

## ДОДАТОК А

## Слайди презентації

**Магістерська атестаційна робота****ДОСЛІДЖЕННЯ МЕТОДІВ ТА  
ПРОГРАМНИХ СИСТЕМ КУРСОВОЇ  
СТАБІЛІЗАЦІЇ РУХУ АВТОМОБІЛЯ**

Виконав:

ст. гр. ПЗСм-18-1  
Гавриш О.В.

Керівник:

проф. Білоус Н.В.

Консультант:

к.т.н., доц. Назаров О.С.

1

**Актуальність**

- При використанні дистанційно керованих автомобілів виникають позаштатні ситуації
- на практиці виникають певні труднощі, що залежать від впливу оточуючого середовища, поверхні пересування, кінематичних та динамічних характеристик шасі, геометричної трансформації шин, різного коефіцієнта щеплення, різного роду нерівності тощо.
- існуючі системи стабілізації не беруть до уваги поточну швидкість пересування автомобіля.

2

## Мета роботи

- Об'єкт дослідження – системи стабілізації руху колісних роботів.
- Предмет дослідження – ефективність та принцип дії систем стабілізації.
- Мета роботи – моделювання процесу руху та стабілізації траєкторії пересування колісних роботів. Створення практичної програмної системи.
- Методи дослідження - математичне моделювання руху автомобіля, концептуальне моделювання даних, використання об'єктно-орієнтованого підходу до створення програмного додатку.

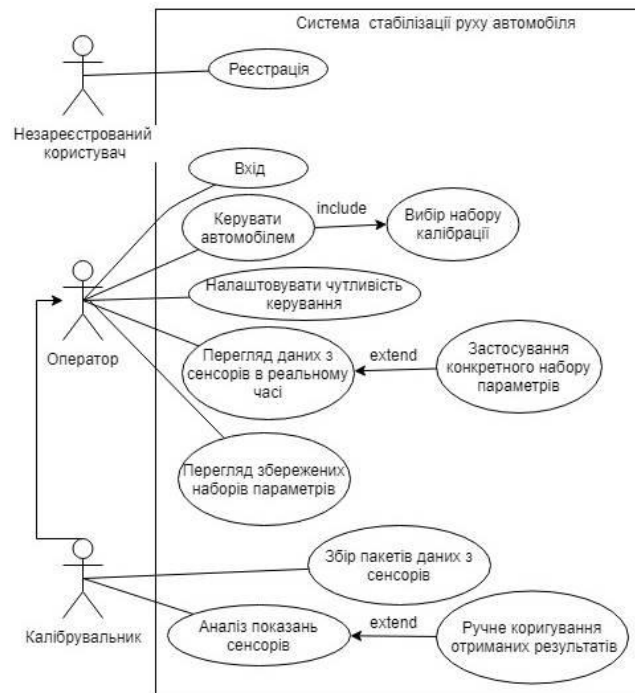
3

## Постановка задачі

- провести аналіз проблемної галузі систем стабілізації;
- розробити математичну модель руху дистанційно керованого автомобіля з урахуванням впливу навколишнього середовища;
- на основі створеної моделі розробити алгоритми стабілізації та коригування вхідних даних;
- розробити схему бази даних для зберігання даних з датчиків та готові набори параметрів;
- виконати програмну реалізацію системи стабілізації руху автомобіля.

