

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 22М.07-мм

Лень Юлия Александровна

Разработка требований к
программно-аппаратной платформе для
управления роботами

Отчёт по учебной практике

Научный руководитель:
д.ф.-м.н., проф., О.Н. Граничин

Санкт-Петербург
2023

Оглавление

1. Введение	3
2. Постановка задачи	4
3. Требования к веб-серверу	5
3.1. Обзор программно-аппаратной платформы	5
3.2. Программно-аппаратная платформа для работы с роботами	8
3.3. Требования к внешнему(веб) серверу	8
4. Разработка макетов	10
4.1. Навигационное меню	10
4.2. Главная	11
4.3. О лаборатории	11
4.4. Симулятор	11
4.5. Работа с роботами	13
4.6. Личный кабинет	13
5. Описание сценариев	14
6. Метод знако-возмущенных сумм	15
6.1. Модель наблюдений	15
6.2. Метод знако-возмущенных сумм	15
7. Заключение	18
Список литературы	19

1. Введение

В современном мире интерес к беспилотным летательным аппаратам (БПЛА, БЛА) сильно возрос. БПЛА применяются как военными, так и гражданскими лицами, помогая решать сложные или опасные для человека задачи. Рассмотрим некоторые из них.

Среди военных задач одной из распространенных является воздушная разведка: поиск мин, расположений орудий противника и других стратегически важных объектов. Среди гражданских задач можно выделить поиск людей на сложных местностях, проверка оборудования в труднодоступных местах, исследование местности для составления более точных карт.

Главным преимуществом БЛА является его маневренность, возможность исследовать местность или участок, представляющий опасность или вызывающий физические затруднения для человека. Отсюда следует главная ценность использования таких средств — безопасность оператора (человека, управляющего этим аппаратом). Однако в сравнении с пилотируемым летательным аппаратом, стоимость разработки и обслуживания такого оборудования выше. [2]

Методов, позволяющих беспилотникам передвигаться самостоятельно и корректно определять свое положение в пространстве достаточно большое количество. В силу дороговизны апробации на физическом устройстве реализация некоторых методов остается только на бумаге. Поэтому есть необходимость в разработке системы, которая позволит моделировать разные алгоритмы поведения БПЛА до его физической реализации, чтобы ценить для разных задач, какой подход будет потенциально лучше работать. Также наличие открытой платформы для таких исследований позволит увеличить число разрабатываемых алгоритмов, которые можно будет применить на практике.

В рамках работы будет предложен макет открытой платформы. Для апробации будет использован метод знако-вмощенных сумм (SPS), который будет модифицирован для задачи корректировки положения аппарата на местности.

2. Постановка задачи

Целью работы является разработка требований к открытой программно-аппаратной платформе (ПАП). Для достижения этой цели необходимо решить следующие задачи:

- разработать требования к веб-серверу;
- разработать макеты страниц;
- описать сценарии взаимодействия с веб-сервером;
- апробировать модификацию метода знако-возмущенных сумм для корректировки положения БПЛА в пространстве.

3. Требования к веб-серверу

3.1. Обзор программно-аппаратной платформы

Для информатизации общества и бизнеса необходим широкий спектр программно-аппаратных средств, включая компьютеры и средства связи. Эти технические средства обеспечивают прием и передачу информации разных видов (речь, текст, графика) с помощью слуха, осязания и зрения. Для связи с человеком используются разнообразные устройства, такие как дисплеи, клавиатуры, мыши и планшеты. Технические средства связи обеспечивают передачу информации во внешней среде. На предприятиях могут использоваться множество компьютеров для хранения и обработки информации. Программные средства обеспечивают обработку данных в экономической информационной системе (ЭИС) и включают общее и прикладное программное обеспечение.

Программные средства обеспечивают обработку данных в экономической информационной системе (ЭИС) и состоят из общего и прикладного программного обеспечения. Схема представлена на рисунке 1.

