

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет електроенерготехніки та автоматики

(повне найменування інституту, факультету)

Кафедра автоматизації електромеханічних систем та електроприводу

(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

Сергій ПЕРЕСАДА

(підпис)

“ ” 20\_\_ р.

**Дипломний проект**

**на здобуття ступеня бакалавра**

за освітньо-професійною програмою «Електромеханічні системи автоматизації,

електропривод та електромобільність»

спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»

(код і назва)

на тему: Електромеханічна система моноколеса на основі безколекторного

двигуна постійного струму

Виконав (-ла): студент (-ка) III курсу, групи ЕП-п71

(шифр групи)

Крячко Ярослав Сергійович

(прізвище, ім'я, по батькові)

(підпис)

Керівник асистент Желінський Микола Миколайович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по-батькові)

(підпис)

Консультант \_\_\_\_\_

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ім'я, по-батькові)

(підпис)

Рецензент асистент, к.т.н. Монахов Євген Андрійович

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по-батькові)

(підпис)

Засвідчую, що у цьому дипломному проекті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент (-ка)

(підпис)

Київ – 2020 р.

## ВІДОМІСТЬ ДИПЛОМНОГО ПРОЕКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проект	2	
2	A4	141.0007.006.БР	Пояснювальна записка	81	
3	A1	141.0007.006.БР	Креслення загального виду	1	
4	A1	141.0007.006.БР	Схема електрична принципова	1	
5	A1	141.0007.006.БР	Структурна схема алгоритму керування та результати досліджень перехідних процесів алгоритму керування	1	

				141.0007.006.БР		
	ПБ	Підп.	Дата	Відомість дипломного проекту	Лист	Листів
Розробн.	Крячко Я. С.		15.06		2	81
Керівн.	Желінський М. М.		15.06			
Консульт.					КПІ ім. Ігоря Сікорського Каф. АЕМС-ЕП Гр. ЕП-п71	
Н/контр.	Теряєв В. І.					
Зав.каф.	Пересада С. М.					

## **Пояснювальна записка до дипломного проекту**

на тему: Електромеханічна система моноколеса на основі  
безколекторного двигуна постійного струму

---

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет електроенерготехніки та автоматики  
(повна назва)

Кафедра автоматизації електромеханічних систем та електроприводу  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 141 Електроенергетика, електротехніка та електромеханіка  
(код і назва)

Освітньо-професійна програма – Електромеханічні системи автоматизації, електропривод та електромобільність

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
\_\_\_\_\_ Сергій ПЕРЕСАДА  
(підпис)

«\_\_» \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ**  
**на дипломний проект студенту**  
Крячко Ярослав Сергійович  
(прізвище, ім'я, по батькові)

1. Тема проекту: Електромеханічна система моноколеса на основі безколекторного двигуна постійного струму

керівник проекту асистент Желінський Микола Миколайович  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «\_\_» \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Термін подання студентом проекту 09.06.2020 р.

3. Вихідні дані до проекту Моноколесо InMotion V8 Black (IM-V8-BK) як реальний аналог, траєкторія руху в умовах міського циклу, мінімальний запас ходу на одному заряді АКБ – 50 км, максимальна швидкість руху – 30 км/год

4. Зміст пояснювальної записки: 1 Аналітичний огляд електромеханічної системи на основі моноколесу; 2 Розрахунок та вибір основних елементів електромеханічної системи; 3 Проектування системи керування; 4 Дослідження статичних та динамічних режимів електромеханічної системи; 5 Спецчастина. Розробка програмного забезпечення для системи керування двигуна

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): Креслення загального виду електромеханічної системи моноколесу (формат A1); схема електрична принципова силової частини електроприводу, схема електрична принципова керуючого пристрою (формат A1); структурна схема системи, результати дослідження статичних та динамічних режимів (формат A1)

6. Консультанти розділів проекту\*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

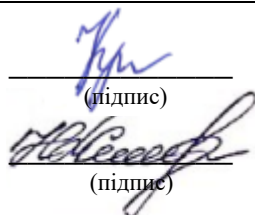
\* Якщо визначені консультанти. Консультантом не може бути зазначено керівника дипломного проекту.

7. Дата видачі завдання 20 березня 2020 р.

