Они находят широкое применение на транспорте, в медицине, в промышленных системах измерения и управления, в инерциальных системах навигации.

Пленочные пьезоэлектрические акселерометры выполняются на основе многослойной пьезоэлектрической полимерной пленки. Многослойная пленка закреплена на подложке из окиси алюминия, и к ней присоединена инерционная масса из порошкового металла. При изменении скорости движения датчика в результате действия инерционных сил происходит деформация пленки. Благодаря пьезоэффекту возникает разность потенциалов на границах слоев пленки, зависящая от ускорения.

Это позволяет измерять переменные ускорения со сравнительно низкой частотой. Датчики этого типа имеют плохую повторяемость характеристик в серийном производстве, высокую чувствительность к изменению температуры и давления. Они не могут контролировать постоянные ускорения и гравитационные силы. Основная область применения - схемы управления надувными подушками безопасности.

Другие области практического применения акселерометров:

- измерение динамических характеристик автомобиля при движении автомобиля;
- измерение вибраций машин;
- контроль передвижений разводных мостов;
- контроль укладки багажа.

При выборе акселерометров руководствуются следующими основными характеристиками:

Динамический диапазон акселерометра – максимальная амплитуда, которую акселерометр может измерить без искажения или потери выходного сигнала.

Динамический диапазон обычно указывается в единицах системы СИ (м/с^2) либо в относительных единицах – отношении максимально измеряемого ускорения к величине ускорения свободного падения g .

Чувствительность акселерометра – масштабный коэффициент датчика или системы, выраженный в единицах измерения выходного сигнала относительно изменения входного сигнала.

Чувствительность определяет способность акселерометра фиксировать движение. Чувствительность акселерометра обычно указана в милливольтметрах за g ($\text{мВ/}g$).

Частотный диапазон акселерометра – диапазон частот, в котором датчик обнаруживает движение и выдает действительный выходной сигнал. Частотный отклик обычно указан в герцах (Гц).

2.2 Датчики угловой скорости (ДУС)

ДУС — устройство, первичный прибор (датчик) для измерения угловой скорости поворота подвижного объекта относительно неподвижной стартовой инерциальной системы координат. ДУС используется в системах управления различных летательных аппаратов: ракет, самолетов, вертолетов и др. Выходной сигнал устройства обычно электрический, пропорциональный угловой скорости и используется в пилотажных системах летательных аппаратов, в частности, автопилоте, системах стабилизации траектории полёта ракет. Обработка сигнала производится во внешней системе.

Для измерения угловых скоростей по взаимно ортогональным координатным осям почти всегда применяют три взаимно ортогонально ориентированных датчика по каналам крена, тангажа и рысканья.

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		16

Принципиально возможно вычисление угловых скоростей по осям путем дифференцирования по времени углов поворота механического гироскопа в кардановом подвесе, но такой метод не даёт достаточной чувствительности и точности.

Поэтому широко применяются ДУС с поплавковыми (погружёнными в вязкую жидкость в герметичном кожухе) гироскопами. В таком гироскопе жидкость выполняет роль вязкой демпфирующей среды. При повороте корпуса датчика за счет сил вязкого трения между корпусом и кожухом ротора гироскопа создается демпфирующий момент. В результате гироскопического эффекта происходит прецессия гироскопа, измеряемая внутренними вспомогательными датчиками поворота. Углы поворота преобразуются в электрический информационный сигнал, выдаваемый во внешние электрические цепи.

Для правильной ориентации при установке на летательный аппарат на корпусе ДУС обязательно указывают направление оси чувствительности. ДУС, как правило, устанавливают вблизи центра тяжести летательного аппарата.

ДУСы помимо основного применения в системах инерциальной навигации находят свое применение в системах курсовой устойчивости автотранспортных средств, стабилизации изображений в цифровых кинокамерах и фотоаппаратах, механических игрушках и др. В этих применениях обычно используются микромеханические гироскопы.[10]

3 ПОСТРОЕНИЕ НАВИГАЦИОННЫХ СИСТЕМ НА БАЗЕ ДАТЧИКОВ ИНЕРЦИАЛЬНОЙ ИНФОРМАЦИИ

Навигация — это процесс определения положения подвижного объекта в пространстве относительно других объектов. Задача навигации решается методами и приборами мореходной, воздушной и космической навигации, которые позволяют определить местоположение и ориентацию движущегося объекта относительно принятой системы координат, величину и направление скорости движения, направление и расстояние до места назначения и т.д. [2]

Общей задачей навигации является обеспечение перемещения летательного аппарата в заданную точку по заданной траектории за определенное время и с необходимой точностью. Решение этой задачи осуществляется с помощью пилотажно-навигационных комплексов (ПНК).

Частной задачей навигации является вычисление текущих координат местоположения подвижного объекта. Эта задача решается навигационными устройствами и системами, установленными как на борту подвижного объекта, так и на Земле. [1]

Из всех навигационных систем инерциальные навигационные системы (ИНС) являются единственными, которые удовлетворяют таким важным требованиям, как универсальность, полная автономность, помехозащищенность и помехоустойчивость. Помимо вычисления линейных координат, ИНС определяет (вычисляет) также угловое положение летательного аппарата, то есть курс, крен, тангаж. Угловое положение определяется путем измерения отклонений летательного аппарата от запомненной в ИНС выбранной системы координат. В зависимости от способа моделирования (запоминания), ИНС подразделяются на платформенные и бесплатформенные. В платформенных ИНС навигационная система координат моделируется физически – в виде ориентации по трем осям специальной гиросtabilизированной платформы (ГСП). ГСП в течение всего рабочего времени сохраняет неизменной ориентацию, заданную в процессе подготовки (выставки в начальном положении) ИНС.

Датчики инерциальной информации - акселерометры устанавливаются на ГСП. В бесплатформенных ИНС акселерометры устанавливаются непосредственно на ЛА, а опорная система координат моделируется математически. Так как акселерометры вращаются вместе с ЛА, то необходим непрерывный пересчет значений ускорений из связанной с ЛА вращающейся системы координат в опорную неподвижную систему координат. Алгоритмы пересчета требуют, помимо значений ускорений, дополнительную информацию – значения угловых скоростей вращения ЛА. Значения угловых скоростей вращения летательного аппарата определяются с помощью одноосных или многоосных датчиков угловой скорости. Сложность алгоритмов и большой объем вычислений требуют наличия мощной, быстродействующей и имеющей большой объем памяти бортовой цифровой вычислительной машины. (БЦВМ). Бесплатформенные ИНС в настоящее время используются преимущественно на тяжелых маломаневренных ЛА.

Среди которых платформенных ИНС следует выделить наиболее простой их тип – так называемые инерциальные курсовертикали (ИКВ).

В ИКВ-системе применяются упрощенные алгоритмы вычисления и управления и более простые датчики и элементы, вследствие чего приемлемая точность достигается лишь в определении курса, крена и тангажа. Принцип построения элементарной ИНС, определяющий линейные и географические координаты, рассмотрен на рис. 4.

Рисунок 4 - Принцип построения ИНС

Из рисунка видно, что гиростабилизированная платформа (ГСП) имеет 3 степени свободы. В узлах подвеса ГСП размещены датчики курса, крена, тангажа (на рис.4 изображен лишь один из них). Линейные координаты ЛА формируются путем двойного интегрирования сигналов акселерометров, а географические – интегрированием сигнала угловой скорости движения ЛА вокруг Земли. Сигнал

углового смещения ЛА появляется на выходе датчика угла при любом вращении ЛА вокруг неподвижной оси подвеса ГСП.

Собственно, сущность инерциального счисления заключается в измерении с помощью акселерометров составляющих вектора ускорения центра масс ЛА, обусловленного негравитационными силами и в последующем двукратном интегрировании этих составляющих с учетом начальных условий и ускорений, вызванных гравитацией, силами инерции Кориолиса и центробежными силами.

[3]

3.1 Платформенные навигационные системы

В платформенных ИНС взаимосвязь блока измерителей ускорений и гироскопических устройств, обеспечивающих ориентацию акселерометров в пространстве, определяет тип инерциальной системы. Известны три основных типа платформенных инерциальных систем.

Инерциальная система геометрического типа имеет две платформы. Одна платформа с гироскопами ориентирована и стабилизирована в инерциальном пространстве, а вторая с акселерометрами — относительно плоскости горизонта. Координаты объекта определяются в вычислителе с использованием данных о взаимном расположении платформ. Обладает высокой точностью позиционирования относительно поверхности планеты, но неудовлетворительно работает на высокоманевренных аппаратах и в космическом пространстве. Применяется, в основном, на самолётах с большой дальностью полёта (гражданские, военно-транспортные, стратегические бомбардировщики), подводных лодках и крупных надводных кораблях.

В инерциальных системах аналитического типа и акселерометры, и гироскопы неподвижны в инерциальном пространстве. Координаты объекта получаются в вычислителе, обрабатывающем сигналы, поступающие с акселерометров и устройств-определителей поворота самого объекта относительно гироскопов и акселерометров. Обладает сравнительно невысокой точностью при движении у

										Лист
										20
Изм.	Лист	№ документа	Подпись	Дата	27.03.04.2017.219.28.00 ПЗ					

поверхности Земли, но хорошо работает на маневренных объектах (истребители, вертолеты, ракеты, скоростные маневренные надводные суда) и в космическом пространстве.

Полуаналитическая система имеет платформу, которая непрерывно стабилизируется по местному горизонту. На платформе имеются гироскопы и акселерометры. Координаты самолёта или иного летательного аппарата определяются в вычислителе, расположенном вне платформы.

В основе построения инерциальных навигационных систем (ИНС) лежит метод счисления пути. Суть его состоит в том, что сигналы измерителей скорости или ускорения, оси чувствительности на которых удерживаются в заданной системе координат и интегрируются. Интегралы скорости соответствуют приращениям пути, интегралы ускорений соответствуют приращениям скорости. Складывая значения приращений с начальными значениями пути или скорости, получают текущие значения пройденного пути и скорости. В системах, где первичными измерителями являются акселерометры, для получения скорости сигнал акселерометра интегрируют один раз, а для получения пройденного пути этот интеграл (скорость) интегрируют второй раз. Зная направления проекций пройденного пути на оси системы координат, определяют координаты подвижного объекта.

Устройство ИНС, реализующей метод счисления пути, можно проиллюстрировать обобщенной схемой, приведенной на рис.5 [4]

Рисунок 5 - Обобщенная схема платформенной ИНС

На рисунке 5 показаны:

1 (А) – блок акселерометров (трехосный акселерометр);

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		21

- 2 (ГСП) – гиросtabilизированная платформа, удерживающая оси чувствительности акселерометров в заданной системе координат;
- 3,6,8 – сумматоры;
- 4,7 – интеграторы;
- 5 – вычислитель вектора гравитационного ускорения;
- 9 – обратные связи;

Введены следующие обозначения:

- векторы кажущегося и гравитационного ускорений соответственно;
 - векторы абсолютного ускорения и абсолютной скорости;
 - приращения абсолютной скорости и радиус-вектора местоположения объекта;
 - радиус-вектор местоположения объекта;
 - начальные значения векторов;
- ψ, ϑ, γ – углы ориентации объекта (курс, тангаж, крен).

Если в системе имеются обратная связь 9 по скорости или по координатам, систему называют замкнутой, если обратной связи нет – система разомкнутая.

