

# Разработка веб-приложения для мониторинга параметров работы манипуляционных механизмов

М. О. Чайкин

**Аннотация** — Разработка манипуляционных механизмов с различной кинематической структурой напрямую связана с созданием системы управления, которая состоит из аппаратного и стратегического уровней. Аппаратный уровень системы управления объединяет электронно-компонентную базу (приводы, датчики обратной связи и т.д.) исполнительного механизма с целью обеспечения заданного движения кинематических цепей. Стратегический уровень предназначен для упрощения взаимодействия человека-оператора и механизма. В данной работе описывается создание RESTful API веб-приложения в качестве стратегического уровня управления для мониторинга параметров работы механизма, а также организация взаимодействия между уровнями управления посредством веб-сокетов. Расчёт управляющих сигналов для приводов механизма осуществлялся на микроконтроллере семейства STM32 с последующей отправкой экспериментальных данных на микроконтроллер ESP8266, на котором был реализован веб-сокет сервер. Для хранения получаемых данных была спроектирована база данных PostgreSQL, а также API для управления доступом к информации с использованием языка программирования Golang. Для прототипирования интерфейса клиентского приложения был выбран реактивный фреймворк Vue JS. В качестве результата работы представлены примеры разработанного функционала по организации сеансов наблюдений, сохранения их истории, а также редактированию конфигурации доступных манипуляционных механизмов.

**Ключевые слова** — управление манипуляционными механизмами, контроль параметров манипуляционных механизмов, веб-сокеты, Go, Vue Js, PostgreSQL, RESTful API

## I. ВВЕДЕНИЕ

Разработка и исследование манипуляционных механизмов неразрывно связана с синтезом системы управления, что включает в себя необходимость контроля показателей работы механизма в процессе испытаний или прохождения по заданной траектории при выполнении поставленной задачи. В общем случае управление манипуляционным механизмом подразделяют на аппаратный и стратегический уровень [7]. В аппаратный уровень управления входят микроконтроллер, приводы, датчики обратной связи и другое оборудование, необходимое для приведения в движение кинематических пар механизма.

Стратегический уровень управления необходим для создания системы взаимодействия оператора и манипуляционного механизма [3, 4] с целью обеспечения удобного интерфейса в области передачи и обработки информации.

В качестве стратегического уровня управления предлагается создать веб-приложение для унификации сбора, обработки, хранения и визуализации параметров работы манипуляционных механизмов. Данное программное обеспечение предполагается использовать в процессе синтеза управления механизмов последовательной и параллельной кинематики, в том числе для механизмов совместного относительного манипулирования (MCOM), которые сочетают в себе манипуляционные возможности нескольких манипуляторов, достигая при этом высокую производительность, точность позиционирования и достаточное число степеней свободы для ориентации выходного звена в различных плоскостях относительно базовой системы координат.

## II. ОПРЕДЕЛЕНИЕ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

Основной задачей разрабатываемого веб-приложения является мониторинг параметров манипуляционных механизмов в зависимости от их электронно-компонентной базы. Следовательно главные функциональные требования можно сформулировать следующим образом:

1. Организация командной работы по разработке и исследованию механизма посредством авторизации в личном кабинете или регистрации нового пользователя;
2. Взаимодействие с манипуляционными механизмами:
  - a. Просмотр доступных механизмов для контроля;
  - b. Редактирование описания механизма;
  - c. Редактирование электронно-компонентной базы механизма;
3. Добавление новых и удаление старых параметров для контроля.
4. Контроль параметров робототехнического механизма при проведении сеанса испытаний.
5. Проведение дополнительных наблюдений за параметрами, не принадлежащих исполнительному механизму.

Статья получена 23 января 2023.

Чайкин Максим Олегович, Калужский филиал МГТУ им. Н.Э. Баумана (chaykin.maksim@yandex.ru)

6. Добавление, редактирование и удаление параметров для наблюдения.
7. Добавление и редактирования описания доступных приводов и датчиков обратной связи в электронно-компонентной базе.
8. Сохранение истории наблюдений с возможностью её повторной визуализации.

Диаграмма возможного использования веб-приложения показана на рисунке 1.