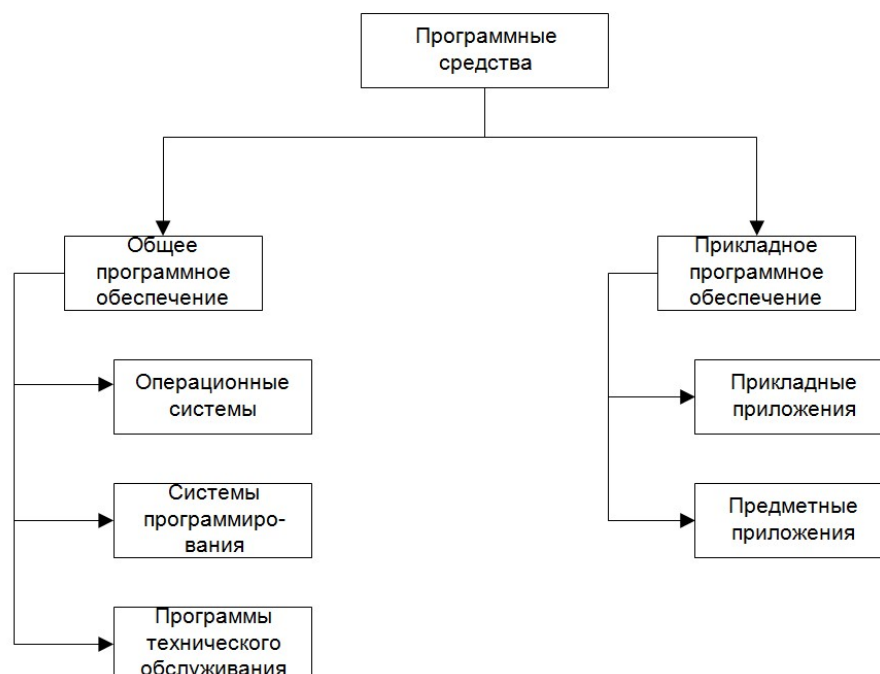


Рис. 1: Виды программных средств

Распространенное программное обеспечение включает операцион-

ные системы, системы программирования и программы технического обслуживания.

Система программирования в основном используется при проектировании EIS и представляет собой язык программирования и программы перевода (трансляторы, компиляторы, интерпретаторы) с этого языка в машинный код. Наиболее перспективным является объектно-ориентированное программирование. Объектно-ориентированное программирование недавно стало визуальным (VO-Визуальные объекты). Это означает, что интерфейс среды программирования меняется. Как правило, визуальный редактор добавляется к существующему пакету для добавления визуального объектно-ориентированного интерфейса.

Программы технического обслуживания - это типовые программы, предназначенные для наладки, планирования и диагностики функционирования аппаратуры ВМ [7].

Прикладное программное обеспечение (ПО) включает в себя различные приложения и пакеты программ, которые используются во множестве сфер человеческой деятельности. Оно может быть универсальным, таким как текстовые процессоры и системы управления базами данных, или представлять собой специализированные пакеты программ, предназначенные для решения определенных задач или подсистем экономических информационных систем.

Платформа представляет собой тип оборудования и программного обеспечения, на котором может быть установлено приобретаемое информационное технологическое средство. Она включает в себя компьютерный тип и операционную систему, которая работает на этом компьютере. Некоторые информационные технологии не требуют дополнительного оборудования и программных средств, в то время как другие зависят от них, например, сетевые технологии зависят от сетевого оборудования и средств его обслуживания.

Выделяют следующие виды платформ [6]:

- Настольная платформа – однопользовательская или для небольшой группы, в которой необязательно используется сервер базы данных;

- Корпоративная платформа – для рабочей группы или компании, в которой почти всегда оперируют с одним или несколькими серверами баз данных;
- Интернет-платформа – для интернет или интернет приложений

В традиционном понимании платформа—это комплекс аппаратных и программных средств, на котором функционирует программное обеспечение пользователя ЭВМ.

Программно-аппаратная платформа - комплекс, состоящий из аппаратуры(чипсеты,процессоры), а также конфигурации запускаемых на них программ [5].

Программная платформа, или концепция ”программного обеспечения”, вошла в жизнь с развитием компьютерной индустрии. Без программного обеспечения компьютер - это просто электронное устройство, не управляемое, поэтому оно бесполезно. В соответствии с функциями, выполняемыми программным обеспечением, его можно разделить на 2 большие группы: системное и прикладное программное обеспечение.

Системное программное обеспечение - это аппаратно-программная оболочка, предназначенная для отделения других программ от непосредственного взаимодействия с оборудованием и организации процессов обработки информации в компьютере. Прикладное программное обеспечение предназначено для решения конкретных задач пользователя. Системное программное обеспечение включает в себя различные типы программ, такие как операционная система, различные сервисные инструменты, инструментальные средства (система управления базами данных, программирование, оболочка экспертной системы), которые функционально дополняют функции операционной системы.

3.2. Программно-аппаратная платформа для работы с роботами

Данная программно-аппаратная платформа разрабатывается для взаимодействия роботами, находящимися в другом городе. Для этого человеку будет достаточно открыть приложение и выбрать необходимую вкладку, где будут предоставлены разные виды роботов и функции. Общая схема платформы представлена на рисунке 2.