3.2 Бесплатформенные навигационные системы (БИНС)

В БИНС вместо гиросtabilизированной платформы используют блок гироскопов и акселерометров с вычислителем. Гироскопы с акселерометрами (блок инерциальных чувствительных элементов) при этом устанавливают жестко на борту объекта. [4]

Рисунок 6 - Обобщенная схема БИНС

В целом к достоинствам ИНС вышеуказанных типов можно отнести:

1. Возможность измерения и формирования нескольких типов выходной информации, например: ускорение, координаты, угловое положение, т. е. курс, тангаж, крен и ряда других вспомогательных;
2. Высокая степень автономности при обеспечении необходимой точности;
3. Высокая скорость определения и выдачи данных (100 Гц и более).

К недостаткам ИНС можно отнести:

1. Необходимость ввода начальных условий (начального положения платформы (объекта для БИНС), начальных значений скорости, начальных координат), что необходимо для использования метода счисления пути; необходимость учета формы Земли и параметров гравитационного поля в точке расположения подвижного объекта. [11]

2. Необходимость непрерывной работы после запуска, иначе потребуется заново вводить начальные условия для работы.

3. Большим недостатком системы инерциальной навигации является то, что ошибки выходной навигационной информации со временем накапливаются. Это обусловлено интегрирующими составляющими алгоритма работы системы. Если накопленная ошибка становится слишком большой, ее можно корректировать с помощью внешних вспомогательных средств. В конечном результате система становится неавтономной.

4 ТИПЫ ПОГРЕШНОСТЕЙ И ВИДЫ КАЛИБРОВКИ ЧУВСТВИТЕЛЬНЫХ ЭЛЕМЕНТОВ

4.1. Типы погрешностей чувствительных элементов

Указанные в данном разделе зависимости и графики приведены на примере чувствительного элемента – акселерометра, но, в общем виде, применимы для чувствительного элемента любого типа и назначения.

Идеальным чувствительным элементом считается чувствительный элемент, обладающий следующей статической характеристикой (зависимостью выходного сигнала чувствительного элемента – акселерометра (ЧЭ) от измеряемой физической величины):

вых а пр

Рисунок 7 - Идеальная характеристика акселерометра

Любой из существующих на сегодняшний день ЧЭ является неидеальным и имеет следующие основные типы погрешностей:

1. Нулевое смещение выходного сигнала чувствительного элемента.

вых пр

Рисунок 8 - Характеристика чувствительного элемента с нулевым смещением
в выходном сигнале

2. Нелинейность масштабного коэффициента.

вых пр

Рисунок 9 – Нелинейная статическая характеристика ЧЭ

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		25

3. Наличие зон нечувствительности.

Рисунок 10 – Статическая характеристика ЧЭ
с наличием зон нечувствительности

4. Ограничения по диапазону измеряемой физической величины.

Рисунок 11 – Статическая характеристика ЧЭ с наличием зон насыщения

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		26

Как правило в реальных ЧЭ присутствуют несколько видов погрешностей статической характеристики. На рис. 11 приведена статическая характеристика ЧЭ, являющаяся одновременно нелинейной, имеющей ограничения по выходному напряжению, имеющая зону нечувствительности и нулевое смещение.

Рисунок 11 – Статическая характеристика ЧЭ с несколькими видами погрешностей ЧЭ

5. Наличие случайной составляющей в выходном сигнале чувствительного элемента (так называемого «шума»).

Рисунок 12 – Пример случайной составляющей в выходном сигнале ЧЭ

6. Нестационарность параметров ЧЭ (дрейфы): температурный, временной дрейф, изменение параметров ЧЭ при долговременном хранении.

Рисунок 13 – Пример дрейфа сигнала ЧЭ с течением времени при фиксированном по величине входном физическом воздействии

Наличие погрешностей любого из вышеприведенных типов вынуждает производителей ЧЭ формировать математическую модель погрешностей ЧЭ с целью дальнейшей компенсации погрешностей при работе ЧЭ в составе навигационной системы. В случае невозможности создания адекватной (полной) математической модели ЧЭ, используют принцип «паспортизации» ЧЭ, храня таблицу соответствия реально прикладываемой к ЧЭ физической величины и значения выходного сигнала ЧЭ, при данной физической величине.

И в том, и в другом случае бортовая навигационная система на основе имеющейся математической модели или таблицы калибровки пытается «привести» сигнал ЧЭ к сигналу идеального ЧЭ. При этом, чем точнее

сформирована математическая модель, тем лучше выполняется приближение к идеальному ЧЭ.

4.2. Методы калибровки чувствительных элементов.

Для формирования математической модели ЧЭ требуется наличие некоторого набора исходных экспериментальных данных, полученных от ЧЭ при различных видах испытаний, зависящих от паспортизируемого параметра математической модели. Для акселерометров типовыми испытаниями являются:

1. Получение статических характеристик ЧЭ при нормальных (лабораторных) условиях измерения. При этом возможно разбиение диапазона измерения ЧЭ на отдельные участки с целью детализации получаемых результатов.
2. Получение статистической информации о случайной составляющей в выходном сигнале ЧЭ.
3. Получение динамических характеристик ЧЭ – амплитудно-частотной характеристики, фазо-частотной характеристики, переходной характеристики.
4. Получение экспериментальных зависимостей дрейфа параметров ЧЭ от температуры и других внешних факторов.

Наличие максимально полного набора исходной информации позволяет разработчикам ЧЭ сформировать как математическую модель погрешностей ЧЭ, так и провести анализ самого ЧЭ в части оптимизации конструкции для последующего уменьшения величины погрешностей.

При получении экспериментальных данных о статических характеристиках ЧЭ (см п. 1 выше) возможно использование упрощенных методов, например, построение такой характеристики по двум точкам, однако очевидно, что чем

									Лист
									28
Изм.	Лист	№ документа	Подпись	Дата	27.03.04.2017.219.28.00 ПЗ				

таких точек измерения больше, тем более детально мы можем сформировать статическую характеристику. Следующим этапом является формирование интерполированной статической характеристики, в простейшем случае это может быть кусочно-линейная интерполяция, при наличии технической возможности - полиномиальная или сплайн-интерполяция.

В рамках данной ВКР разработано программное обеспечение, позволяющее в автоматизированном режиме получать исходные данные для построения статической характеристики акселерометра.

Наличие случайной («шумовой») составляющей в выходном сигнале ЧЭ требует применения методов математической статистики до начала формирования статической характеристики.

Математическая статистика — наука, разрабатывающая математические методы систематизации и использования статистических данных для научных и практических выводов.

Во многих своих разделах математическая статистика опирается на теорию вероятностей, дающую возможность оценить надёжность и точность выводов, делаемых на основании ограниченного статистического материала (напр., оценить необходимый объём выборки для получения результатов требуемой точности при выборочном обследовании). [13]

Базовыми числовыми характеристиками случайного сигнала являются: математическое ожидание, среднеквадратическое отклонение, и, в целом, закон распределения случайной величины. В ряде случаев, при априорно известных характерных распределениях случайной величины (например, распределение Гаусса) можно ограничиться вычислением математического ожидания и среднеквадратического отклонения. [4]

Математическим ожиданием дискретной случайной величины называется сумма произведений всех возможных значений случайной величины на их вероятности.

(4.1)

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		29

Математическое ожидание приближенно равно среднему арифметическому наблюдаемых значений случайной величины.

Математическое ожидание не может полностью характеризовать случайный процесс. Кроме математического ожидания надо ввести величину, которая характеризует отклонение значений случайной величины от математического ожидания. Это отклонение равно разности между случайной величиной и ее математическим ожиданием.

Дисперсией (рассеиванием) дискретной случайной величины называется математическое ожидание квадрата отклонения случайной величины от ее математического ожидания.

(4.2)

Дисперсия вычисляется по формуле:

(4.3)

Среднеквадратическим отклонением случайной величины X называется квадратный корень из дисперсии.

X — (4.4)

Если при проведении с одинаковой тщательностью и в одинаковых условиях ряда наблюдений одной и той же физической величины получены отличающиеся друг от друга результаты, то это свидетельствует о наличии в них случайных погрешностей.

Поступающие в режиме реального времени от блока ИНС данные с помощью автономного ПО подвергаются предварительной математической обработке в части получения характеристик случайной величины. Далее значения математического ожидания используются в алгоритмах получения статической характеристики.

Стоит заметить, что проведение испытаний ЧЭ является метрологической задачей, что само по себе требует наличие точных измерительных инструментов,

оборудования. При этом, состав оборудования и требования к его точности могут варьироваться от вида испытаний.

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		31

5 ПРОЦЕДУРЫ ПРЕДСТАРТОВОЙ ПРОВЕРКИ ПЛАТФОРМЕННОЙ НАВИГАЦИОННОЙ СИСТЕМЫ, УСТАНОВЛЕННОЙ НА ПОДВИЖНОМ ОБЪЕКТЕ

Сложность технических систем современных подвижных объектов предполагает и требует проведения контроль их работоспособности на различных этапах изготовления, сборки, монтажа. Одним из важных этапов таких проверок являются предстартовые проверки непосредственно перед запуском подвижного объекта. Состав и объем предстартовых проверок утверждается разработчиками изделия и, в том числе, включает в себя проверки бортовых вычислительных систем, чувствительных элементов и исполнительных органов подвижного объекта. Важной составляющей является предстартовая проверка чувствительных элементов бортовой навигационной системы. В случае платформенных навигационных систем задача предстартовых проверок чувствительных элементов и навигационной системы в целом несколько упрощается, т.к. принцип работы и конструкция навигационных систем такого типа позволяет выполнять переориентации внутренних колец платформы и площадки с ЧЭ в некотором диапазоне углов и угловых скоростей.

В случае с чувствительными элементами – акселерометрами, выполнение предстартовых проверок, как правило, связано с переориентацией их осей чувствительности в несколько типовых положений: по направлению действия вектора ускорения свободного падения, против направления действия вектора ускорения свободного падения, и, в ряде случаев, переориентация до совпадения оси чувствительности акселерометра с плоскостью, перпендикулярной вектору ускорения свободного падения.

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		32

В общем виде данная задача относится к классу задач оптимизации многомерных функций. В рамках данной ВКР рассмотрены математические методы и их реализация для одномерных и двумерных функций.

Рисунок 14 – Варианты ориентации подвижного объекта на старте

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		34

6 МАТЕМАТИЧЕСКИЕ СПОСОБЫ (МЕТОДЫ), ИСПОЛЬЗУЕМЫЕ НА ЭТАПАХ ПРЕДСТАРТОВОЙ ПРОВЕРКИ

На сегодняшний день для задач одномерной и многомерной оптимизации и методов поиска экстремума существует ряд математических методов. В рамках данной ВКР рассмотрена и промоделирована часть этих методов. Моделирование выполнялось с использованием программного пакета MATLAB.

MATLAB (MATrix LABoratore) – одна из наиболее мощных систем компьютерной математики, построенная на расширенном применении матричных вычислений. Она включает пакеты расширения Simulink, Toolbox и Blockset, ориентированные на самые разнообразные задачи, где требуются математические расчеты, глубокий анализ и моделирование.

MATLAB имеет развитый высокоуровневый язык программирования и прекрасную двумерную и трехмерную графику, позволяющую наглядно представлять результаты расчетов, экспериментов и процессы моделирования.

Пакеты расширения Toolbox специализируются на решении задач в различных предметных областях: в физике и астрономии, в специальных разделах математики и математическом моделировании, в проектировании и управлении технических систем и т.д. Основным инструментом MATLAB являются функции, которые реализуют определенные операции и алгоритмы. Имеются встроенные функции, в том числе все элементарные, и внешние, оформленные в виде так называемых m-файлов. Внешние функции могут пополняться собственными функциями пользователя.