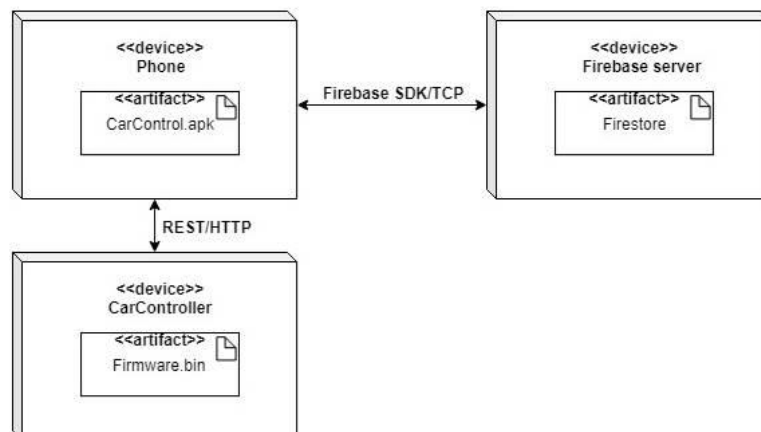
4

## Діаграма прецедентів



5

## Діаграма розгортання



6

## Вибір засобів розробки

### □ Android SDK

- ✓ Зручний та гнучкий редактор розмітки;
- ✓ Широкий набір підтримуваних пристроїв;
- ✓ Потенційна кросплатформеність написаного коду;
- ✓ Доступ до низькорівневих системних компонентів;
- ✓ Підтримка сучасних архітектур.

### □ ESP 32 IDF

- ✓ Вбудована підтримка роботи з Wi-Fi модулем;
- ✓ Висока швидкодія;
- ✓ Надійність роботи;
- ✓ Простота у використанні;
- ✓ Підтримка HTTP стеку.