## КАЛЕНДАРНИЙ ПЛАН – ГРАФІК

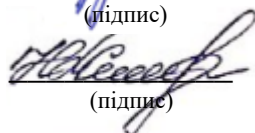
Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Обсяг, с	Термін виконання	Позначки виконання
1 Аналітичний огляд електромеханічної системи на основі моноколесу. 1.1 Опис електромеханічної системи моноколесу. 1.2 Способи керування електромеханічної системи моноколесу. 1.3 Формулювання вимог до електромеханічної системи моноколесу.	15	03.04.20	Виплокаро
2 Розрахунок та вибір основних елементів електромеханічної системи. 2.1 Енергосиловий розрахунок параметрів циклу руху. 2.2 Обґрунтування вибору потужності електродвигуна та акумуляторної батареї 2.3 Розрахунок та вибір силового обладнання 2.4 Розробка і опис схеми силової частини і схеми керування	30	24.04.20	Виплокаро
3 Проектування системи керування 3.1 Обґрунтування принципу керування електроприводом 3.2 Структурно-параметричний синтез системи керування 3.3 Розрахунок параметрів регуляторів і зворотних зв'язків	10	15.05.20	Виплокаро
4 Дослідження статичних та динамічних режимів електромеханічної системи	5	22.05.20	Виплокаро
5 Спецчастина. Розробка програмного забезпечення для системи керування двигуна	15	03.06.20	Виплокаро
Оформлення пояснювальної записки та графічної частини. Підготовка презентації		05.06.20	Виплокаро
Разом	60-90		

Студент

  
 (підпис)

Ярослав КРЯЧКО

Керівник проекту

  
 (підпис)

Микола ЖЕЛІНСЬКИЙ

## РЕФЕРАТ

Бакалаврська робота складається із вступу, п'яти розділів, висновка, списку використаної літератури і додатка (81 сторінки пояснювальної записки, 36 рисунків, 5 таблиць, 18 використаних джерел).

*Мета роботи* – розробка електромеханічної системи моноколеса на основі безколекторного двигуна постійного струму.

*Об'єкт дослідження* – безколекторний двигун постійного струму.

*Предмет дослідження* – алгоритм векторного керування моментом безколекторного двигуна постійного струму.

В першому розділі проведено аналітичний огляд електромеханічної системи моноколеса, розглянуто способи керування, сформувано вимоги до системи.



В другому розділі досліджено режим роботи двигуна та розраховано основні його параметри. Проведено розрахунок та вибір основних компонентів силової схеми та схеми керування. Розроблено схему електричну.

В третьому розділі проведено структурно-параметричний синтез системи керування, створена її математична модель в середовищі MatLab Simulink.

В четвертому розділі досліджено перехідні процеси системи.

В п'ятому розділі розроблено «прошивку» для контролера керування.

МОНОКОЛЕС, ЕЛЕКТРОМЕХАНІЧНА СИСТЕМА, АВТОНОМНИЙ ІНВЕРТОР НАПРУГИ, СИНХРОННИЙ ДВИГУН З ПОСТІЙНИМИ МАГНІТАМИ, ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРА, БЕЗКОЛЕКТОРНИЙ ДВИГУН ПОСТІЙНОГО СТРУМУ

					141.0007.006.БР						
					Електромеханічна система моноколеса на основі безколекторного двигуна постійного струму Реферат	Лім.			Маса	Масштаб	
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Крячко Я. С.		15.06							
Перевір.		Желінський М. М.		15.06							
Т. Контр.						Арк. 7			Аркушів 81		
Реценз.		Монахов Є. А.				КПІ ім. Ігоря Сікорського, кафедра АЕМС-ЕП, ФЕА, група ЕП-п71					
Н. Контр.		Теряєв В. І.									
Затверд.		Пересада С. М.									

## SUMMARY

The bachelor's thesis consists of an introduction, five chapters, a conclusion, a list of references and an appendix (81 pages of explanatory note, 36 figures, 5 tables, 18 sources used).

The purpose of the work is to develop an electromechanical system of a unicycle based on a brushless DC motor.

The object of research is a brushless DC motor.

The subject of research is the algorithm of vector control of the moment of the brushless DC motor.

In the first section the analytical review of electromechanical system of a unicycle is carried out, ways of management are considered, requirements to system are formed.


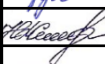
In the second section the mode of operation of the engine is investigated and its basic parameters are calculated. The calculation and selection of the main components of the power scheme and control scheme. An electrical circuit has been developed.

In the third section the structural-parametric synthesis of the control system is carried out, its mathematical model in the MatLab Simulink environment is created.

The fourth section examines the transients of the system.

The fifth section develops "firmware" for the control controller.

**MONOWHEEL, ELECTROMECHANICAL SYSTEM, AUTONOMOUS VOLTAGE INVERTER, SYNCHRONOUS MOTOR WITH PERMANENT MAGNETS, PROGRAMMING MICROCONTROLLER**

					141.0007.006.БР							
					Electromechanical system of a unicycle based on a brushless DC motor Summary	Let.			Mass		Scale	
Ch.	Sheet	№ document	Sign.	Date								
Develop.		Крячко Я. С.		15.06								
Check.		Желінський М. М.		15.06								
T. contr.						Sh. 8			Sheets 81			
Review.		Монахов Е. А.				Igor Sikorsky KPI, FEA, AEMS-ED, EP-p71						
N. contr.		Теряев В. І.										
Approv.		Пересада С. М.										