В расширения Toolbox входят два пакета, предназначенные для решения задач оптимизации: Toolbox Optimization и Toolbox Global Optimization.

Toolbox Optimization включает набор функций, в которых реализованы методы и алгоритмы одномерной и многомерной безусловной и условной минимизации, линейного, булевого и квадратичного программирования, метод наименьших квадратов, а также алгоритмы многокритериальной оптимизации.

Для одномерной минимизации непрерывных функций предназначена функция пакета `fminbnd`. Она решает задачу вида $\min f(x)$, $a < x < b$. Алгоритм минимизации базируется на методах золотого сечения и квадратичной аппроксимации (параболической интерполяции).

Для поиска минимума нелинейной функции многих переменных при отсутствии ограничений можно использовать функции `fminunc` или `fminsearch` из пакета `Toolbox Optimization`.

Из наиболее широко распространенных рассмотрены следующие виды поиска экстремума:

6.1 Метод золотого сечения

Метод основан на делении текущего отрезка, где содержится искомый экстремум, на две неравные части, подчиняющиеся правилу золотого сечения, для определения следующего отрезка, содержащего минимум (максимум).

Золотое сечение определяется по правилу: отношение всего отрезка к большей его части равно отношению большей части отрезка к меньшей.

6.2 Метод квадратичной интерполяции

Метод заключается в замене нелинейной функции $R(x)$ квадратичной параболой $R_2(x)$, построенной по трем точкам, принадлежащим $R(x)$, с последующим нахождением минимума (максимума) параболической функции, используя аналитические условия оптимальности: $dR/dx = 0$.

Пример. Найти минимум функции $y = x + 25/x$ в интервале $2 < x < 9$.

Для решения задачи в командное окно записываем программу:

```
>> x=2:0.1:9;
y=x+25./x;
plot(x,y);
xlabel('x');
ylabel('y');
title('Minimization by fminbnd');
```

```
hold on;
options=optimset('Outputfcn', @outmy);
[x,fval,exitflag,output]=fminbnd('x+25./x',2,9,options)
```

Рисунок 15 – Графическое представление кода программы и результатов поиска локального минимума

6.3 Квазиньютоновские методы

Методы оптимизации, основанные на накоплении информации о кривизне целевой функции по наблюдениям за изменением градиента, чем принципиально отличаются от ньютоновских методов. Класс квазиньютоновских методов исключает явное формирование матрицы Гессе, заменяя её некоторым приближением.

6.3.1 Квази-ньютоновский метод BFGS

(Бройдена – Флетчера – Голдфарба – Шэнно)

Итерационный метод численной оптимизации, предназначенный для нахождения локального максимума/минимума нелинейного функционала без ограничений.

BFGS — один из наиболее широко применяемых квазиньютоновских методов. В квазиньютоновских методах не вычисляется напрямую гессиан функции. Вместо этого гессиан оценивается приближенно, исходя из сделанных до этого шагов. Также существуют модификация данного метода с ограниченным использованием памяти, который предназначен для решения нелинейных задач с большим количеством неизвестных, а также модификация с ограниченным использованием памяти в многомерном кубе.

Данный метод находит минимум любой дважды непрерывно дифференцируемой выпуклой функции. Несмотря на эти теоретические ограничения, как показывает опыт, метод BFGS хорошо справляется и с невыпуклыми функциями. [11]

Пример 2.

Найти минимум функции Розенброка

$$f(X)=100(x_2 - x_1^2)^2 + (x_1 - 1)^2.$$

```
function history = runfminunc(x0)
% Set up shared variables with OUTFUN
history.x = [];
history.fval = [];
% call optimization
options = optimset('largescale','off','outputfcn',@outfun,'display','on');
[x,fval,exitflag,output]=fminunc(@rozen,x0,options);
disp('Xopt=');disp(x);disp('fopt=');disp(fval);
disp(output);
[X,Y]=meshgrid(-4:0.05:4);Z=rozen_g(X,Y);[c,h]=contour...
(X,Y,Z,[5 12 70 120 200 300 450 600 800],'blue');
clabel(c,h,[70 300 600],'FontSize',7);
y=history.x(:,1);z=history.x(:,2);
hold on;plot(y,z,'black.','y,z','red-');
xlabel('x1');ylabel('x2');
text(x0(1)-0.1,x0(2)+0.2,'x0');
title('Minimization by fminunc for Rozenbrok');
function stop = outfun(x,optimValues,state)
stop = false;
switch state
case 'init'
```

```

hold off
case 'iter'
% Concatenate current point and objective
% function value with history. x must be a row
% vector. history.fval=[history.fval;
% optimValues.fval];
history.x = [history.x; x];
end
end
end

```

В этой программе функция Розенброка представлена в двух вариантах: как входной аргумент `fminunc` и для визуализации ее контуров (линий уровня). Им соответствуют m-файлы `rozen` и `rozen_g`, приводимые ниже.

```

function f = rozen(x)
f=100*(x(2)-x(1)^2)^2+(x(1)-1)^2;
end

```

```

function f = rozen_g(X,Y)
f=100*(Y-X.^2).^2+(X-1).^2;
end

```

Рисунок 16 - Графическое представление кода программы и результатов поиска локального минимума по методу BFGS

6.3.2 Метод наискорейшего спуска

При использовании метода наискорейшего спуска на каждой итерации величина шага a_k выбирается из условия минимума функции $f(x)$ в направлении спуска.

Это условие означает, что движение вдоль антиградиента происходит до тех пор, пока значение функции $f(x)$ убывает.

В этом методе воспользуемся примером 2 (см. выше), добавим в строку options: 'HessUpdate', 'steepdesc').

В общем виде эта функция будет выглядеть так:

```
options = optimset('largescale','off','outputfcn',@outfun,'display','on', 'HessUpdate',  
'steepdesc');
```

Рисунок 17 – Графическое представление кода программы и результатов поиска локального минимума по методу наискорейшего спуска

6.3.3 Метод DFP (Дэвидона – Флетчера – Пауэлла)

Метод Дэвидона-Флетчера-Пауэлла называют также методом переменной метрики. Он попадает в общий класс квазиньютоновских процедур, в которых направления поиска задаются в виде $-D_j f(y)$. Направление градиента является, таким образом, отклоненным в результате умножения на $-D_j$, где D_j - положительно определенная симметрическая матрица порядка n , аппроксимирующая обратную матрицу Гессе. На следующем шаге матрица D_{j+1} представляется в виде суммы D_j и двух симметрических матриц ранга один каждая. В связи с этим схема иногда называется схемой коррекции ранга два.[12]

В этом методе воспользуемся примером 2 (смотреть выше), добавим в строку options: 'HessUpdate', 'DFP')

В общем виде эта функция будет выглядеть так:

```
options = optimset('largescale','off','outputfcn',@outfun,'display','on', 'HessUpdate',  
'DFP');
```

Рисунок 18 – Графическое представление кода программы и результатов поиска локального минимума по методу DFP

6.4 Метод Нелдера — Мида

Также известный как метод деформируемого многогранника и симплекс-метод - метод безусловной оптимизации функции от нескольких переменных, не использующий производной (точнее градиентов) функции, а поэтому легко применим к негладким или зашумлённым функциям.

Суть метода заключается в последовательном перемещении и деформировании симплекса вокруг точки экстремума.

Метод находит локальный экстремум и может «застрять» в одном из них. Если всё же требуется найти глобальный экстремум, можно пробовать выбирать другой начальный симплекс. Более развитый подход к исключению локальных экстремумов предлагается в алгоритмах, основанных на методе Монте-Карло, а также в эволюционных алгоритмах.

Пример 3.

Найти минимум функции Розенброка

$$f(x) = 100(x_2 - x_1^2)^2 + (x_1 - 1)$$

Воспользуемся программой примера 2, но в ней скорректируем несколько строк в функции `function history = runfminsearch(x0)`, описание которой совпадает с приведенным для функции `runfminunc`, в нем заменяется только `fminunc` на `fminsearch`.

Рисунок 19 – Графическое представление кода программы и результатов поиска локального минимума по методу Нелдера — Мида

Каждый из опробованных методов обладает как достоинствами, так и недостатками и наиболее оптимален для функций конкретных классов, обеспечивая наименьшее число итераций (шагов) при поиске экстремума. В данной ВКР практически реализованы методы одномерной оптимизации: метод деления отрезка пополам, метод золотого сечения.

7 СТЕНДЫ ДЛЯ КАЛИБРОВКИ И ИСПЫТАНИЙ ЧУВСТВИТЕЛЬНЫХ ЭЛЕМЕНТОВ

Системы инерциальной навигации и ее отдельные компоненты, в том числе чувствительные элементы, в процессе опытного и серийного производства проходят различные этапы испытаний с целью подтверждения заявленных функциональных характеристик и точностных параметров. Сюда относятся испытания на снятие статических и динамических характеристик чувствительных элементов, климатический испытания, испытания на вибро- и ударо- прочность и т.д. Чувствительный элемент может выступать как объект автономных испытаний, либо быть компонентом испытываемой системы более высокого уровня. По этому критерию испытания могут быть разделены на лабораторно-отрабочные испытания, совместные отрабочные испытания (отрабатывается совместное функционирование компонентов системы или нескольких систем), приемо-сдаточные испытания (сдача комплекса технических средств представителю заказчика) и т.д.

При проведении любого типа испытаний, проверок, снятия характеристик требуется различное испытательное оборудование (испытательные стенды) с заведомо известными точностными характеристиками. В части испытаний и проверок чувствительных элементов – акселерометров можно выделить следующие категории стендов:

1. Стенды, организующие (задающие) измеряемую физическую величину – линейное ускорение.
2. Стенды, организующие вибро-ударные испытания.
3. Климатические стенды.
4. Комплексные стенды, обеспечивающие проведение многокритериальных испытаний.

К первой категории можно отнести стенды, ориентирующие чувствительный элемент в пространстве (формируя при этом проекцию ускорения свободного падения на ось чувствительности ЧЭ в диапазоне от [-1g; 1g]), центрифуги различного типа, обеспечивающие величины линейных ускорений больше величины 1g. На рис. 20 приведено изображение стендов подобного класса.

Рисунок 20 – Варианты исполнений стендов для испытаний ЧЭ
путем их переориентации в пространстве

Комплексные испытательные стенды могут включать в свой состав, например, термостабилизирующую камеру, позволяя выполнять испытания при различных температурных воздействиях (рис. 21)

Рисунок 21 – Вариант исполнения комплексного стенда с термокамерой

8 ИСПОЛЬЗОВАНИЕ МНОГОЗВЕННЫХ МАНИПУЛЯТОРОВ В КАЧЕСТВЕ СТЕНДОВ КАЛИБРОВКИ

В случае простых или оценочных испытаний чувствительных элементов возможно использование менее дорогостоящих и более функциональных, в части формирования пространственных конфигураций, многозвенных манипуляторов. Для подобных задач можно применять как роботы-манипуляторы классических схем (многозвенные манипуляторы с линейными или вращательными степенями свободы), так и манипуляторы более сложных конфигураций (например, платформа Стюарта). С точки зрения выполнения калибровок и испытаний функциональные задачи у манипулятора аналогичны классическим «ориентирующим» стендам – формирование произвольных пространственных фиксированных положений с заданной угловой точностью, а также обеспечение разворота чувствительных элементов с определенной угловой скоростью.