Рис. 1. Диаграмма возможного использования

### III. ОПИСАНИЕ РЕШЕНИЯ

#### A. Архитектура аппаратного уровня управления

Для получения и передачи данных между аппаратным и стратегическим уровнями используется протокол связи WebSocket, поскольку он позволяет запрашивать или отправлять данные тогда, когда они станут доступны при этом поддерживая активным единое открытое соединение [8]. В результате веб-сокеты обеспечивают передачу данных в режиме реального времени, что является критически важным в рамках решения задачи мониторинга параметров манипуляционных механизмов. Для разворачивания сервера WebSocket был использован микроконтроллер ESP8266, который имеет интерфейс WiFi.

Опрос датчиков обратной связи и расчет сигнала управления производится на микроконтроллере STM32. Затем рассчитанный сигнал управления передаётся на драйвера приводов, которые приводят в движение кинематические звенья манипуляционного механизма. Тип используемых приводов определяется конструктивными особенностями механизма и требованиями на скорость вращения вала привода, максимальный крутящий момент, пусковой ток и т.д. Передача данных между STM32 и ESP8266 осуществляется при помощи UART.

Обобщённая архитектура аппаратного уровня управления представлена на рисунке 2.

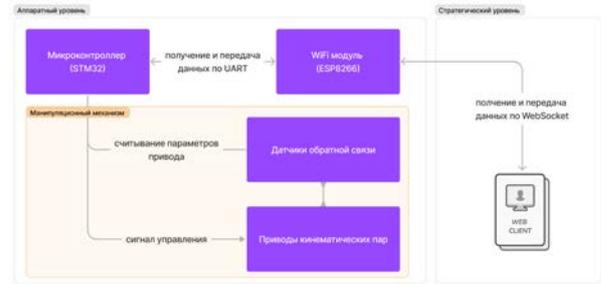


Рис. 2. Обобщённая архитектура управления манипуляционным механизмом

#### B. Архитектура веб-приложения

В качестве решения поставленной задачи было создано RESTful API приложение [2], которое обеспечивает взаимодействие между клиентом и сервером базы данных благодаря единому интерфейсу API (Рис. 3). Данное решение позволяет разделить логику клиента и сервера, что в свою очередь обеспечивает:

- возможность масштабирования за счет повышения производительности взаимодействия клиента и сервера;
- независимую разработку клиента и сервера благодаря разделению функций приложения на уровни;
- независимость от используемых технологий для написания клиента или серверной части приложений.

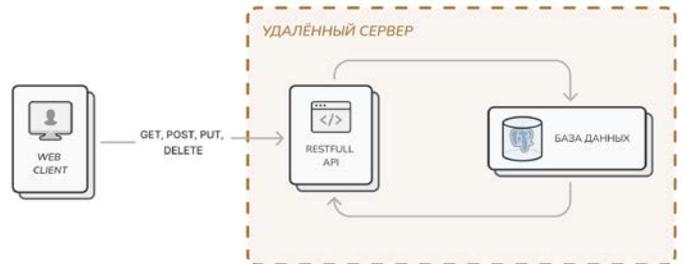


Рис. 3. Архитектура разрабатываемого приложения

В качестве фреймворка для разработки клиента был выбран Vue JS [5], поскольку данное решение позволяет:

- Создавать реактивные приложения, которые автоматически обновляют своё представление при изменении отображаемых данных;
- Разрабатывать отдельные компоненты для решения определённых функциональных задач с возможностью последующего повторного использования, что существенно снижает объем реализуемого кода;
- Гибко настраивать обработку действий пользователей при взаимодействии с элементами верстки;
- Хранить и мутировать данные в централизованном хранилище Vuex, что обеспечивает синхронизацию состояния между компонентами;
- Настраивать маршрутизацию и права доступа к параметрам маршрута;
- Удобно взаимодействовать с интерфейсами API;
- Использовать дополнительные библиотеки UI-

компонентов, которые упрощают прототипирование интерфейсов.

Для хранения данных была использована реляционная база данных PostgreSQL [6], т.к. данное решение

является бесплатным и высокопроизводительным. ERD модель разработанной базы данных представлена на рисунке 4.