Рис. 2: Общая схема платформы

3.3. Требования к внешнему(веб) серверу

Внешний сервер должен содержать следующие элементы:

- клиента внутреннего разработчика(веб-сервер);
- Базу данных, в которой содержатся
 - роли пользователей
 - * гость
 - * разработчик
 - * робот
 - * клиент

– сценарии работ

Клиент внутреннего разработчика должен выполнять следующие функции:

- Авторизация. Пользователь вводит логин и пароль, происходит проверка данных и пользователь получает соответствующую роль.
- После авторизации пользователю предоставляются варианты работы с платформой:
- Запуск симулятора. Пользователь может запустить симулятор для тестирования задания с виртуальными роботами:
- Запуск с заданными параметрами
 - Поддержка открытия нескольких сеансов разными пользователями
 - Предоставление выбора возможных роботов для работы
 - Трансляция видео с заданными параметрами

4. Разработка макетов

Так же для ПАП были разработаны макеты страниц:

- главная;
- о лаборатории;
- работа с роботами;
- симулятор;
- личный кабинет.

Разберем каждую из них подробнее.

4.1. Навигационное меню

Навигационное меню находится в правом верхнем углу каждой страницы. Оно содержит пункты:

- главная;
- симулятор;
- о лаборатории;
- работа с роботами;
- личный кабинет.

Навигационное меню изображено на рисунке 3.

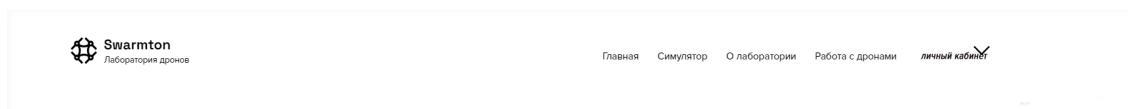


Рис. 3: Навигационная панель

4.2. Главная

На главной странице находится приветствие с краткой информацией о возможностях платформы и ее владельцах. На верхней части страницы есть логотип с названием, а также навигационное меню. Макет главной страницы представлен на рисунке 4

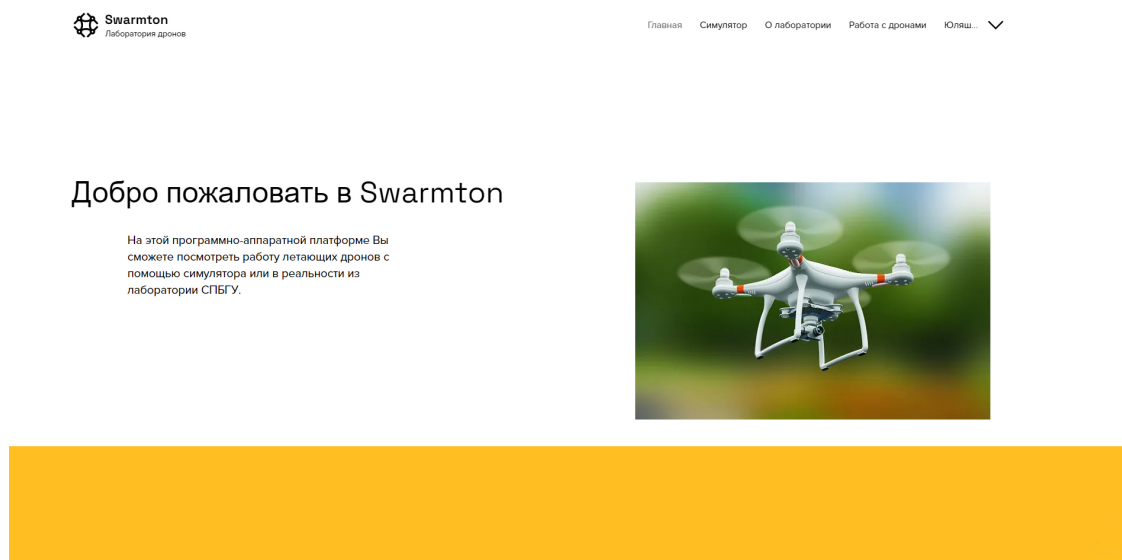


Рис. 4: Главная страница

4.3. О лаборатории

На этой странице содержится подробная информация о лаборатории и её сотрудниках. Информация о участнике лаборатории представлена в виде визитной карточки с фотографией: ФИО сотрудника, его должность, образование и научная степень. Макет страницы "О лаборатории" представлен на рисунке 5

Макет личного кабинета представлен на рисунке 6.

4.4. Симулятор

На странице "Симулятор" представлено рабочее окно для работы с симулятором, на котором можно протестировать и визуализировать работу определенного набора роботов в виртуальной реальности.