В рамках данной ВКР в качестве задатчика тестовых ориентаций задан промышленный электромеханический робот-манипулятор UR10 фирмы Universal Robots, обладающий следующими характеристиками:

Число вращательных степеней свободы	6
Число линейных степеней свободы	0
Диапазон угловых перемещений по каждому звену	$\pm 360^\circ$
Угловая точность позиционирования любого звена	$\pm 0,005^\circ$
Диапазон угловых скоростей по каждому звену	$\pm 120^\circ/\text{с}$
Длина в вытянутом состоянии	1300 м
Масса полезной нагрузки	10 кг
Собственная масса	30 кг

Манипулятор обеспечивает приведение любого из звеньев манипулятора с точностью $0,005^\circ$ по угловому положению, что обеспечивает линейную точность позиционирования схвата манипулятора с точностью $\pm 0,1$ мм.

Рисунок 21 – Общий вид манипулятора.

На рисунке 22 приведена кинематическая схема манипулятора со связанными с каждым звеном системами координат, на рисунке 23 – кинематическая схема с линейными размерами и обозначением углов поворота звеньев.

Рисунок 22 – Кинематическая схема UR10 с выбранными системами координат

Рисунок 23 – Кинематическая схема UR10 с линейными размерами и обозначенными углами поворота звеньев

Совместно с блоком управления, манипулятор UR10 обеспечивает гибкие возможности по заданию угловых положений и угловых скоростей отдельных звеньев и манипулятора в целом. Имеющийся пульт управления обеспечивает достаточно удобный человеко-машинный интерфейс в части обеспечения первоначальных настроек манипулятора, программирования его траектории движения и т.д. Однако возможностей пульта управления недостаточно для реализации всех возможностей манипулятора (например, невозможно задать движение одновременно по двум степеням свободы).

Однако блок управления предоставляет эксплуатанту манипулятора ряд внешних интерфейсов для формирования команд управления и опроса конфигурации манипулятора внешним программным обеспечением. Одним из таких интерфейсов является стандартный сетевой интерфейс Ethernet, стандартной системой сетевой адресации TCP/IP.

Для управления манипулятором существует несколько классов команд управления, передаваемых совместно с одним или несколькими параметрами на присвоенный манипулятору IP-адрес и порт.

К данным командам относятся:

- Команда на переориентацию звеньев манипулятора (6 входных аргументов, соответствующих углам звеньев манипулятора + 1 аргумент – время переориентации);
- Команда на задание угловых скоростей разворота звеньев манипулятора (6 входных аргументов, соответствующих угловым скоростям каждого из звеньев);
- Команда на опрос параметров манипулятора (по данной команде возвращается информационный массив, содержащий углы и угловые скорости ориентации звеньев, потребляемый электродвигателем каждого звена ток, температуру в блоке электрики и т.д.)
- Команда на аварийный останов манипулятора.

Программное обеспечение автоматизированной калибровки акселерометра в необходимые моменты времени формирует команды управления, переориентируя UR10 в необходимые пространственные положения оси чувствительности акселерометра, выполняя расчет истинного значения проекции ускорения свободного падения для данной конфигурации манипулятора, измерение реального сигнала акселерометра и далее расчет и визуализацию этих данных.

Ориентацию системы координат (СК) схвата по отношению к базовой СК можно описать матричными преобразованиями. При варианте задания систем координат звеньев манипулятора UR10 в соответствии с рис. 22, 23 переход от одной предыдущей СК к последующей СК описывается следующими матрицами:

от СК₀ к СК₁

от СК₁ к СК₂

от СК₂ к СК₃

от СК₃ к СК₄

от СК₄ к СК₅

от СК₅ к СК₆

Проекция вектора G на оси системы координат схвата (СК₆) вычисляются путем матричного умножения следующего вида:

=

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		47

Для моделирования проекций векторов ускорения свободного падения G была подготовлена Excel-таблица (рис. 24), позволяющая задать проекции исходного вектора на оси стартовой (базовой) СК 0 и величины угловых перемещений в кинематических парах UR10 в соответствии с выбранными СК в каждой из пар (см. рис. 23) Результатом работы данной таблицы являются величины проекций исходного вектора на оси системы координат схвата (СК 6).

Рисунок 24 – Внешний вид Excel-таблицы для моделирования проекций
входного вектора в осях СК схвата

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		48

9 БЛОК ИНЕРЦИАЛЬНОЙ НАВИГАЦИОННОЙ СИСТЕМЫ (БЛОК ЧУВСТВИТЕЛЬНЫХ ЭЛЕМЕНТОВ)

На манипуляторе UR10 установлен блок чувствительных элементов, включающий в себя следующие компоненты:

- Интегрированный чувствительный элемент ADIS16485, включающий в себя трехосный акселерометр, трехосный датчик угловой скорости, термодатчик, аналого-цифровой преобразователь, логику реализации последовательного интерфейса SPI;
- Микропроцессор LPC2138, реализующий функции сбора, предварительной обработки данных от ЧЭ ADIS16485, а также передачу накопленной информации на наземную часть (персональный компьютер) по беспроводному интерфейсу;
- Модуль беспроводного интерфейса Bluetooth.

Персональный компьютер через блок приема данных осуществляет прием цифрового потока с информацией от каждого из 7 чувствительных элементов, обеспечивая визуализацию данных в режиме реального времени. Программное обеспечение данной части системы выполняет предварительную статистическую обработку информации (вычисление математического ожидания, дисперсии, минимума и максимума по каждому датчику), результаты статистической обработки передаются в программное обеспечение автоматизированной калибровки.

Общий вид блока чувствительных элементов, установленного на манипуляторе UR10 представлен на рис. 25.

									Лист
									49
Изм.	Лист	№ документа	Подпись	Дата	27.03.04.2017.219.28.00 ПЗ				

Рисунок 25 – Общий вид блока чувствительных элементов, установленного на манипуляторе UR10

Калибровка данного блока чувствительных элементов в части акселерометров выполняется путем переориентации оконечной степени подвижности манипулятора (угол φ_6 рис. 23) от программного обеспечения, функционирующего на персональном компьютере.

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		50

10 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ АВТОМАТИЗИРОВАННОЙ КАЛИБРОВКИ ЧУВСТВИТЕЛЬНЫХ ЭЛЕМЕНТОВ - АКСЕЛЕРОМЕТРОВ

В состав программного обеспечения автоматизированной калибровки входят следующие модули:

1. Модуль сбора данных с блока чувствительных элементов. Реализован в период 2015-2016. Предоставляет статистическую информацию основному модулю программы.

2. Модуль управления манипулятором UR10. Реализован в 2016 году. Обеспечивает программный интерфейс взаимодействия основного модуля программы с аппаратной частью манипулятора UR10.

3. Основной модуль программного обеспечения автоматизированной калибровки – разработан в рамках данной ВКР. Обеспечивает следующие функции:

а) формирование программы калибровки, состоящую из определенных шагов (этапов), каждому из которых соответствует определенная конфигурация манипулятора по углу ориентации схвата, а также дополнительных параметров: время ожидания после переориентации схвата, число замеров в данной ориентации и т.д.

б) прием, обработку и визуализацию получаемой от манипулятора и блока чувствительных элементов информации в табличном и графическом видах;

в) реализацию методов поиска экстремума;

г) взаимодействие с модулями пп.1,2.

Разработанное программное обеспечение написано на языке C++ с использованием графических и системных библиотек Qt 5.7. Выбор инструмента разработки обусловлен следующими факторами:

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		51

а) широко распространенная связка инструментов, используемая для создания графических кросс-платформенных приложений (функционирующих на операционных системах класса Windows и Linux); б) данный состав инструментов разработки является бесплатным, создание прикладных программ не требуют никаких лицензионных отчислений.

Общий алгоритм калибровки акселерометров на манипуляторе UR10 выглядит следующим образом (рис. 26):



Рисунок 26 – Общий алгоритм выполнения калибровки чувствительного элемента

На подготовительном этапе в ручном, автоматизированном или автоматическом режиме формируется программа проведения калибровки (набор тестовых воздействий) как функция от времени или ряда других параметров (например, температуры). При выполнении процедуры калибровки (этап 2 рис. 26) сформированные наборы последовательно воздействуют на чувствительный элемент. При этом, одновременно с задаваемым воздействием, фиксируется величина, измеряемая датчиком.

На финальном этапе калибровки по табличным результатам может быть сформирована функциональная зависимость выходного сигнала датчика от истинно задаваемой физической величины (ускорения).

С учетом возможностей манипулятора по переориентации блока чувствительных элементов реальный алгоритм калибровки на манипуляторе UR10 может выглядеть следующим образом (рис. 27)

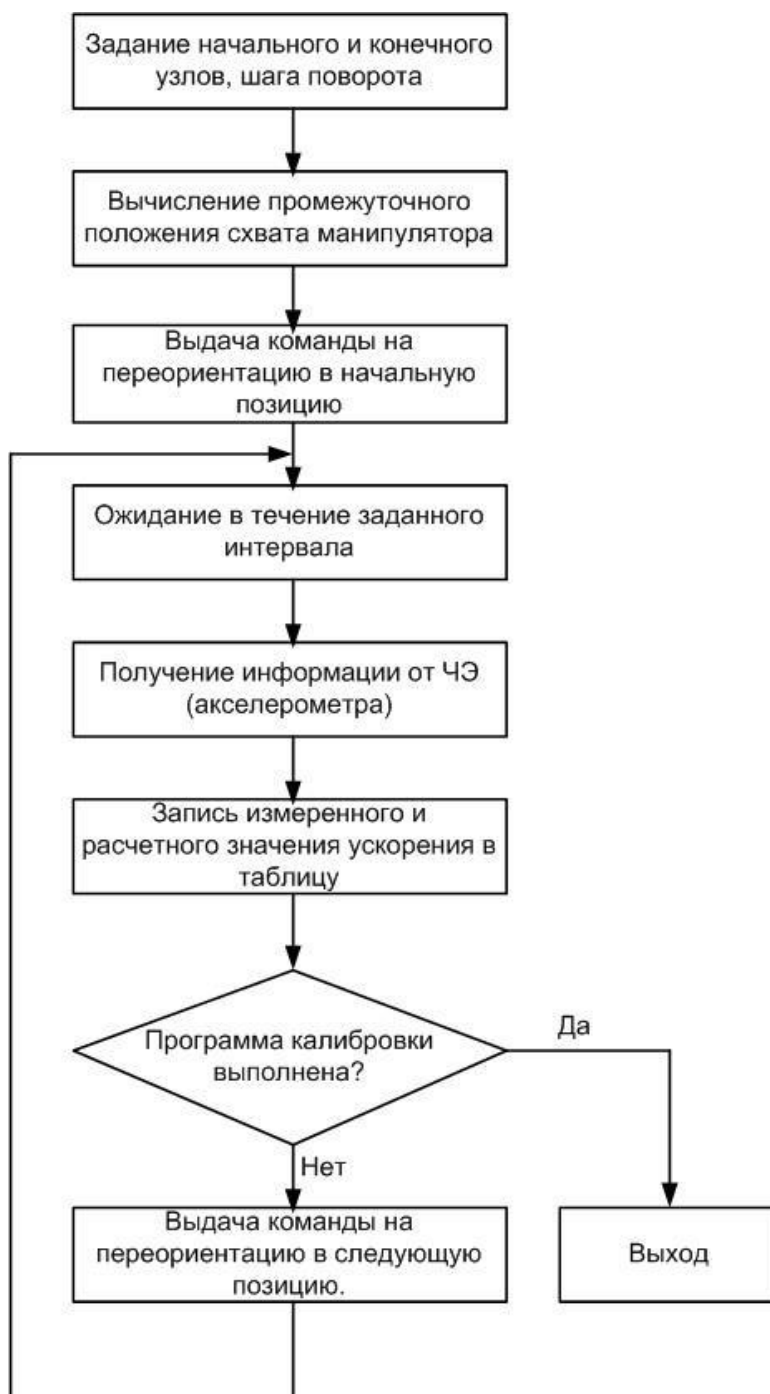


Рисунок 27 – Алгоритм подготовки и выполнения калибровки

Общие виды данных графиков приведены на рис. 28, 29, 30.