## ЗМІСТ

Вступ.....	11
1 Аналітичний огляд електромеханічної системи на основі моноколесу .	13
1.1 Опис електромеханічної системи моноколесу.....	13
1.2 Способи керування електромеханічної системи моноколесу .....	16
1.2.1 Трапецеїдальне керування .....	17
1.2.2 Полеорієнтоване керування .....	18
1.2.3 Полеорієнтоване керування СДПМ по датчику положення .....	19
1.2.4 Полеорієнтоване керування СДПМ без датчика положення ....	19
1.3 Формулювання вимог до електромеханічної системи моноколесу..	20
Висновки до розділу 1 .....	22
2 Розрахунок та вибір основних елементів електромеханічної системи ..	23
2.1 Енергосиловий розрахунок параметрів циклу руху.....	24
2.2 Обґрунтування вибору потужності електродвигуна та акумуляторної батареї.....	25
2.3 Розрахунок та вибір силового обладнання.....	34
2.4 Розробка і опис схеми силової частини і схеми керування.....	36
Висновки до розділу 2 .....	42
3 Проектування системи керування .....	44
3.1 Обґрунтування принципу керування електроприводом.....	44
3.2 Структурно-параметричний синтез системи керування.....	44
3.2.1 Постановка задачі .....	46
3.2.2 Розв'язання задачі .....	47
3.3 Розрахунок параметрів регуляторів і зворотних зв'язків .....	52
Висновки до розділу 3 .....	53
4 Дослідження статичних та динамічних режимів електромеханічної системи .....	54
Висновки до розділу 4 .....	54

5 Спецчастина. розробка програмного забезпечення для системи керування  
двигуна ..... 59

Висновки до розділу 5 ..... 74

Висновки ..... 75

Перелік джерел посилання ..... 76

Додаток А..... 78

## ВСТУП

Час – один з найцінніших ресурсів сучасної людини і раціональне його використання ледь не найважливіше завдання. Тому для цієї мети розроблено безліч електронних гаджетів. Одним з нових, але вже таких популярних, електричних транспортних засобів є моноколес. Це стильний, компактний, легкий в керуванні та досить швидкісний (до 45 км/год) засіб пересування – те, що потрібно для великого міста з постійними заторами.

Перші моноколеси з'явилися в 2009-2010 роках від виробника «Solowheel». Пізніше подібні транспортні засоби стали популярними і їх виробництвом зайнялися багато інших виробників.

Зовні такий транспорт виглядає дуже компактно і просто. Це колесо під пластиковим корпусом і з двома педалями з боків. Усередині знаходяться мотор і акумулятор, які і забезпечують рух. Таке моноколесо можна використовувати для того, щоб швидко дістатися до офісу, для прогулянок у парку або щодо близьких поїздки. Звісно, моноколесо не зможе повноцінно замінити автомобіль, але його цілком можна використовувати замість електросамоката, гіроскутера або навіть велосипеда.

Як і в інших видах електричного транспорту, в моноколесах величезну роль грає ємність акумулятора. Чим вона вища - тим довше і більше проїде ваше моноколесо.

Фактичні показники слід вираховувати з урахуванням ваги користувача, якості дороги і її типу. У більшості випадків за фактом моноколесо проїжджає набагато менше, ніж заявлено виробником. Тому ємний акумулятор ніколи не буде недоліком.

Електричний мотор, встановлений в моноколесі, багато в чому визначає максимальну швидкість і дальність ходу транспорту.

Важливим фактором появи таких пристроїв стало розповсюдження безколекторних двигунів постійного струму (brushless DC motor, BLDC) або, як його ще називають, синхронний двигун зі збудженням від постійних магнітів

(synchronous motor on permanent magnets, SMPM), які, по суті, мало чим відрізняються.

Причиною такого широкого розповсюдження став розвиток електроніки і, в тому числі, поява недорогих силових транзисторних ключів. Також важливу роль зіграла поява потужних рідкоземельних постійних магнітів NdFeB та SmCo.

Ідея безколекторного двигуна з'явилася на зорі електротехніки та електромеханіки. Але, в силу неготовності технологій, чекала свого часу до 1962 року, коли з'явився перший комерційний безколекторний двигун постійного струму. Понад півсторіччя існують різні серійні реалізації даного типу електроприводу. В останні 10 років реалізація та ринок безколекторних двигунів розвивається значно швидше, ніж ринок машин з постійними магнітами інших типів.