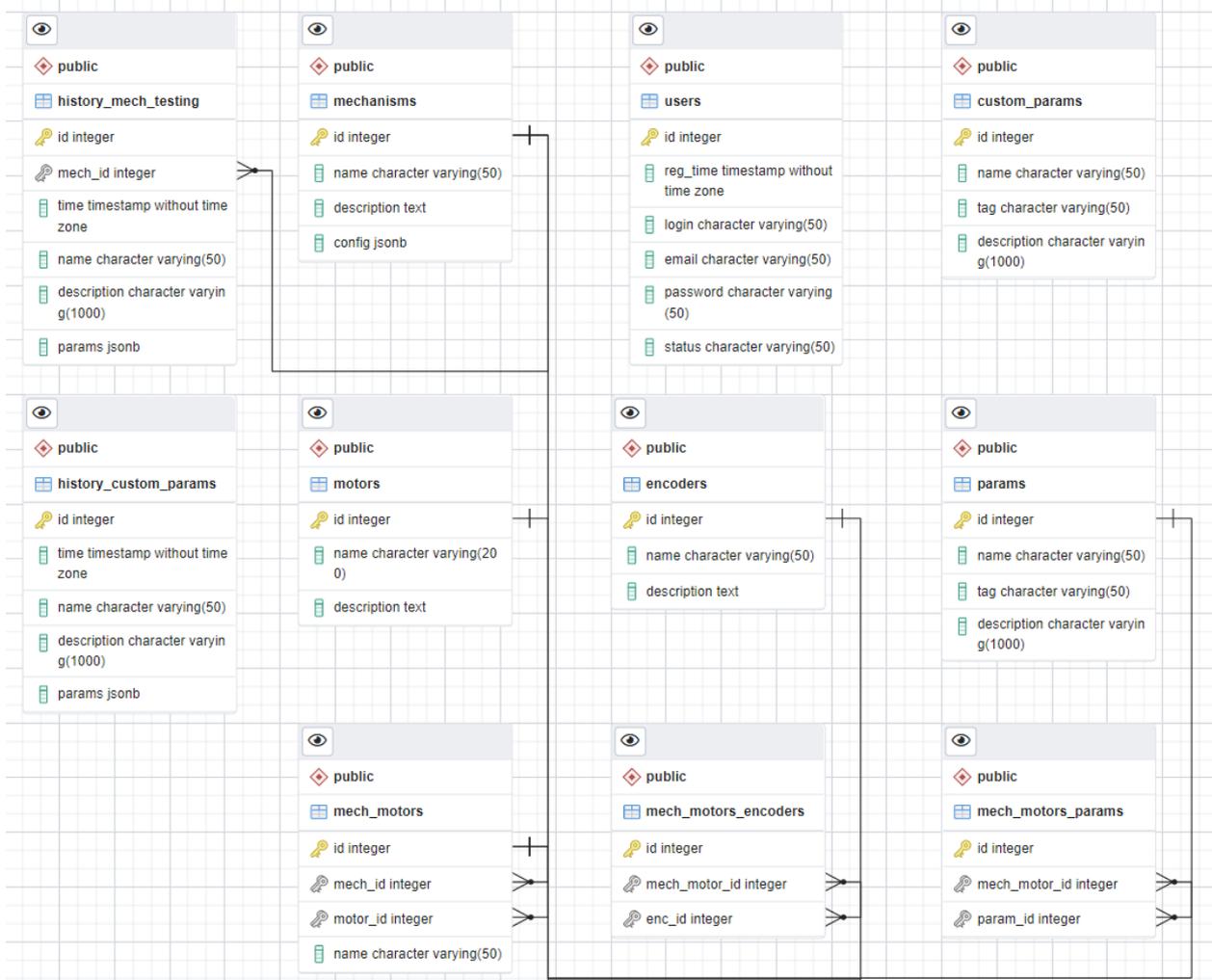


Рис. 4. ERD модель базы данных

Для создания REST API использовался язык программирования Go с фреймворком GIN [1]. Go широко используется для создания web-сервисов благодаря простоте своих конструкций, что упрощает обслуживание кода и обеспечивает его чистоту.

параметров

В обозревателе механизмов пользователь может редактировать описание манипуляционного механизма, а также его кинематическую структуру (Рис. 6).

#### IV. ДЕМОНСТРАЦИЯ РАЗРАБОТАННОГО ВЕБ-ПРИЛОЖЕНИЯ

##### A. Обзорщик механизмов

В разделе «Механизмы» разработанного приложения возможно посмотреть список доступных механизмов (Рис. 5).