Рисунок 28 – График зависимости расчетного и измеренного значений ускорения от угла поворота схвата манипулятора

Рисунок 29 – График зависимости измеренного значения ускорения от истинного значения ускорения

Рисунок 30 - График зависимости абсолютной погрешности датчика от истинного значения ускорения

10.1 Реализация алгоритмов калибровки чувствительных элементов

Реализация данных алгоритмов выполнена в виде внутренних процедур и функций на языке C++, а с точки зрения пользователя представляет собой классическое графическое многооконное приложение с представлением информации в удобной форме. Общий вид программного обеспечения представлен на рис. 31.

Рисунок 31 – Общий вид окна ПО автоматизированной калибровки

Рисунок 32 – Подокна программного обеспечения

В основном окне программы содержатся несколько графических подокон (рис. 32):

1 – подокно формирования программы переориентации манипулятора при выполнении автоматизированной калибровки;

2 – подокно представления текущего состояния манипулятора (углы, угловые скорости звеньев, положение точки схвата);

3 – подокно визуализации этапов выполнения программы калибровки в табличном виде;

4 - подокно визуализации результатов калибровки в графическом виде.

Рисунок 33 - Формирования программы переориентации манипулятора при выполнении автоматической (автоматизированной) калибровки

Окно, представленное на рис. 33 позволяет пользователю выполнять общее управление манипулятором, а также формировать варианты программы калибровки чувствительных элементов.

По нажатию кнопки «Исходное положение» (поз. 1 рис. 33) манипулятор переводится в сложенное состояние. В данное состояние манипулятор переводится по окончании калибровки. По нажатию кнопки «Вертикальное положение» (поз. 2 рис. 33) манипулятор переводится в положение, в котором все звенья опущены вертикально вниз, готовые к работе.

Рисунок 34 - Формирования программы переориентации манипулятора при выполнении автоматической (автоматизированной) калибровки

В программном обеспечении есть возможность вручную задать наклон (поз. 3 рис. 34) и поворот (поз. 4 рис. 34) кисти (углы q_4 , q_5 манипулятора). Это дает возможность ручного переориентирования чувствительного элемента относительно вертикального положения.

Рисунок 35 - Формирования программы переориентации манипулятора при выполнении автоматической (автоматизированной) калибровки

Часть элементов окна управления предназначено для формирования программы (этапов) выполнения калибровки чувствительных элементов (рис. 35).

Оператор на подготовительном этапе формирует (задает) следующую исходную информацию: начальный угол схвата манипулятора (поз. 1 рис. 35), финальный угол схвата манипулятора (поз. 2 рис. 35). Количество переориентаций (поз. 3 рис. 35), отвечает за количество выполняемых переходов манипулятора на фиксированный шаг от исходного к финальному положению. Время ожидания (поз. 4 рис. 35) – промежуток время (секундах) через которое происходит расчет.

Рисунок 36 - Формирования программы переориентации манипулятора при выполнении автоматической (автоматизированной) калибровки

Кнопка «Подготовка программы поворота схвата» (поз. 1 рис. 36) отвечает за внесение данных на блок управления манипулятором. Кнопка «Запуск программы» (поз. 2 рис. 36) начинает исполнение управляющей программы.

Для контроля положения манипулятора используется подокно «Углы ориентации звеньев» (поз. 2 рис. 32) . В нем наглядно показаны углы $q_1, q_2, q_3, q_4, q_5, q_6$ основания манипулятора, плеча, предплечья, наклона кисти, поворота кисти, поворота схвата, а также линейные координаты схвата X, Y, Z . Основным контролируемым параметром манипулятора при проведении калибровки акселерометров в нашем случае является угол поворота схвата манипулятора q_6 (рис. 37).

Рисунок 37 – Углы ориентаций звеньев манипулятора UR10

Следующим окном в ПО является окно представления текущей информации о ходе выполнения программы калибровки, числовые значения регистрируемых параметров в табличном виде. В данном окне визуализируется следующая информация: количество всех шагов (переориентаций) программы калибровки (поз. 1 рис. 38), текущий выполняемый шаг программы калибровки (поз. 2 рис.

38). Текущая ориентация схвата в реальном времени отображается в поле «Ориентация схвата» (поз. 3 рис. 38).

Рисунок 38 – Табличный вид представления данных калибровки

Таблица состоит из шести столбцов,

1. Столбец «Дата/Время» служит для фиксации даты и времени начала выполнения очередного шага программы калибровки (начала переориентации схвата манипулятора).

2. Столбец «№ шага» – номер выполненного или выполняемого шага программы калибровки.

3. Столбец «Угол» отображает заданное значение угла поворота схвата манипулятора для конкретного шага программы калибровки.

4. Столбец « $A_{ист}$ » содержит расчетное значение проекции ускорения свободного падения на ось чувствительности акселерометра при заданном угле ориентации схвата манипулятора. Значение в данном столбце является эталоном при вычислении абсолютной погрешности.

5. Столбец « $A_{изм}$ » отображает измеренное акселерометром усредненное значение проекции ускорения свободного падения на ось чувствительности.

6. В столбце «Абс. погрешность» представлена абсолютная погрешность между истинным (расчетным) и измеренным значением проекции ускорения свободного падения, которая вычисляется по формуле:

$$\frac{A_{изм}}{A_{ист}}$$

При выполнении автоматизированной калибровки с помощью разработанного программного обеспечения существует несколько этапов выполнения шагов программы калибровки, сопровождающееся цветовой индикацией элементов

строки в соответствии со следующей принятой цветовой схемой и циклограммой выполнения шага калибровки:

При начале переориентации (разворота) схвата манипулятора соответствующая строка окрашивается в желтый цвет и сохраняет его в течении времени разворота схвата (рис. 39).

По окончании разворота выполняется цикл измерения значения сигнала с акселерометра, при этом строка окрашена в ярко-зеленый цвет (рис.40).

Все выполненные шаги программы калибровки с результатами измерений остаются окрашенными в светло-зеленый цвет (рис. 41).

Для наглядности приведены на рисунках 48,49,50.

Рисунок 39 - Выполнение разворота схвата и блока датчиков

Рисунок 40 - Цикл измерения ускорения свободного падения

Рисунок 41 - Выполненные шаги программы калибровки

На рисунке 42 показаны этапы выполнения программы калибровки.

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		59

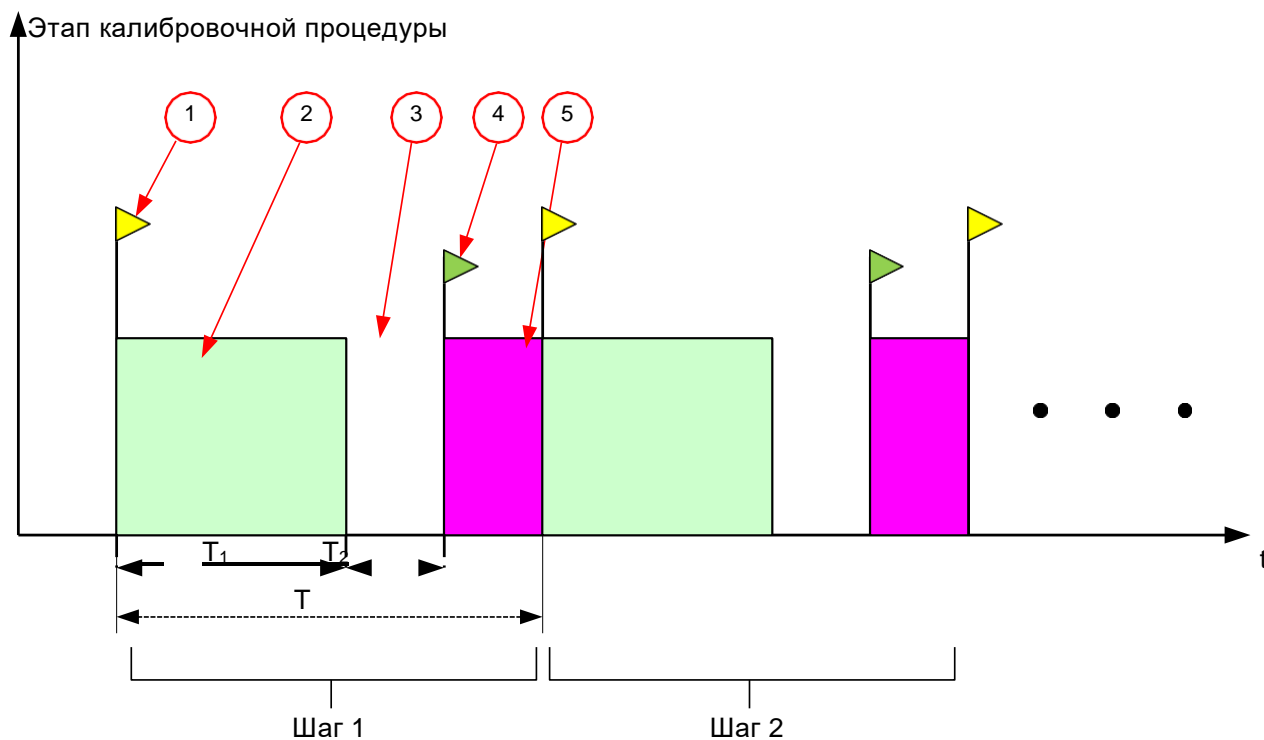


Рисунок 42 – Этапы выполнения программы калибровки

1 – Начало выполнения шага калибровочной программы. Формирование и передача команды манипулятора;

2 – Переориентация в течение времени T_1 ;

3 – Задержка в течение времени T_2 ;

4 – Получение информации от акселерометра;

5 – Обработка измеренного значения и запись в таблицу калибровки.

Непосредственно при выполнении программы калибровки помимо табличной визуализации данных выполняется формирование следующих графических зависимостей:

Зависимости расчетной и измеренной проекций ускорения свободного падения $A_{ист}$ $A_{изм}$ от угла поворота схвата $A_{ист} = f(\Delta)$, $A_{изм} = f(\Delta)$;

Зависимость измеренной проекции ускорения свободного падения от расчетного значения проекции $A_{изм}$ от $A_{ист}$ ($A_{изм} = f(A_{ист})$);

Зависимость абсолютной погрешности проекции ускорения свободного падения Δ от $A_{ист}$ ($\Delta = f(A_{ист})$).

Изм.	Лист	№ документа	Подпись	Дата

На рисунке 43 показан результат построения графика зависимости $A_{\text{ист}}$ $A_{\text{изм}}$ от угла поворота схвата, на котором видна разница между исходными данными и измеренными данными, которые измеряются акселерометром.

Рисунок 43 - Пример графика зависимости $A_{\text{ист}}$ $A_{\text{изм}}$ от угла поворота схвата

На рисунке 44 показан пример графика зависимости $A_{\text{изм}}$ от $A_{\text{ист}}$. Для идеального датчика функциональной зависимостью (передаточной характеристикой) является прямая пропорциональность, реальный датчик как правило отличается от идеального, что и видно на данном типе графика.

Рисунок 44 – Пример графика зависимости $A_{\text{изм}}$ от $A_{\text{ист}}$

На рисунке 45 изображена зависимость Δ от $A_{\text{ист}}$. Он показывает разницу истинного значения от измеренного.