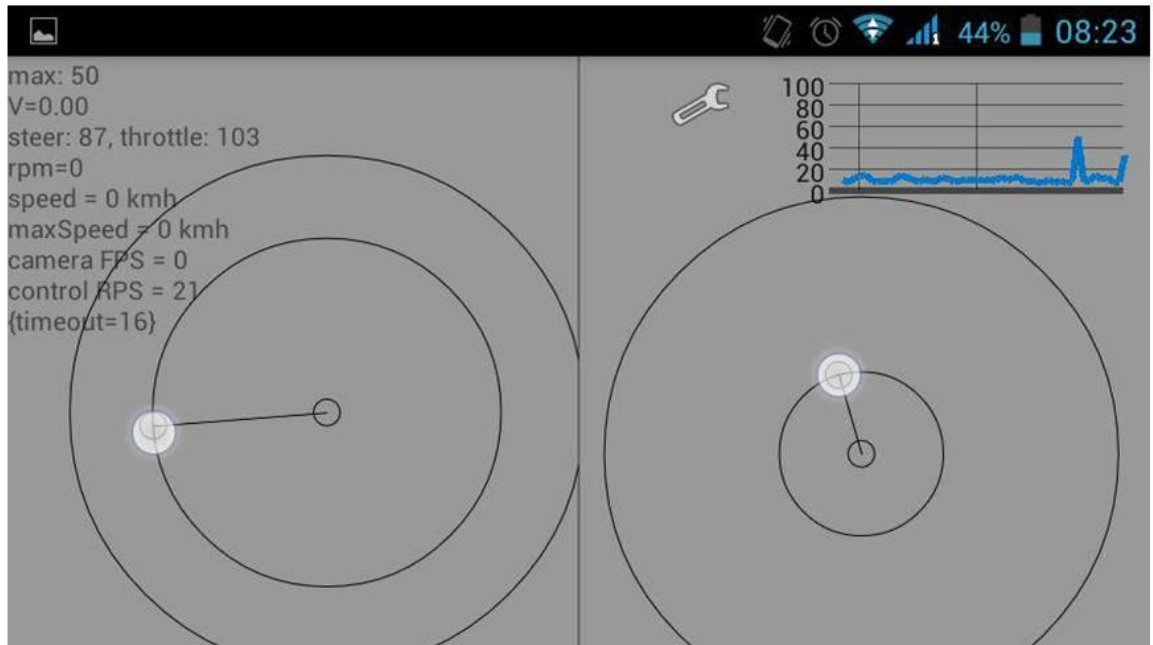
7

## Інтерфейс авторизації



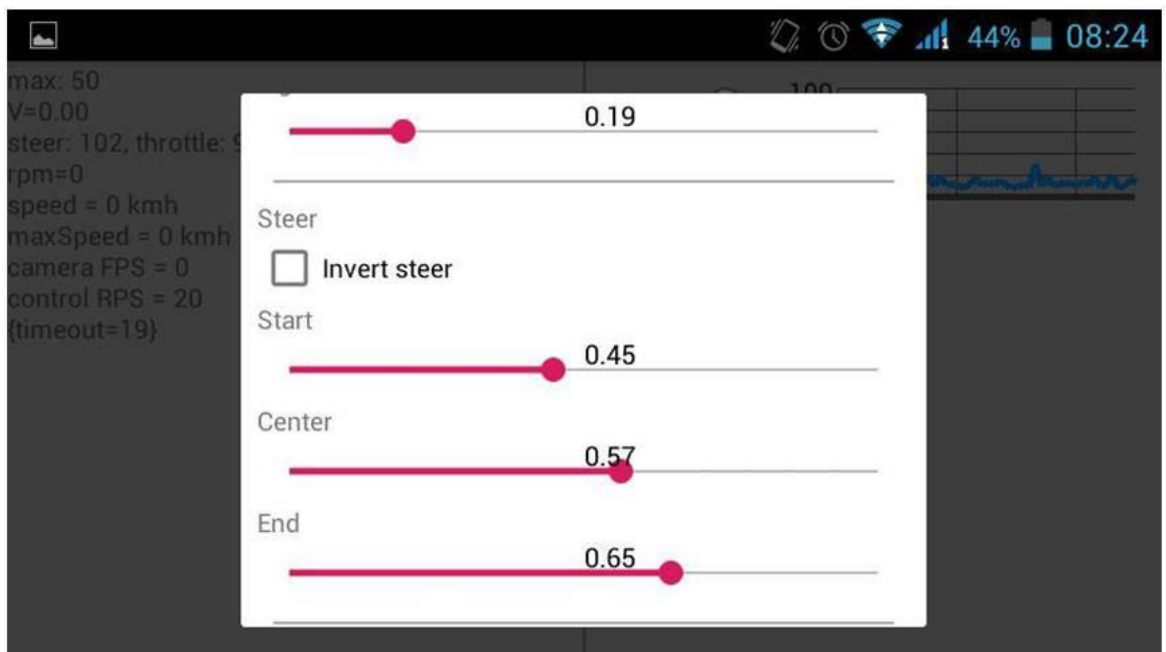
8

## Інтерфейс керування автомобілем



9

## Інтерфейс налаштування системи



10

## **Висновки**

- досліджені основні методи і алгоритми систем стабілізації руху автомобіля;
- обрані методи, які будуть використовуватися під час програмної реалізації системи;
- створено математичну модель руху колісного транспортного засобу з урахуванням зовнішніх чинників;
- розроблено алгоритми стабілізації та визначення оптимальних контрольних параметрів;
- розроблена UML-модель для візуалізації функціоналу системи з точки зору різних категорій користувачів;
- опублікувана стаття «Дослідження методів та програмних систем курсової стабілізації руху автомобіля» в матеріалах IV міжнародної науково-практичної конференції в Ліверпулі «Scientific Achievements of Modern Society»

ДОДАТОК Б

Наукові публікації

**SCIENTIFIC  
ACHIEVEMENTS OF  
MODERN SOCIETY**

Abstracts of IV International Scientific and Practical Conference  
Liverpool, United Kingdom  
4-6 December 2019

**Liverpool, United Kingdom  
2019**

УДК 62-529

**ДОСЛІДЖЕННЯ МЕТОДІВ ТА ПРОГРАМНИХ СИСТЕМ КУРСОВОЇ  
СТАБІЛІЗАЦІЇ РУХУ АВТОМОБІЛЯ**

**Назаров Олексій Сергійович**

к.т.н., доцент

**Гавриш Олег Владиславович**

Магістрант

Харківський національний університет радіоелектроніки

м. Харків, Україна

**Анотація:** В роботі наведено результати дослідження використання методів теорії прийняття рішень в системі руху колісних транспортних засобів. Запропоновано використання методів компенсаційної стабілізації на основі даних з гіроскопа, акселерометра, датчиків кутової швидкості для мінімізації відхилення колісних транспортних засобів від бажаної траєкторії руху.