Рис. 5. Механизмы, доступные для мониторинга

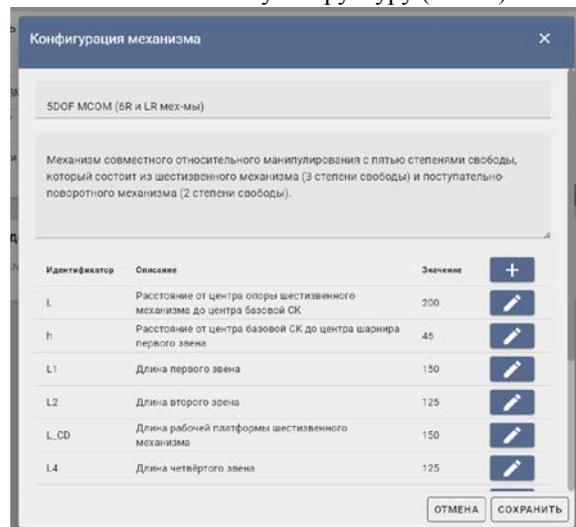


Рис. 6. Настройки описания и конфигурации механизма. Параметры, доступные для мониторинга механизма

зависят от его электронно-компонентной базы. Поэтому важно иметь возможность редактировать используемые компоненты и исходя из этого удалять или добавлять параметры для контроля (Рис. 7).

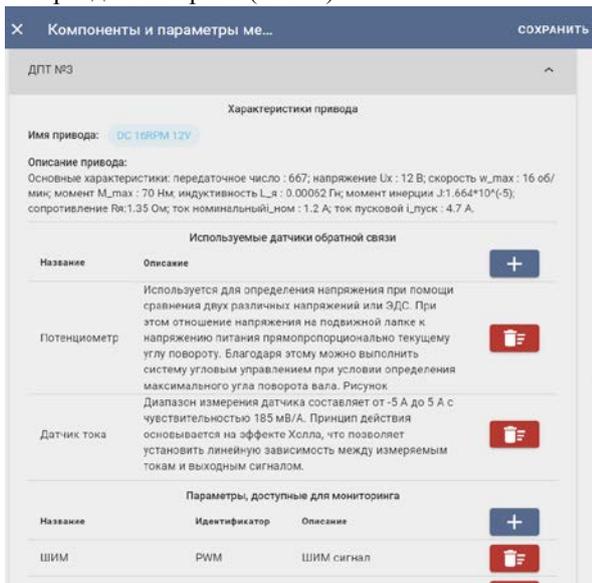


Рис. 7. Редактирование электронно-компонентной базы механизма и параметров для контроля

#### В. Проведение дополнительных наблюдений

Проведение дополнительных наблюдений возможно в разделе «Дополнительные наблюдения», в котором сначала предлагается выбрать параметры для мониторинга (Рис. 8).

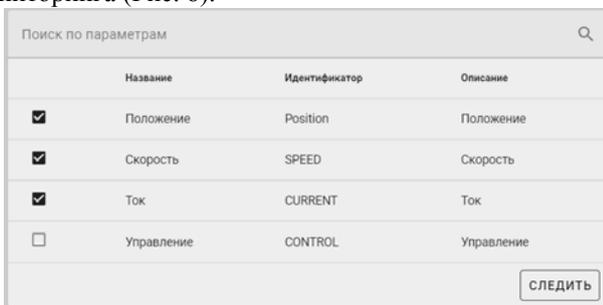


Рис. 8. Выбор дополнительных параметров для контроля

После нажатия на кнопку «Следить» пользователю отображаются графики параметров, а также таблица их значений (Рис. 9).

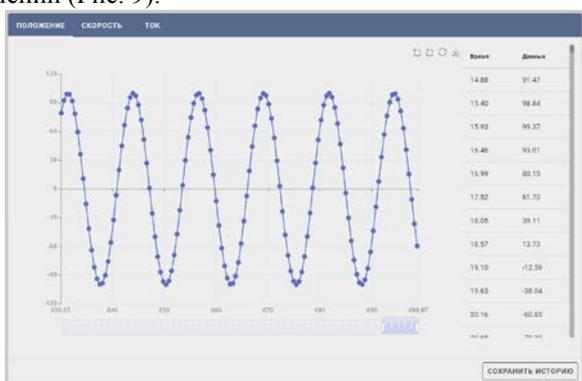


Рис. 9. График и таблица значений параметра

Формат сообщения, считываемого по веб-сокету имеет следующий вид:  $c, tag, t, value$ , где  $c$  – символ 'с', характеризующий начало нового сообщения;  $tag$  –

идентификатор параметра;  $t$  – время считывания параметра;  $value$  – значение параметра.

Также сеанс наблюдений можно сохранить в историю, указав название и краткое описание (Рис. 10).