Рисунок 45 - Пример график зависимости Δ от $A_{\text{ист}}$

10.2 Реализация методов поиска экстремума

Математические методы поиска экстремума одномерных и двумерных функций были приведены выше.

Реализация алгоритма поиска экстремума в рамках данной работы выполнена при следующих допущениях:

1. Блок акселерометров установлен на манипуляторе UR10, угол наклона плеча q_2 которого имитирует стартовое положение (наклон) подвижного объекта.
2. Блок акселерометров переориентируется путем задания угла q_6 (схват манипулятора), т.е. по одной степени свободы.
3. Выполняется поиск экстремума (минимума) одномерной функции $A_{\text{изм}} = f(q_6)$.

4. Алгоритм реализован по методу деления отрезка пополам и по методу золотого сечения.

5. Диапазон прикладываемого к акселерометру ускорения изменяется в диапазоне $[-g;g]$.

6. Величина измеренного значения ускорения округляется до величины 0,005. На вход алгоритма поступает осредненное за определенный интервал времени значение измеренного ускорения.

7. Критерием работы алгоритма является время переориентации оси чувствительности (количество переориентаций).

Общий алгоритм поиска экстремума описывается следующим образом:

1. Шаг 0: схват манипулятора переориентируется в угол $q_6 = 0^\circ$. Фиксируется величина сигнала акселерометра в данной ориентации схвата $A_{изм0}$.

2. Задается начальный шаг переориентации $k=10^\circ$.

3. Шаг i ($i=1\dots n$): выполняется переориентация схвата манипулятора на угол $i*k^\circ$. Фиксируется величина сигнала акселерометра в данной ориентации схвата $A_{изм i}$.

4. Выполняется сравнение величин $A_{изм0}$ и $A_{изм1}$. В случае, если значение $A_{изм1}$ больше $A_{изм0}$ выполняется переориентация схвата к углу $-k^\circ$ и дальнейшие шаги выполняются в «отрицательном направлении».

5. Выполняется пункты 3-5 до тех пор, пока величина $A_{изм i}$ будет меньше величины $A_{изм (i-1)}$ (величины ускорения на предыдущей итерации).

6. В случае равенства $A_{изм i}$ и $A_{изм (i-1)}$, округленных до первого знака после запятой, принимается решение о нахождении локального минимума (угла ориентации схвата манипулятора, при котором ось чувствительности акселерометра наиболее близка по направлению с направлением действия вектора ускорения свободного падения). При этом выполнение алгоритма заканчивается.

В случае $A_{изм i} > A_{изм (i-1)}$ выполняется переход к шагу 7.

7. Величина шага переориентации делится пополам ($k:=k/2$). Направление дальнейшей переориентации изменяется на противоположное. Выполняется переход на шаг 3.

Рисунок 46 поясняет работу алгоритма, графическое представление данного алгоритма приведено на рис. 47.

Рисунок 46 - Пример выполнения шагов алгоритма

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		63

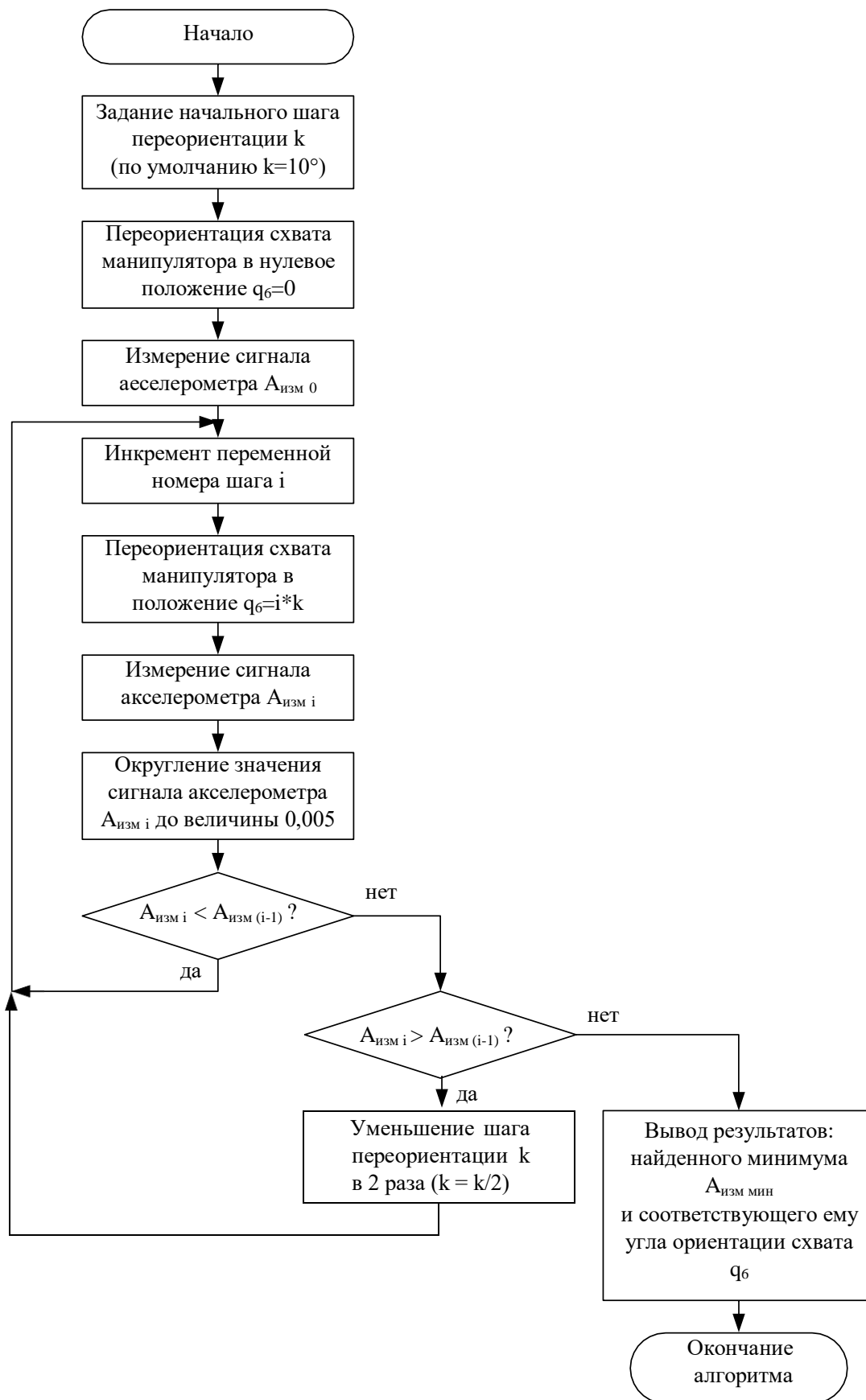


Рисунок 47 - Реализация алгоритма поиска минимума

Изм.	Лист	№ документа	Подпись	Дата

Программная реализация алгоритма выполнена в виде отдельного модуля программы. Внешний вид окна модуля приведен на рис. 49.

Модуль реализует следующие функции:

1. Возможность задания случайного начального отклонения манипулятора от вертикали путем формирования команды на изменение угла наклона плеча (q_1).
2. Возможность ввода начального шага переориентации схвата (угол q_6) при выполнении поиска экстремума.
3. Вывод последовательности выполняемых переориентаций в табличном виде с цветовой маркировкой выполняемых операций.
4. Графический вывод результатов выполнения шагов алгоритма в осях «Угол схвата q_6 » - «Измеренное ускорение $A_{изм\ i}$ »

На рис. 49 обозначены:

1. Кнопка задания произвольного (случайного) угла тангажа подвижного объекта (угол q_1 манипулятора).
2. Кнопка запуска алгоритма поиска экстремума.
3. Текущий тангаж подвижного объекта (угол q_1 манипулятора).
4. Начальное и текущее значение шага переориентации схвата манипулятора (q_6).
5. Мгновенное значение угла схвата манипулятора (q_6).
6. Мгновенное значение измеряемого ускорения.
7. Таблица представления результатов алгоритма поиска.
8. Столбец вывода даты и времени измерения.
9. Столбец вывода номера шага алгоритма.
10. Столбец заданного и фактического угла схвата манипулятора на данном шаге.
11. Столбец измеренного ускорения при данной ориентации схвата.

12. Маркированная строка, соответствующая ориентации (номеру шага), при которой происходит изменение величины и знака дальнейшей переориентации.

Графический вид результатов работы алгоритма представлен на рис. 48. Точки графика строятся по мере выполнения шагов алгоритма в режиме реального времени.

Рисунок 48 - Графическое представление результатов работы алгоритма поиска экстремума

Рисунок 49 - Внешний вид окна модуля поиска экстремума

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		66

11 ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

Для подтверждения правильности функционирования разработанных алгоритмов и программного обеспечения на действующем манипуляторе UR10 и установленном на нем блоке инерциальных чувствительных элементов (датчиков угловых скоростей и акселерометров) были проведены различные по методике и достигаемому результату серии экспериментов.

1. Серия экспериментов по получению (и подтверждению) передаточной характеристики акселерометра (погрешностей акселерометра).

В данной серии экспериментов выполнялась следующая процедура:

- а) плечо и локоть манипулятора ориентировались в вертикальное положение;
- б) оператор задавал диапазон изменений угла схвата манипулятора и количество его переориентаций;
- в) манипулятор в автоматическом режиме выполнял программу переориентаций с фиксацией показаний акселерометра;

В данной серии были выполнены 3 экспериментальных измерения с целью подтверждения сходства передаточной характеристики акселерометра (или зависимости погрешности от измеряемого истинного ускорения). Результаты приведены на рис. 50-52.

Рисунок 50 – Результат эксперимента №1

Рисунок 51 – Результат эксперимента №2

Рисунок 52 – Результат эксперимента №3

Численные результаты экспериментов приведены в таблице 3.

Таблица 3. Сравнительная таблица данных трех экспериментов.

Результаты, полученные в ходе данного эксперимента подтверждают функционирование алгоритма получения передаточных характеристик акселерометра на манипуляторе UR10. Из полученных графиков видно, что для одного и того же установленного на манипуляторе акселерометра, при единой методике выполнения эксперимента результаты являются схожими.

2. Серия экспериментов по оценке алгоритмов поиска положения экстремума (минимума) измеряемого акселерометром ускорения при случайной ориентации подвижного объекта (в нашем случае манипулятора) по углу тангажа.

Серия экспериментов выполнялась следующим образом:

- а) на подготовительном этапе случайным образом задается начальное положение плеча манипулятора (угол q_1);
- б) задавался начальный шаг переориентации схвата манипулятора при работе алгоритма поиска экстремума. Начальное значение шага – 10° ;
- в) запускался автоматический алгоритм поиска экстремума (минимума).

При проведении данной серии экспериментов приняты следующие допущения:

- а) измеряемое акселерометром ускорение округляется (квантуется по уровню) до величины, кратной $0,005 \text{ м/с}^2$;
- б) в случае необходимости уменьшения шага переориентации схвата манипулятора (см. рис. 47) величина этого изменения составляет 50% от исходной величины шага (т.е. при начальном шаге 10° последующие шаги изменения составят $5^\circ, 2,5^\circ, \dots$);
- в) алгоритм завершает работу, если значение измеренного акселерометром ускорения на текущем и предыдущем шаге измерения равны.

Эксперименты проводились при следующих случайно сформированных начальных положения манипулятора:

$$q_{1\ 1} = -101^\circ$$

$$q_{1\ 2} = -88,7^\circ$$

$$q_{1\ 3} = -70,2^\circ$$

Результаты работы алгоритма представлены на рисунках 53-55.