**Ключові слова:** стабілізація, робот, автомобіль, гіроскоп, акселерометр, компенсація, інтернет речей, мобільний додаток.

**Постановка проблеми**

В наш час, з розвитком технічного прогресу, все більше галузей промисловості, медицини, військові та навіть цивільні люди в побуті використовують різноманітні пристрої для автоматизації чи полегшення певних процесів та завдань. Одним із напрямків такої автоматизації являються безпілотні транспортні машини, які можуть бути використані для доставки вантажів, тактичної розвідки, спортивних змагань, розміщення спеціалізованих пристроїв тощо. Найпростішими та доступними у використанні являються саме наземні. На практиці виникають певні труднощі, що залежать від впливу оточуючого середовища, поверхні пересування, кінематичних та динамічних характеристик шасі, геометричної трансформації шин, різного коефіцієнта щеплення, різного

роду нерівності тощо. Це досить сильно ускладнює прогнозованість керування такими засобами, потребує деяких ручних коригувань з боку оператора, знижує продуктивність то точність роботи такого пристрою [1].

#### **Аналіз наукових підходів до вирішення проблеми**

Стабілізація, нехай то автоматизована, чи ручна, залежить від типу середовища, поверхні, характеристик шасі, габаритів та ваги вантажу який перевозиться, обладнання, тощо. Зазвичай системи для стабілізації руху радіокерованого автомобіля працюють шляхом коригування поворотного приводу. Система використовує гіроскоп для визначення поточного напрямку руху та фізичного положення автомобіля в просторі. На основі цих даних програмна частина виконує коригування вхідного параметру повороту передніх коліс відповідно до отриманих результатів. Цей підхід є дієвим, проте він не враховує факт різниці кутової швидкості різних колісних осей [2].

#### **Постановка задачі**

Отже, на основі моделей та методів теорії керування колісними транспортними засобами необхідно розробити математичну модель стабілізації руху, яка допоможе вирішити задачі підтримки пересування по заданій траєкторії, збільшить ефективність виконуваної роботи. Також необхідно програмно реалізувати розроблену модель та алгоритми вирішення поставлених задач в складі програмно-апаратної системи для дистанційного керування автомобілем, провести дослідження розробленої моделі та методів.

#### **Математична модель системи стабілізації автомобіля**

Поворотність автомобіля – це властивість збільшувати, зберігати чи зменшувати кривизну траєкторії встановленого руху при збільшенні бокового прискорення. Розрізняють надмірну, недостатню та нейтральну поворотність [3]. Дані типи можна математично описати на основі кута уведення передньої осі  $\delta_F$ , задньої  $\delta_R$  та бокового прискорення  $j_y$ .

Надмірна поворотність характеризує автомобіль, який збільшує кривизну траєкторії зі збільшенням бокового прискорення (рис. 1).

$$\frac{\partial(\delta_F - \delta_R)}{\partial j_Y} < 0.$$

**Рис. 1 – нерівність для надмірної поворотності**

Нейтральна поворотність виникає коли автомобіль не змінює кривизну своєї траєкторії руху при збільшенні бокового прискорення (рис. 2).

$$\frac{\partial(\delta_F - \delta_R)}{\partial j_Y} = 0;$$

**Рис. 2 – рівняння для нейтральної поворотності**

Надмірна поворотність характеризує автомобіль, який зменшує кривизну траєкторії зі збільшенням бокового прискорення (рис. 3).