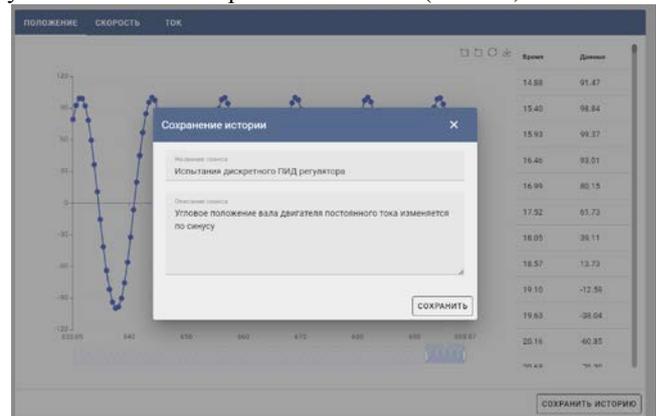


Рис. 10. Сохранение сеанса наблюдений

#### С. Сеанс наблюдений за механизмом

Для проведения сеанса наблюдений необходимо выбрать интересующий механизм, а затем для каждого из приводов указать параметры для наблюдения (Рис. 11).

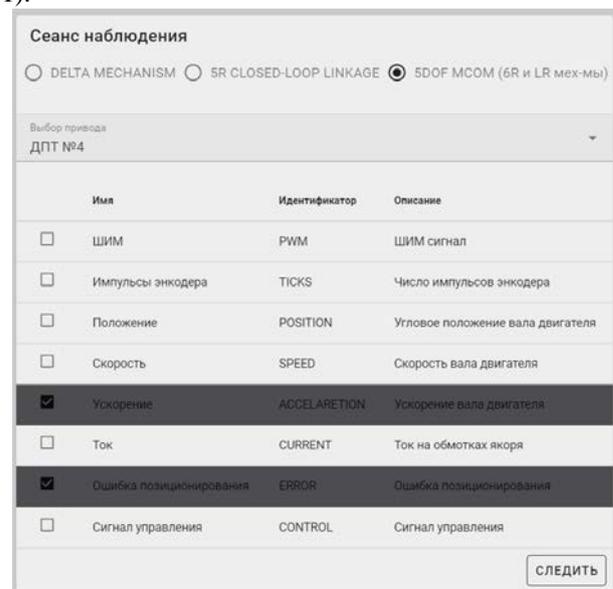


Рис. 11. Выбор параметров для наблюдений при сеансе испытаний механизма

После нажатия на кнопку «следить» откроется окно с графиками параметров, изменяющихся по мере обработки сообщений, получаемых по веб-сокету. Сообщение для веб-сокета формируется следующим образом:  $id_{мех.}, id_{прив.}, tag, t, value$ , где  $id_{мех.}$  – идентификатор механизма;  $id_{прив.}$  – идентификатор привода механизма;  $tag$  – идентификатор параметра;  $t$  – время считывания параметра;  $value$  – значение параметра.

#### Д. Менеджер параметров

В разделе «Менеджер параметров» возможно просмотреть все параметры, которые доступны в системе (Рис. 12).

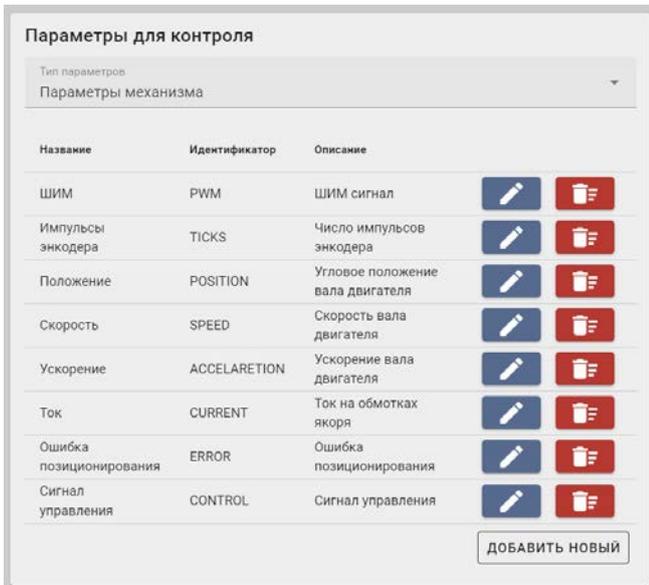


Рис. 12. Обзорщик доступных параметров для контроля

Также пользователю предоставляется возможность добавления, редактирования и удаления параметров.