Рисунок 53 – Результат калибровки эксперимента №1

Рисунок 54 – Результат калибровки эксперимента №2

Рисунок 55 – Результат калибровки эксперимента №3

Результаты выполнения данной серии экспериментов подтверждают функционирование алгоритма поиска экстремума с использованием метода деления отрезка пополам.

В дальнейшем возможно добавление дополнительных математических алгоритмов для оптимизации времени поиска.

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		70

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы на тему «Программное обеспечение автоматической калибровки датчиков инерциальной информации» проведен анализ исходной информации (технического задания), анализ видов чувствительных элементов, их функциональных и точностных характеристик, рассмотрены общие подходы к проведению калибровок чувствительных элементов.

На первом этапе реализации программного обеспечения сформирована структура ПО, определен состав функциональных модулей. Отладка программного обеспечения выполнена на рабочем экземпляре стенда калибровки чувствительных элементов лаборатории «Моделирования динамики движения летательных аппаратов» в составе робота-манипулятора UR10 и блока инерциальной навигационной системы. Опробованы методы оптимизации ориентации оси чувствительности акселерометра вдоль направления действия ускорения свободного падения.

Разработанное программное обеспечение позволяет проводить исследование характеристик чувствительных элементов – акселерометров в рамках выполнения лабораторных и курсовых работ дисциплин учебного курса направления подготовки 27.03.04 «Управление в технических системах».

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		71

ЛИТЕРАТУРА

1. Соловьев В.И. Инерциальные навигационные системы: учеб. пособие / В.И. Соловьев, П.Г. Шабалов. – Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2011. – 72 с.: ил.
2. Лысенко Л.Н. – Наведение и навигация баллистических ракет: Учеб. пособие. – М.:Изд-во МГТУ им. Н.Э. Баумана, 2007. - 672 с.:ил.
3. В.И. Соловьев, П.Г Шабалов Инерциальные навигационные системы
4. Мелешко В.В., Нестеренко О.И. М-47 Бесплатформенные инерциальные навигационные системы. Учебное пособие. – Кировоград: ПОЛИМЕД - Сервис, 2011. – 171с.
5. Источник: https://electric-220.ru/news/princip_raboty_akselerometra/2014-04-14-579
6. http://kit-e.ru/articles/sensor/2006_3_10
7. http://iitt.fvt.sfedu.ru/files/documents/up/up_datchiki_uskoreniya.pdf
8. <https://www.vibrtest.ru/полезная-информация/виды-пьезоэлектрических-акселерометров/>
9. <http://masters.donntu.org/2009/eltf/kiseliov/library/translate.htm>
10. https://ru.wikipedia.org/wiki/датчик_угловой_скорости
11. https://ru.wikipedia.org/wiki/алгоритм_бройдена_—_флетчера_—_гольдфарба_—_шанно
12. http://otherreferats.allbest.ru/programming/00056709_0.html
13. https://ru.wikipedia.org/wiki/математическая_статистика

ПРИЛОЖЕНИЕ А

Листинг программы

Файл main.cpp

```
#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.showMaximized();
    return a.exec();
}
```

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		73

Файл mainwindow.cpp

```
#include "mainwindow.h"
```

```
MainWindow::MainWindow(QWidget *parent)
```

```
: QMainWindow(parent)
```

```
{
```

```
    commClass2 = new UR10MiddleControClass(this,"192.168.1.2",100);
```

```
    commClass2->setHomePosition(45.0,0.0,-160.0,-90.0,0.0,0.0);
```

```
    commClass2->setStartPosition(0,-90,-65,-115,0,0);
```

```
    commClass2->setVerticalPosition(0,-90,0,-180,0,90);
```

```
    commClass2->setMotionProgramCycled(false);
```

```
    commClass2->startPoll(true);
```

```
    commClass2->buildProgram(5,0,40,5,5,10);
```

```
    binsSocket = new QTcpSocket(this);
```

```
    connect(binsSocket,SIGNAL(connected()),this,SLOT(binsConnected()));
```

```
    connect(binsSocket,SIGNAL(readyRead()),this,SLOT(readBINSData()));
```

```
connect(binsSocket,SIGNAL(error(QAbstractSocket::SocketError)),this,SLOT(binsCo  
nnectionError(QAbstractSocket::SocketError)));
```

```
    pollPerformed = false;
```

```
    binsSocket->connectToHost("192.168.1.5",24000);
```

```
    lcdViewer = new QLCDViewer(this, "Манипулятор 1");
```

```
    positions = new TypicalPositions(this);
```

```
    calibrationWidget = new CalibrationWidget(this);
```

```
    calibrationResultsWidget2 = new calibrationResultsWidget(this);
```

```
    plot1 = new MyCustomPlot(this,2);
```

```
    finderWidget = new FinderWidget(this);
```

```
    finderPlotWidget = new FinderPlotWidget(this);
```

```
    windowLCDViewer = new QMdiSubWindow(this);
```

```
    windowTypicalPositions = new QMdiSubWindow(this);
```

```
    windowCalibrationWidget = new QMdiSubWindow(this);
```

```
    windowCalibrationResultWidget = new QMdiSubWindow(this);
```

```
    windowFinderWidget = new QMdiSubWindow(this);
```

```
    windowFinderPlotWidget = new QMdiSubWindow(this);
```

```
    mdiArea = new QMdiArea(this);
```

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		74


```
windowLCDViewer->setWidget(lcdViewer);
windowLCDViewer->setWindowTitle("УГЛЫ ориентации звеньев");
windowLCDViewer->move(140,0);
windowLCDViewer->resize(500,1000);
```

```
windowTypicalPositions->setWidget(positions);
windowTypicalPositions->setWindowTitle("Типовые положения манипулятора");
windowTypicalPositions->move(0,0);
windowTypicalPositions->resize(200,200);
```

```
windowCalibrationWidget->setWidget(calibrationWidget);
windowCalibrationResultWidget->setWidget(calibrationResultsWidget2);
windowFinderWidget->setWidget(finderWidget);
windowFinderWidget->setWindowTitle("Окно управления поиском экстремума");
```

```
windowFinderPlotWidget->setWidget(finderPlotWidget);
windowFinderPlotWidget->setWindowTitle("График результатов автопоиска экстремума");
```

```
mdiArea->addSubWindow(windowLCDViewer);
mdiArea->addSubWindow(windowTypicalPositions);
mdiArea->addSubWindow(windowCalibrationWidget);
mdiArea->addSubWindow(windowCalibrationResultWidget);
mdiArea->addSubWindow(windowFinderWidget);
mdiArea->addSubWindow(windowFinderPlotWidget);
```

```
this->setCentralWidget(mdiArea);
```

```
connect(commClass2,SIGNAL(newURdataStringArray(QStringList)),lcdViewer,SLOT(setDisplayValues(QStringList)));
```

```
connect(commClass2,SIGNAL(newURdataStringArray(QStringList)),calibrationWidget,SLOT(manipulatorOrientation(QStringList)));
```

```
connect(commClass2,SIGNAL(startedProgramStep(CalibrationEntry)),calibrationWidget,SLOT(addCalibrationEntry(CalibrationEntry)));
```

```
connect(commClass2,SIGNAL(manipulatorOrientated(CalibrationEntry)),calibrationWidget,SLOT(manipulatorOrientated(CalibrationEntry)));
```

```
connect(positions,SIGNAL(homePosition()),commClass2,SLOT(moveToHomeDefault
()));
```

```
connect(positions,SIGNAL(startPosition()),commClass2,SLOT(moveToStartDefault()
));
```

```
connect(positions,SIGNAL(verticalPosition()),commClass2,SLOT(moveToVerticalDef
ault()));
```

```
connect(positions,SIGNAL(newJointPosition(int,double,double)),commClass2,SLOT(
moveSingleJointAbsolute(int,double,double));
```

```
    connect(positions,SIGNAL(startProgram()),commClass2,SLOT(startProgram0()));
```

```
connect(positions,SIGNAL(buildProgram(int,double,double,int,double,double)),comm
Class2,SLOT(buildProgram(int,double,double,int,double,double));
```

```
connect(calibrationWidget,SIGNAL(newCalibrationData(QList<CalibrationEntry>)),ca
librationResultsWidget2,SLOT(addData(QList<CalibrationEntry>));
```

```
connect(commClass2,SIGNAL(newRandomPitch(double)),finderWidget,SLOT(newRa
ndomPitch(double));
```

```
connect(finderWidget,SIGNAL(requestNewRandomPitch()),commClass2,SLOT(setRa
ndomMainPitch));
```

```
connect(finderWidget,SIGNAL(newSingleJointAbsolute(int,double,double)),commCla
ss2,SLOT(moveSingleJointAbsolute(int,double,double));
```

```
connect(commClass2,SIGNAL(newURdataStringArray(QStringList)),finderWidget,SL
OT(manipulatorOrientation(QStringList));
```

```
connect(finderWidget,SIGNAL(newCalibrationData(QList<CalibrationEntry>)),finder
PlotWidget,SLOT(addData(QList<CalibrationEntry>));
```

```
    setWindowTitle("UR10 control");
```

```
}
```

```
MainWindow::~MainWindow()
```

```
{
```

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		76

```

}

void MainWindow::tempSlot(int num)
{
    qDebug("Current program step is: "+QString::number(num).toLatin1());
}

void MainWindow::tempSlot2(QStringList str)
{
    qDebug("%d",str.length());
}

void MainWindow::startPoll(bool start)
{
    if (start)
    {
        pollCount = 0;
        pollTimer->start();
    }
    else
        pollTimer->stop();
}

void MainWindow::startSave(bool startSave)
{
    if (startSave)
    {
        saveToFile = true;
        createFileName();
    }
    else
    {
        saveToFile = false;
        if (outFile->isOpen())
            outFile->close();
    }
}

void MainWindow::setSliderAngles(QStringList list)
{
    float q1f,q2f,q3f,q4f,q5f,q6f;
    int q1, q2, q3, q4, q5, q6;
}

```

```

if (pollPerformed == false)
{
    q1f = list.at(0).toFloat();
    q2f = list.at(1).toFloat();
    q3f = list.at(2).toFloat();
    q4f = list.at(3).toFloat();
    q5f = list.at(4).toFloat();
    q6f = list.at(5).toFloat();

    q1 = deg360to180(q1f);
    q2 = deg360to180(q2f);
    q3 = deg360to180(q3f);
    q4 = deg360to180(q4f);
    q5 = deg360to180(q5f);
    q6 = deg360to180(q6f);

    okno->setInitialSliders(q1,q2,q3,q4,q5,q6);
    pollPerformed = true;
}
}

void MainWindow::qmlTestSlot(QString test)
{
    qDebug(test.toLatin1());
}

void MainWindow::readBINSData()
{
    QByteArray ba;
    qDebug("MainWindow::readBINSData()");
    ba = binsSocket->readAll();
    // qDebug(binsSocket->readAll());
    qDebug(ba);
    calibrationWidget->binsData(ba);
    finderWidget->binsData(ba);
}

void MainWindow::binsConnected()
{
    qDebug("binsConected()");
}

void MainWindow::binsConnectionError(QAbstractSocket::SocketError error)