$$\frac{\partial(\delta_F - \delta_R)}{\partial j_Y} > 0;$$

**Рис. 3 – нерівність для недостатньої поворотності**

Визначивши поточний стан руху автомобіля, відносна швидкість  $v_r$  та його положення на основі даних з гіроскопа  $G_y$ , можна вирахувати компенсаційний кут повороту коліс  $S_c$  за наступною формулою (рис. 4).

$$S_c = \frac{G_y * k}{v_r}$$

**Рис. 4 – формула кута компенсації**

Для налаштування та калібрації системи під реальні умови використання, введено коефіцієнт  $k$ , що дозволяє змінювати інтенсивність компенсації. Також слід відзначити, що чим більша лінійна швидкість пересування, тим меншим має бути кут коригування.

Розроблена задача про коригування траєкторії пересування автомобіля дозволить оптимальним чином забезпечити ефективність та стабільність пересування колісних транспортних засобів.

**Висновки та перспективи**

В ході роботи були досліджені основні методи і алгоритми систем стабілізації руху автомобіля, можливість їх застосування при розробці практичної системи керування та стабілізації.

Отримана програмна система дозволяє дистанційно керувати колісними транспортними засобами та забезпечувати достатній рівень стабільності ходу завдяки реалізованому алгоритму стабілізації. В подальшому планується вдосконалення алгоритму та розширенню набору датчиків, що слугують вхідними даними для системи. Це дозволить зробити керування ще більш точним та ефективним.

**СПИСОК ЛІТЕРАТУРИ**

4. Csaba Szepesvári. Algorithms for Reinforcement Learning. США: M&C Publishers. 2009. 98 с.
5. Таха Хемді. Дослідження операцій. Москва: Вільямс, 2016. 912 с.
6. Лата В.Н. Основы моделирования управляемого движения автомобиля. Тольятті: ТГУ, 2012. 69 с.

## ДОДАТОК В

## Лістинг програмного коду

```

class CarSenderImpl(preferences: ControlPreferences) : CarSender {

    private val client = OkHttpClient.Builder()
        .callTimeout(preferences.requestTimeout, TimeUnit.MILLISECONDS)
        .addInterceptor(HttpLoggingInterceptor().apply {
            level = HttpLoggingInterceptor.Level.BODY
        })
        .build()

    override suspend fun send(params: CarParams): CarResponse {
        val request = Request.Builder()
            .url(buildUrl(params))
            .get()
            .build()

        val response = client.newCall(request).execute()

        if (response.isSuccessful) {
            val responseString = response.body()!!.string()
            val voltageRaw = runCatching { parseVoltageRaw(responseString) }.getOrNull() ?:

Of
            val rpm = try {
                parseRpm(responseString)
            } catch (e: Exception) {
                0
            }
            val time = response.receivedResponseAtMillis() - response.sentRequestAtMillis()
            return CarResponse(voltageRaw, time, rpm)
        }

        throw IOException("Request failed")
    }

    override suspend fun getImage(): Bitmap {
        val request = Request.Builder()
            .url("http://192.168.4.1:81/camera")
            .get()
            .build()

        val response = client.newCall(request).execute()

        if (response.isSuccessful) {
            return BitmapFactory.decodeStream(response.body()!!.byteStream())
        }
        throw IOException("Request failed")
    }

    private fun parseRpm(response: String): Int {
        val fields = response.split(',')
        return fields[1].split('=')[1].toInt()
    }

    private fun parseVoltageRaw(response: String): Float {
        val fields = response.split(',')
        val rawVoltage = fields[0].split('=')[1].toInt()
        val voltage = rawVoltage.toFloat() / VOLTAGE_RAW_MAX
        return voltage
    }

    @VisibleForTesting
    fun buildUrl(params: CarParams): HttpUrl {
        return HttpUrl.Builder()