#### Е. Электронно-компонентная база

В разделе «Компонентная база» показаны все доступные приводы и датчики обратной связи (Рис. 13).

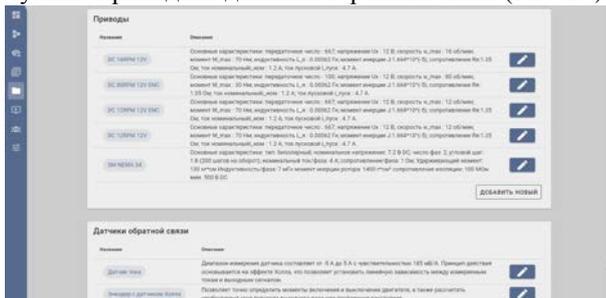


Рис. 13. Электронно-компонентная база

Пользователь может добавлять новые компоненты, а также редактировать старые (Рис. 14).

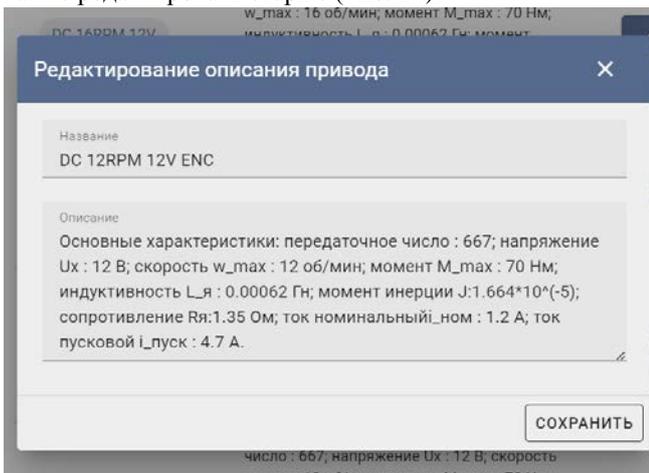


Рис. 14. Редактирование описания привода

#### Г. Электронно-компонентная база

В разделе «История наблюдений» возможно посмотреть все сохранённые сеансы испытаний, а также выбрать историю для отображения, редактирования или удаления (Рис. 15).

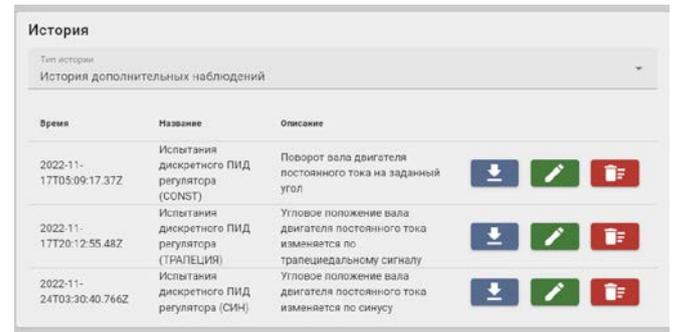


Рис. 15. Доступные сеансы испытаний

## V. ЗАКЛЮЧЕНИЕ

В работе рассмотрены принципы разработки программно-алгоритмического комплекса управления манипуляционными механизмами с последовательной и параллельной кинематикой. Была представлена обобщённая схема аппаратного уровня системы управления манипуляционного механизма, а также предложен возможный вариант создания RESTful API приложения для реализации стратегического уровня управления. Клиент был создан при помощи фреймворка Vue JS, для создания REST API сервера использовался язык программирования Go, а в качестве базы данных была выбрана PostgreSQL. Взаимодействие между уровнями управления осуществляется при помощи передачи сообщений посредством веб-сокета.

Результующее программное обеспечение позволяет организовывать командную работу для мониторинга параметров работы манипуляционных механизмов, редактировать их описание, электронно-компонентную базу и параметры, доступные для контроля, а также сохранять и воспроизводить истории наблюдений.