```

```

{
    qDebug("bins connection error");
}

void MainWindow::createFileName()
{
    QDir dataPath;

    if (!dataPath.exists(DATA_PATH))
        dataPath.mkdir(DATA_PATH);

    outFile = new
    QFile(DATA_PATH+QDateTime::currentDateTime().toString("yyyy_MM_dd_hh_m
m_ss")+ ".dat");
    fileNameValueLabel->setText(outFile->fileName());
    qDebug(outFile->fileName().toLatin1());

    if (!outFile->open(QIODevice::WriteOnly | QIODevice::Text))
        return;

    outTextStream = new QTextStream(outFile);
    outTextStream->setCodec("UTF-8");

    channelNamesFull.prepend("Время (мс)");
    channelNamesFull.prepend("Время (с)");
    channelNamesFull.prepend("Дата");

    QString tmp = channelNamesFull.join("\t")+"\n";
    *outTextStream << tmp;
}

float MainWindow::deg360to180(float deg)
{
    float deg180 = deg;
    if (deg180 >= 180.0)
        deg180 -=360.0;
    return(deg180);
}

```

					27.03.04.2017.219.28.00 ПЗ	Лист
Изм.	Лист	№ документа	Подпись	Дата		79

Файл calibrationwidget.cpp

```
#include "calibrationwidget.h"

CalibrationWidget::CalibrationWidget(QWidget *parent) : QWidget(parent)
{
    initGUI();
}

void CalibrationWidget::addCalibrationEntry(CalibrationEntry entry)
{
    calibrationList.append(entry);

    tableWidget->item(entry.calibrationStep,0)-
>setText(entry.dateTime.toString("dd.MM.yyyy hh:mm:ss"));
    tableWidget->item(entry.calibrationStep,1)-
>setText(QString::number(entry.calibrationStep));
    tableWidget->item(entry.calibrationStep,2)-
>setText(QString::number(entry.calibrationAngle));
    tableWidget->item(entry.calibrationStep,3)-
>setText(QString::number(entry.calibrationIdealValue));
    tableWidget->resizeColumnsToContents();

    currentStepLCD->display(entry.calibrationStep);
    totalStepsLCD->display(entry.calibrationSteps);

    if (entry.calibrationStep > 0)
//    colorRow(Qt::lightGray,entry.calibrationStep-1);
    colorRow(QColor(Qt::green).light(180),entry.calibrationStep-1);

    colorRow(Qt::yellow,entry.calibrationStep);
}

void CalibrationWidget::manipulatorOrientated(CalibrationEntry entry)
{
    calibrationList.last().calibrationMeasuredValue = binsDataString.at(1).toDouble();
    tableWidget->item(entry.calibrationStep,4)->setText(binsDataString.at(1));
    tableWidget->item(entry.calibrationStep,5)-
>setText(QString::number(calibrationList.last().calibrationIdealValue-
calibrationList.last().calibrationMeasuredValue));
    colorRow(Qt::green,entry.calibrationStep);
    tableWidget->resizeColumnsToContents();
    emit newCalibrationData(calibrationList);
}
```

```

}

void CalibrationWidget::manipulatorOrientation(QStringList angles)
{
    orientationLCD->display(angles.at(5));
}

void CalibrationWidget::binsData(QByteArray ba)
{
    qDebug(ba);
    QString tmpStr(ba);
    binsDataString=tmpStr.split('\t');
    qDebug(binsDataString.at(1).toLatin1());
    accelerationLCD->display(binsDataString.at(1).toDouble());
}

void CalibrationWidget::initGUI()
{
    QStringList headerList {"Дата/Время", "№ шага", "Угол", "Аист", "Аизм", "Абс.
погрешность"};
    QVBoxLayout *mainLayout = new QVBoxLayout(this);
    QHBoxLayout *headerLayout = new QHBoxLayout(this);
    QFormLayout *formLayout = new QFormLayout(this);

    currentStepLabel = new QLabel("Текущий шаг программы",this);
    totalStepsLabel = new QLabel("Всего шагов программы",this);

    currentStepLCD = new QLCDNumber(this);
    currentStepLCD->setSegmentStyle(QLCDNumber::Flat);
    totalStepsLCD = new QLCDNumber(this);
    totalStepsLCD->setSegmentStyle(QLCDNumber::Flat);
    orientationLCD = new QLCDNumber(this);
    orientationLCD->setSegmentStyle(QLCDNumber::Flat);
    accelerationLCD = new QLCDNumber(this);
    accelerationLCD->setSegmentStyle(QLCDNumber::Flat);

    headerLayout->addWidget(currentStepLabel);
    headerLayout->addWidget(currentStepLCD);
    headerLayout->addWidget(totalStepsLabel);
    headerLayout->addWidget(totalStepsLCD);

    formLayout->addRow("Ориентация схвата",orientationLCD);
    formLayout->addRow("Величина ускорения",accelerationLCD);

```

```

tableWidget = new QTableWidgetItem(20,6,this);
for (int i=0;i<tableWidget->columnCount();i++)
{
    for (int j=0;j<tableWidget->rowCount();j++)
    {
        QTableWidgetItem *item = new QTableWidgetItem("-");
        tableWidget->setItem(j,i,item);
    }
}
tableWidget->setHorizontalHeaderLabels(headerList);
tableWidget->resizeColumnsToContents();
mainLayout->addLayout(headerLayout);
mainLayout->addLayout(formLayout);
mainLayout->addWidget(tableWidget);
setLayout(mainLayout);
}

void CalibrationWidget::colorRow(QColor color, int row)
{
    for (int i=0;i<tableWidget->columnCount();i++)
    {
        tableWidget->item(row,i)->setBackgroundColor(color);
    }
}

```


Файл finderwidget.cpp

```
#include "finderwidget.h"
```

```
FinderWidget::FinderWidget(QWidget *parent) : QWidget(parent)
{
    initGUI();
}
```

```
void FinderWidget::newRandomPitch(double pitch)
{
    pitchLCD->display(pitch);
}
```

```
void FinderWidget::binsData(QByteArray ba)
{
    qDebug(ba);
    QString tmpStr(ba);
    binsDataString=tmpStr.split('\t');
    qDebug(binsDataString.at(1).toLatin1());
    accelerationLCD->display(binsDataString.at(1).toDouble());
    measuredAcceleration = round(binsDataString.at(1).toDouble()*200)/200;
}
```

```
void FinderWidget::manipulatorOrientation(QStringList angles)
{
    pitchLCD->display(angles.at(1));
    orientationLCD->display(angles.at(5));
}
```

```
void FinderWidget::startSearchProgram()
{
    qDebug("FinderWidget::startSearchProgram()");
    emit
newSingleJointAbsolute(UR10CommunicationClass::JOINT_WRIST_3,0.0,3.0);
    pollTimer->start(5000);
    wristStep = stepLineEdit->text().toDouble();
    currentStep = 0;
    currentAngle = 0.0;
}
```

```
void FinderWidget::pollTimerSlot()
{
```

```

qDebug("FinderWidget::pollTimerSlot()");
// pollTimer->stop();

tableWidget->item(currentStep,0)-
>setText(QDateTime::currentDateTime().toString("dd.MM.yyyy hh:mm:ss"));
tableWidget->item(currentStep,1)->setText(QString::number(currentStep));
tableWidget->item(currentStep,2)->setText(QString::number(currentAngle)+
("+QString::number(orientationLCD->value())+"));
tableWidget->item(currentStep,3)-
>setText(QString::number(measuredAcceleration));
tableWidget->resizeColumnsToContents();

colorRow(QColor(Qt::green).light(180),currentStep);

measuredAccelerations.append(measuredAcceleration);

CalibrationEntry entry;
entry.calibrationAngle = currentAngle;
entry.calibrationMeasuredValue = measuredAcceleration;

calibrationList.append(entry);

currentStep++;
currentAngle += wristStep;

emit newCalibrationData(calibrationList);

if (currentStep>2)
{
    if (measuredAccelerations.at(currentStep-1) >
measuredAccelerations.at(currentStep-2))
    {
        currentAngle -= wristStep;
        wristStep = -wristStep/2;
        currentAngle += wristStep;
        stepLineEdit->setText(QString::number(wristStep));
        colorRow(Qt::yellow,currentStep-1);
        emit
newSingleJointAbsolute(UR10CommunicationClass::JOINT_WRIST_3,currentAngle,
3.0);
    }
    else if (measuredAccelerations.at(currentStep-1) ==
measuredAccelerations.at(currentStep-2))

```

```

    {
        pollTimer->stop();
    }
    else
    {
        emit
newSingleJointAbsolute(UR10CommunicationClass::JOINT_WRIST_3,currentAngle,
3.0);
    }
}
else
{
    emit
newSingleJointAbsolute(UR10CommunicationClass::JOINT_WRIST_3,currentAngle,
3.0);
}
}

void FinderWidget::stepTimerSlot()
{

}

void FinderWidget::initGUI()
{
    QStringList headerList {"Дата/Время","№ шага", "Угол схвата", "Аизм"};
    QVBoxLayout *mainLayout = new QVBoxLayout(this);
    QHBoxLayout *headerLayout = new QHBoxLayout(this);
    QFormLayout *formLayout = new QFormLayout(this);

    stepLineEdit = new QLineEdit("10.0",this);

    randomPitchPushButton = new QPushButton("Задать случайный угол
тангажа",this);
    startSearchPushButton = new QPushButton("Начать программу поиска
экстремума",this);
//    startSearchPushButton = new QPushButton("Начать поиск экстремума",this);

connect(randomPitchPushButton,SIGNAL(pressed()),this,SIGNAL(requestNewRando
mPitch()));

```

```
connect(startSearchPushButton,SIGNAL(pressed()),this,SLOT(startSearchProgram()));
```

```
pitchLabel = new QLabel("Тангаж подвижного объекта (q1)",this);  
pitchLCD = new QLCDNumber(this);  
pitchLCD->setSegmentStyle(QLCDNumber::Flat);  
pitchLCD->setDigitCount(7);
```

```
orientationLCD = new QLCDNumber(this);  
orientationLCD->setSegmentStyle(QLCDNumber::Flat);  
accelerationLCD = new QLCDNumber(this);  
accelerationLCD->setSegmentStyle(QLCDNumber::Flat);  
orientationLCD->setDigitCount(7);  
accelerationLCD->setDigitCount(7);
```

```
headerLayout->addWidget(pitchLabel);  
headerLayout->addWidget(pitchLCD);
```

```
formLayout->addRow("Шаг изменения угла при поиске",stepLineEdit);  
formLayout->addRow("Ориентация схвата",orientationLCD);  
formLayout->addRow("Величина ускорения",accelerationLCD);
```

```
tableWidget = new QTableWidgetItem(100,4,this);  
for (int i=0;i<tableWidget->columnCount();i++)  
{  
    for (int j=0;j<tableWidget->rowCount();j++)  
    {  
        QTableWidgetItem *item = new QTableWidgetItem("-");  
        tableWidget->setItem(j,i,item);  
    }  
}  
tableWidget->setHorizontalHeaderLabels(headerList);  
tableWidget->resizeColumnsToContents();
```

```
mainLayout->addWidget(randomPitchPushButton);  
mainLayout->addWidget(startSearchPushButton);  
mainLayout->addLayout(headerLayout);  
mainLayout->addLayout(formLayout);  
// mainLayout->addWidget(startSearchPushButton);  
mainLayout->addWidget(tableWidget);
```

```
stepTimer = new QTimer(this);
```

```

pollTimer = new QTimer(this);

connect(stepTimer,SIGNAL(timeout()),this,SLOT(stepTimerSlot()));
connect(pollTimer,SIGNAL(timeout()),this,SLOT(pollTimerSlot()));

setLayout(mainLayout);
}

void FinderWidget::colorRow(QColor color, int row)
{
    for (int i=0;i<tableWidget->columnCount();i++)
    {
        tableWidget->item(row,i)->setBackgroundColor(color);
    }
}

```