```

```

        .scheme("http")
        .host("192.168.4.1")
        .port(80)
        .addPathSegments("car")
        .addQueryParameter("steer", params.steer.toString())
        .addQueryParameter("throttle", params.throttle.toString())
        .build()
    }

    companion object {
        private const val VOLTAGE_RAW_MAX = 1023
    }

class MainActivity : AppCompatActivity() {

    private var sendingJob: Job? = null

    private lateinit var sender: CarSender

    private lateinit var controlPreferences: ControlPreferences

    private var cycles = 0
    private var maxTime = 0L

    private val timeSeries = LineGraphSeries<DataPoint>()

    private val errorsMap = mutableMapOf<String, Int>()

    private var maxSpeed = 0

    override fun onResume() {
        super.onResume()
        startSending()
    }

    override fun onPause() {
        super.onPause()
        stopSending()
    }

    var cameraJob: Job? = null

    private fun startSending() {
        sendingJob = GlobalScope.launch(Dispatchers.IO) {
            while (true) {
                try {
                    val throttle =
                        throttleTouchView.progress?.y
                            ?.times(controlPreferences.throttleMax *
ServoConstants.AMPLITUDE)
                            ?.plus(ServoConstants.CENTER)
                    val steer = steerTouchView.progress?.x
                            ?.times(if (controlPreferences.invertSteer) -1 else 1)
                            ?.run { mapSteer(this) }
                    sendValue(steer, throttle)
                } catch (e: IOException) {
                    e.printStackTrace()
                }
                yield()
            }
        }

        cameraJob = GlobalScope.launch(Dispatchers.IO) {
            while (true) {
                val bitmap = runCatching { sender.getImage() }.getOrNull()
                launch(Dispatchers.Main) {
                    bitmap?.also { image.setImageBitmap(it) }
                }
            }
        }
    }

```

```

    }
}

private fun mapSteer(steerValue: Float): Float {
    val range = if (steerValue < 0)
        controlPreferences.steerCenter.toServoValue() -
controlPreferences.steerMin.toServoValue()
    else
        controlPreferences.steerMax.toServoValue() -
controlPreferences.steerCenter.toServoValue()
    return controlPreferences.steerCenter.toServoValue() + steerValue * range
}

private suspend fun sendValue(steer: Float?, throttle: Float?) {
    cycles++
    val steerValue = steer?.toInt() ?:
controlPreferences.steerCenter.toServoValue().toInt()
    val throttleValue = throttle?.toInt() ?: ServoConstants.CENTER
    val response =
        try {
            sender.send(CarParams(steerValue, throttleValue))
        } catch (e: IOException) {
            e.printStackTrace()
            val errorName = e.message!!
            val count = errorsMap[errorName] ?: 0
            errorsMap[errorName] = count + 1
            null
        }
    if (response != null) {
        maxTime = max(response.responseTime, maxTime)
    }
    val speed = response?.rpm?.let { getSpeed(it) }
    val voltageString =
        response?.voltageRaw?.times(controlPreferences.voltageMultiplier)
            ?.let { String.format("%.2f", it) }

    withContext(Dispatchers.Main) {
        speed?.apply {
            speedometer.moveToValue(toFloat())
            maxSpeed = max(maxSpeed, this)
        }
        statTextView.text = "max: $maxTime\n" +
            "V=$voltageString\n" +
            "steer: $steerValue, throttle: $throttleValue\n" +
            "rpm=${response?.rpm}\n" +
            "speed = $speed kmh\n" +
            "maxSpeed = $maxSpeed kmh\n" +
            "$errorsMap\n"

        timeSeries.appendData(
            DataPoint(
                cycles.toDouble(),
                response?.responseTime?.toDouble() ?: 0.0
            ), true, 50
        )
    }
}

private fun getSpeed(rpm: Int): Int {
    val wheelLength = 5.5f * Math.PI
    return ((rpm * 60 * wheelLength)
        / (100 * 1000)).toInt()
}

private fun stopSending() {
    sendingJob?.cancel()
    cameraJob?.cancel()
}
}

```

## ДОДАТОК Г

Рецензії на атестаційну роботу

Рецензія  
на атестаційну роботу магістранта групи ПЗСм-18-1  
Гавриша Олега Владиславовича  
(спеціальність - Інженерія програмного забезпечення,  
освітньо-професійна програма - Програмне забезпечення систем).  
Тема: «Дослідження методів та програмних систем курсової стабілізації руху  
автомобіля»

Структура атестаційної роботи: пояснювальна записка 56 сторінок; 5 розділів; 3 додатки; графічна частина 11 слайдів.