Наши сотрудники

Здесь Вы можете ознакомиться с сотрудниками нашей лаборатории



Рис. 5: Страница "О лаборатории"

The screenshot shows the 'Личный кабинет' (Personal Cabinet) page. At the top, there is a yellow header bar with the text 'Имя пользователя' (User Name) and 'роль' (Role). Below the header, there are two tabs: 'Проекты' (Projects) and 'Личный кабинет' (Personal Cabinet). The main content area is divided into three sections: 'Мой аккаунт' (My Account), 'Отображение информации' (Information Display), and 'Аккаунт' (Account). The 'Мой аккаунт' section has buttons for 'Сбросить' (Reset) and 'Обновить' (Update). The 'Отображение информации' section shows a profile card with fields for 'Имя для показа' (Name to display) and 'Название' (Name), and a 'Фото профиля' (Profile Photo) placeholder. The 'Аккаунт' section has fields for 'Имя' (Name), 'Фамилия' (Surname), 'Эл. почта *' (Email), and 'Телефон' (Phone), with buttons for 'Сбросить' (Reset) and 'Обновить' (Update).

Рис. 6: Страница "Личный кабинет"

4.5. Работа с роботами

На странице "Работа с роботами", аналогично странице "Симулятор", представлено рабочее окно, в котором можно будет работать с настоящими роботами.

4.6. Личный кабинет

В личном кабинете имеются текстовые поля для ввода данных имени, фамилии почты и телефона. Так же у каждого человека есть определенная роль, которая определяется администратором платформы:

- гость - не авторизированный пользователь;
- разработчик - человек, который может менять программу платформы;
- клиент- авторизированный пользователь, который может пользоваться ПАП, но изменять исходный платформы не может.

5. Описание сценариев

Взаимодействие пользователя с программно-аппаратной платформой содержит такие этапы как: регистрация и авторизация, работа с симулятором, работа с роботами через приложение, ознакомление с лабораторией.

1. Регистрация и авторизация. Пользователь может зарегистрироваться на платформе, для того чтобы получить роль и работать с симулятором или роботами. Ему будет необходимо для регистрации почту и пароль. В дальнейшем пользователю будет присвоена роль, после чего он сможет работать с платформой.
2. Работа с симулятором. Пользователь заходит на вкладку "Симулятор", после чего запускается программа, с помощью которой пользователь может визуализировать работу дронов или протестировать какой-то алгоритм для роботов.
3. Работа с роботами через приложение. Пользователь заходит на вкладку "Работа с роботами", где запускается рабочее окно с программой, позволяющей выполнять определенные команды с настоящими роботами, находящимися в лаборатории.
4. Ознакомление с лабораторией. Пользователь может перейти на вкладку "О лаборатории", где может ознакомиться с данными о сотрудниках лаборатории, проектах и возможностях платформы.

6. Метод знако-возмущенных сумм

Метод знако-возмущенных сумм позволяет при малом количестве наблюдений построить доверительный интервал с заданной вероятностью. В этой главе рассмотрим модель наблюдения и сам метод.

6.1. Модель наблюдений

В качестве основной модели наблюдения возьмем постановку из статьи [4].

Для функции двух векторных аргументов $f(u, \theta)$: $u \in \mathbb{R}^k$ и $\theta \in \Theta \subseteq \mathbb{R}^d$, $f : \mathbb{R}^k \times \Theta \rightarrow \mathbb{R}$ есть непрерывная производная во всех внутренних точках множества Θ . Для f заранее предполагается, что это достаточно адекватная модель, которая описывает систему с неизвестным параметром $\theta = \theta^*$, Именно этот параметр необходимо оценить. Соответственно для рассмотрения будет актуальна следующая модель наблюдений:

$$y_t = f(u_t, \theta^*) + v_t, \quad (1)$$

где $y_t \in \mathbb{R}$ — входы или наблюдения, $v_t \in \mathbb{R}$ — случайные внешние помехи с симметричным распределением, u_t — выходы или известный план наблюдений, который задается заранее, $\theta^* \in \Theta$ — истинное значение параметра, T — общее количество экспериментов, $t \in 1..T = \{1, 2, \dots, T\}$ — номер эксперимента.

Задача метода по входам u_1, \dots, u_T и выходам y_1, \dots, y_T , полученными в рамках эксперимента, построить доверительное множество $\hat{\Theta}_T \subseteq \Theta$ такое, что при заданном заранее уровне достоверности p выполняется условие: $P(\theta^* \in \hat{\Theta}_T) \geq p$.