## БИБЛИОГРАФИЯ

- [1] Andrawos, Mina, and Martin Helmich. Cloud Native Programming with Golang: Develop microservice-based high performance web apps for the cloud with Go. Packt Publishing Ltd, 2017.
- [2] Biehl, Matthias. RESTful Api Design. Vol. 3. API-University Press, 2016.
- [3] Kaczmarek, W., Lotys, B., Borys, S., Laskowski, D., & Lubkowski, P. (2021). Controlling an Industrial Robot Using a Graphic Tablet in Offline and Online Mode. Sensors (Basel, Switzerland), 21(7), 2439. <https://doi.org/10.3390/s21072439>
- [4] Kozlov S.V., Problematic aspects of integrating a complete group of processes in the lifecycle of robotic control systems, Procedia Computer Science, Volume 186, 2021, pages 210-216.
- [5] Li, Nian, and Bo Zhang. "The Research on Single Page Application Front-end development Based on Vue." Journal of Physics: Conference Series. Vol. 1883. No. 1. IOP Publishing, 2021.
- [6] Obe, Regina O., and Leo S. Hsu. PostgreSQL: up and running: a practical guide to the advanced open source database. " O'Reilly Media, Inc.", 2017.
- [7] Romanov A.M. A review on control systems hardware and software for robots of various scale and purpose. Part 1. Industrial robotics. Russian Technological Journal. 2019;7(5):30-46. (In Russ.) <https://doi.org/10.32362/2500-316X-2019-7-5-30-46>
- [8] Shiyong Xiong, Xue Wang, Zhengpeng Lan. Model Research of Visual Report Components. Procedia Computer Science. Volume 208, 2022, Pages 478-485, ISSN 1877-0509. <https://doi.org/10.1016/j.procs.2022.10.066>

# Development of a web application for monitoring the parameters of manipulation mechanisms

Maxim O. Chaikin

**Abstract** — The development of manipulation mechanisms with different kinematic structures is directly related to the creation of a control system that consists of hardware and strategic levels. The hardware level of the control system combines the electronic component base (drives, sensors, etc.) of the actuator in order to ensure a given movement of kinematic circuits. The strategic level is designed to simplify the interaction of the human operator and the mechanism. This paper describes the creation of a RESTful API of a web application as a strategic management layer for monitoring the parameters of the mechanism, as well as the organization of interaction between management levels via web sockets. The calculation of control signals for the mechanism drives was carried out on a microcontroller of the STM32 family, followed by sending experimental data to the ESP8266 microcontroller, on which a web socket server was implemented. To store the received data, a PostgreSQL database was designed, as well as an API for managing access to information using the Golang programming language. The Vue JS reactive framework was chosen for prototyping the client application interface. As a result of the work, examples of the developed functionality for organizing observation sessions, saving their history, as well as editing the configuration of available manipulation mechanisms are presented.

**Keywords** — manipulation mechanism control, manipulation mechanism parameter control, web sockets, Go, Vue Js, PostgreSQL, RESTful API

## REFERENCES

- [1] Andrawos, Mina, and Martin Helmich. Cloud Native Programming with Golang: Develop microservice-based high performance web apps for the cloud with Go. Packt Publishing Ltd, 2017.
- [2] Biehl, Matthias. RESTful Api Design. Vol. 3. API-University Press, 2016.
- [3] Kaczmarek, W., Lotys, B., Borys, S., Laskowski, D., & Lubkowski, P. (2021). Controlling an Industrial Robot Using a Graphic Tablet in Offline and Online Mode. *Sensors (Basel, Switzerland)*, 21(7), 2439. <https://doi.org/10.3390/s21072439>
- [4] Kozlov S.V., Problematic aspects of integrating a complete group of processes in the lifecycle of robotic control systems, *Procedia Computer Science*, Volume 186, 2021, pages 210-216.
- [5] Li, Nian, and Bo Zhang. "The Research on Single Page Application Front-end development Based on Vue." *Journal of Physics: Conference Series*. Vol. 1883. No. 1. IOP Publishing, 2021.
- [6] Obe, Regina O., and Leo S. Hsu. *PostgreSQL: up and running: a practical guide to the advanced open source database.* " O'Reilly Media, Inc.", 2017.
- [7] Romanov A.M. A review on control systems hardware and software for robots of various scale and purpose. Part 1. *Industrial robotics. Russian Technological Journal*. 2019;7(5):30-46. (In Russ.) <https://doi.org/10.32362/2500-316X-2019-7-5-30-46>
- [8] Shiyong Xiong, Xue Wang, Zhengpeng Lan. Model Research of Visual Report Components. *Procedia Computer Science*. Volume 208, 2022, Pages 478-485, ISSN 1877-0509. <https://doi.org/10.1016/j.procs.2022.10.066>