В наш час, з розвитком технічного прогресу, все більше галузей промисловості, медицини, військові та навіть цивільні люди в побуті використовують різноманітні пристрої для автоматизації чи полегшення певних процесів та завдань. Це дозволяє виконувати збільшити безпеку, швидкість, точність, потужність виконання процесів, що в свою чергу дає ще більший поштовх для розвитку індустрії. Певні процеси є небезпечними чи навіть невиконуваними за тими чи іншими причинами для людини і надзвичайно важливим являється автоматизація саме таких видів діяльності.

В першому розділі пояснювальної записки проведено аналіз проблемної області розробки систем стабілізації руху колісних транспортних засобів та аналіз аналогів, сформульовано постановку задачі. В другому розділі наведено вимоги до програмної системи стабілізації руху. В третьому розділі розроблено загальну архітектуру системи, математичну модель задачі та спроектовано базу даних для зберігання даних з датчиків і готових наборів параметрів стабілізації, створено алгоритм стабілізації руху на базі розробленої моделі. В четвертому розділі описано програмну реалізацію системи стабілізації руху колісних роботів. В п'ятому розділі наведено результати тестування системи.

Результати роботи наочно і досить повно відображені в пояснювальній записці та на слайдах презентації. Проект є актуальним і має практичну спрямованість. Пояснювальна записка написана грамотно, якість оформлення - висока, вимоги стандартів дотримані.

До зауважень можна віднести те, що в пояснювальній записці не достатньо повно надано формальний опис алгоритму стабілізації руху.

Магістрант гр. ПЗСм-18-1 Гавриш О.В. готовий до самостійної інженерної діяльності. Атестаційна робота відповідає вимогам до атестаційних робіт магістрів та заслуговує оцінки «відмінно (90, В)». Атестаційну роботу можна подати до захисту в ЕК за спеціальністю 121- «Інженерія програмного забезпечення» (освітньо-професійна програма «Програмне забезпечення систем»).

Рецензент

*К.Б.М. Дод, доцент кафедри ЦИТ*

*[Handwritten signature]*

*[Handwritten signature]*

Рецензія  
на атестаційну роботу магістранта групи ПЗСм-18-1  
Гавриша Олега Владиславовича  
(спеціальність - Інженерія програмного забезпечення,  
освітньо-професійна програма - Програмне забезпечення систем).  
Тема: «Дослідження методів та програмних систем курсової стабілізації руху  
автомобіля»

Структура атестаційної роботи: пояснювальна записка 56 сторінок; 5 розділів; 3 додатки; графічна частина 11 слайдів.

В наш час, з розвитком технічного прогресу, все більше галузей промисловості, медицини, військові та навіть цивільні люди в побуті використовують різноманітні пристрої для автоматизації чи полегшення певних процесів та завдань. Це дозволяє виконувати збільшити безпеку, швидкість точність, потужність виконання процесів, що в свою чергу дає ще більший поштовх для розвитку індустрії. Певні процеси є небезпечними чи навіть невиконуваними за тими чи іншими причинами для людини і надзвичайно важливим являється автоматизація саме таких видів діяльності.

В першому розділі пояснювальної записки проведено аналіз проблемної області розробки систем стабілізації руху колісних транспортних засобів та аналіз аналогів, сформульовано постановку задачі. В другому розділі наведено вимоги до програмної системи стабілізації руху. В третьому розділі розроблено загальну архітектуру системи, математичну модель задачі та спроектовано базу даних для зберігання даних з датчиків і готових наборів параметрів стабілізації, створено алгоритм стабілізації руху на базі розробленої моделі. В четвертому розділі описано програмну реалізацію системи стабілізації руху колісних роботів. В п'ятому розділі наведено результати тестування системи.