6.2. Метод знако-возмущенных сумм

Метод знако-возмущенных сумм [1,3] предполагает выполнение инициализации параметров и формирование доверительного множества. Эти два шага являются основой алгоритма. На первом через эмпирически подобранные параметры M и q формируется доверительная

вероятность p , $\beta_{i,t}$ — случайные величины (СВ) для построения вариантов возмущений системы. На втором шаге используется функция $SPS_indicator(\theta)$, которая проверяет интервалы $\theta \in \mathbb{R}$ на валидность для итогового доверительного интервала.

Теперь рассмотрим подробнее шаги алгоритма:

6.2.1. Инициализация

1. Выбираем q и M — натуральные числа: $M > q > 0$. получаем доверительную вероятность по следующей формуле $p = 1 - q/M$.
2. Производим генерацию $(M - 1)T$ СВ $\beta_{i,t} = \pm 1 : P(\beta_{i,t} = 1) = P(\beta_{i,t} = -1) = 0.5$ для $t = 1..T$ и $i = 1..M$.
3. Применяем функцию $SPS_indicator(\theta)$. Если функция вернула для θ функция вернула 1, то добавляем элемент в доверительное множество, иначе нет.

Значит $\hat{\Theta}_t = \{\theta \in \mathbb{R}^d | SPS_indicator(\theta) = 1\}$.

6.2.2. $SPS_indicator(\theta)$

1. Считаем невязки для полученного θ : $\delta_t(\theta) = y_t - f(u, \theta), t = 1..T$.
2. Считаем сумму всех невязок с весом 1 $H_0(\theta)$, взвешенную сумму со сгенерированными ранее весами $\beta_{i,t}$ всех невязок $H_i(\theta)$ для $i \in 1..M$

$$H_0(\theta) = \sum_{t=1}^T \delta_t(\theta) \quad (2)$$

$$H_i(\theta) = \sum_{t=1}^T \beta_{i,t} \delta_t(\theta) \quad (3)$$

3. Возводим в квадрат значения $H_i(\theta), i \in 0..M$ и сортируем по возрастанию. Пусть $R(\theta)$ — это позиция $H_0^2(\theta)$ в этом упорядоченном множестве.

4. Если $R(\theta) \leq M - q$, то $SPS_indicator(\theta)$, то отдаем 1, в другом случае 0.

Преимущество данного метода заключается в использовании малого количества наблюдений для получения интервала с заданной вероятностью. Поэтому в рамках ВКР будет предложена модификация метода для корректировки местоположения на карте.

7. Заключение

В ходе работы выполнены следующие задачи:

- разработаны требования к веб-серверу;
- разработаны макеты страниц;
- описаны сценарии взаимодействия с веб-сервером;
- изучен метод знако-возмущенных сумм.

В дальнейшем планируется модификация метода знако-возмущенных сумм для корректировки местоположения и апробация его в рамках ПАП.

Список литературы

- [1] О возможности применения метода знако-возмущенных сумм для обработки результатов динамических испытаний / М.В. Волкова, О.Н. Граничин, Г.А. Волков, Ю.В. Петров // Вестник СПбГУ. — 2018. — Т. 63, № 1. — С. 30–40.
- [2] Палушенко М.И. Евстафьев Г.М. Беспилотные летательные аппараты: история, применение, угроза распространения и перспективы развития. — М.: "Права человека", 2005. — С. 125.
- [3] Csáji B., Campi M., Weyer E. Sign-Perturbed Sums: A New System Identification Approach for Constructing Exact Non-Asymptotic Confidence Regions in Linear Regression Models // [IEEE Transactions on Signal Processing](#). — 2015. — Vol. 63, no. 1. — P. 169–181.
- [4] Sign-perturbed sums approach for data treatment of dynamic fracture tests / Marina Volkova, G. Volkov, O. Granichin, Y. Petrov // 2017 IEEE 56th Annual Conference on Decision and Control (CDC). — 2017. — P. 1652–1656.
- [5] Алексеевич Шашин Михаил. Аппаратно-программная платформа // Образовательный портал «Справочник». — URL: https://spravochnick.ru/bazy_dannyh/dopolnitelnye_voprosy_primeneniya_baz_dannyh/apparatno-programmnaya_platforma.
- [6] Альбекова Замира Мухамедалиевна Сороченко Виктория Романовна Разумовская Людмила Евгеньевна. Понятие платформы в информационных технологиях // Фундаментальные и прикладные научные исследования: актуальные вопросы, достижения и инновации: в 2 ч.. Том Часть 1. — 2021. — P. 158–160.
- [7] П. Вострикова З. Программирование на языке ассемблера ЕС ЭВМ. — Москва : Наука, 1981.