Результати роботи наочно і досить повно відображені в пояснювальній записці та на слайдах презентації. Проект є актуальним і має практичну спрямованість. Пояснювальна записка написана грамотно, якість оформлення - висока, вимоги стандартів дотримані.

Серед недоліків виконаної роботи можна відзначити суб'єктивність оцінки правильності роботи системи в цілому, не було введено формальних метрик оцінки коректності роботи.

Магістрант гр. ПЗСм-18-1 Гавриш О.В. готовий до самостійної інженерної діяльності. Атестаційна робота відповідає вимогам до атестаційних робіт магістрів та заслуговує оцінки «відмінно (90, В)». Атестаційну роботу можна подати до захисту в ЕК за спеціальністю 121- «Інженерія програмного забезпечення» (освітньо-професійна програма «Програмне забезпечення систем»).

Рецензент

К. М. Н., доцент,

доцент кафедри ПІ

Толка В.В.

## ДОДАТОК Д

Відгук на атестаційну роботу

## ВІДГУК

на атестаційну роботу магістра

Гавриша Олега Владиславовича, ст. гр. ПЗСм-18-1(спеціальність - Інженерія програмного забезпечення,  
освітньо-професійна програма - Програмне забезпечення систем).**Тема атестаційної роботи:**«Дослідження методів та програмних систем курсової стабілізації руху  
автомобіля»

Структура атестаційної роботи: пояснювальна записка 56 сторінок; 5 розділів; 3 додатки; графічна частина 11 слайдів.

В наш час, з розвитком технічного прогресу, все більше галузей промисловості, медицини, військові та навіть цивільні люди в побуті використовують різноманітні пристрої для автоматизації чи полегшення певних процесів та завдань. Це дозволяє виконувати збільшити безпеку, швидкість, точність, потужність виконання процесів, що в свою чергу дає ще більший поштовх для розвитку індустрії. Певні процеси є небезпечними чи навіть невиконуваними за тими чи іншими причинами для людини і надзвичайно важливим являється автоматизація саме таких видів діяльності.

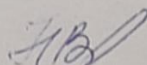
Магістерське дослідження присвячено актуальному питанню розробки математичної моделі стабілізації руху колісних транспортних засобів, що дозволяє підвищити ефективність та безпеку їх роботи.

Магістрант гр. ПЗСм-18-1 Гавриш О.В. проаналізував існуючі аналоги та проблеми, пов'язані з стабілізацією руху та розробив математичну модель задачі стабілізації руху, на базі моделі побудував алгоритм корекції контролюючих параметрів виконуючих агрегатів. Програмна система реалізована з використанням сучасних програмних засобів, клієнт-серверних, вбудовуваних та інших інформаційних технологій.

Магістрант провів досить детальний аналіз технічної та спеціальної літератури, використав отриману інформацію для обґрунтування прийнятих в роботі науково-дослідних рішень, процитував її в пояснювальній записці. Записка написана грамотно, вимоги стандартів дотримані. Результати роботи досить повно відображені в пояснювальній записці та на слайдах презентації. За результатами атестаційної роботи написана та подана до опублікування стаття «Дослідження методів та програмних систем курсової стабілізації руху автомобіля» в матеріалах IV міжнародної науково-практичної конференції в Ліверпулі «Scientific Achievements of Modern Society».

Вважаю, що магістрант гр. ПЗСм-18-1 Гавриш О.В. готовий до самостійної інженерної діяльності. Атестаційну роботу можна подати до захисту в ЕК за спеціальністю 121- «Інженерія програмного забезпечення» (освітньо-професійна програма «Програмне забезпечення систем»).

Керівник атестаційної роботи магістра  
проф. каф. ПІ



Білоус Н.В.

«12» 12 2019 р.