Такі двигуни мають масу переваг перед іншими типами двигунів, наприклад, за рахунок відсутності колектора більш надійні, ніж звичайні двигуни постійного струму, та не мають обмежень по частоті обертання; в порівнянні з асинхронними двигунами мають високий коефіцієнт корисної дії, вищий коефіцієнт потужності, мають менші розміри.

Таким чином, метою даної бакалаврської роботи є створення електромеханічної системи моноколесу на основі безколекторного двигуна постійного струму.

# **1 АНАЛІТИЧНИЙ ОГЛЯД ЕЛЕКТРОМЕХАНІЧНОЇ СИСТЕМИ НА ОСНОВІ МОНОКОЛЕСУ**

## **1.1 Опис електромеханічної системи моноколесу**

Моноколес - це автотранспортний засіб з системою гіростабілізації колеса. За рахунок відносно малих габаритів, його можна транспортувати і зберігати в будь-якому місці. Корпус виготовлений з міцного пластику, усередині якого і розташоване саме колесо. Величина корпусу жодним чином не впливає на підйомну тягу, але чим менше колесо, тим крутіше підйоми і перешкоди можна подолати. На панелі зверху вбудований роз'єм для зарядки, індикатор заряду батареї і кнопка вкл/викл, а з боків розташовані підніжки, які, зазвичай, мають властивість складатися для зручності при переміщенні.

Нижня частина моноколесу і є колесо, яке складається з осі, щодо якого колесо вирівнюється при роботі гіроскопів, ободу, спиць і покришки. Автоматичне балансування моноколесу забезпечують різні гіроскопи, датчики і акселерометри, що дозволяє керувати ним нахилом тіла. В середині колеса знаходиться електродвигун, потужність якого, між 250-2000 Вт, гіроскопи, що відповідають за баланс колеса і акумуляторні батареї. Акумуляторна батарея складається з потужних акумуляторів, з'єднаних між собою і підключених до комп'ютера (контролера), який регулює процес заряду.

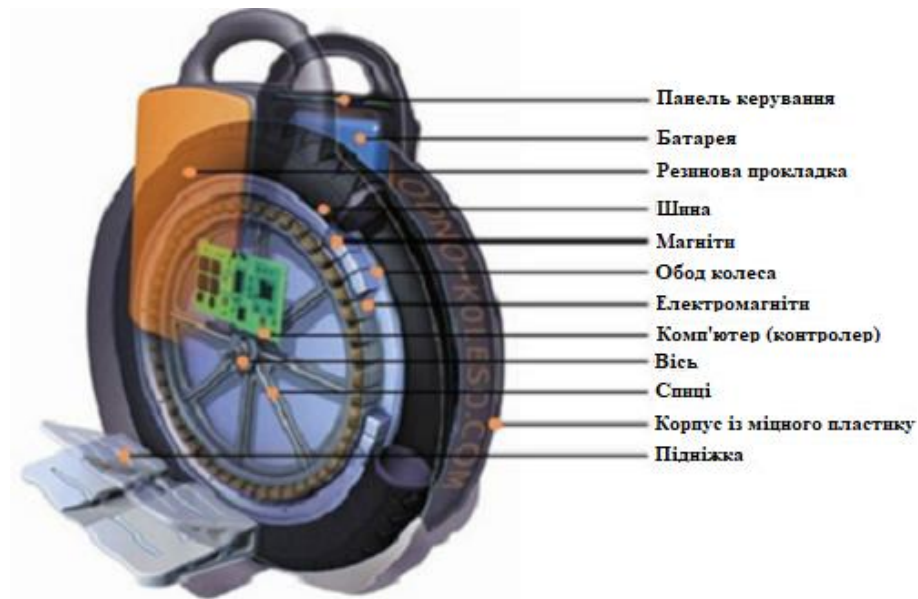


Рисунок 1.1 – Будова моноколесу

Зазвичай, в комплект входить ремінь безпеки, який кріпиться до ручки для утримання колеса при навчанні; маленькі тренувальні колеса, що спрощують навчання і кріпляться до підніжок, зарядний пристрій і подовжувач для підкачки колеса.

Заряду вистачає на 5-100 км в залежності від моделі при швидкості до 30 км/год. Щоб повністю зарядити акумулятор, потрібно 1.5-10 години.

В середині моноколесу знаходяться електромагніти. Після увімкнення гіроскоп, який, як уже зазначалося вище, забезпечує рівновагу колеса, розпізнаючи зміщення центру ваги, визначає нахил, після чого від акумулятора на кожен з електромагнітів надходить струм, згодом формуючи магнітне поле. Магніти мають полюса: північний і південний. При положенні північ/північ і південь/південь, магніти будуть відштовхуватися, що призведе моноколес до руху вперед. Швидкість збільшується прямопропорційно силі натискання ступні на підніжку (рис. 1.2).

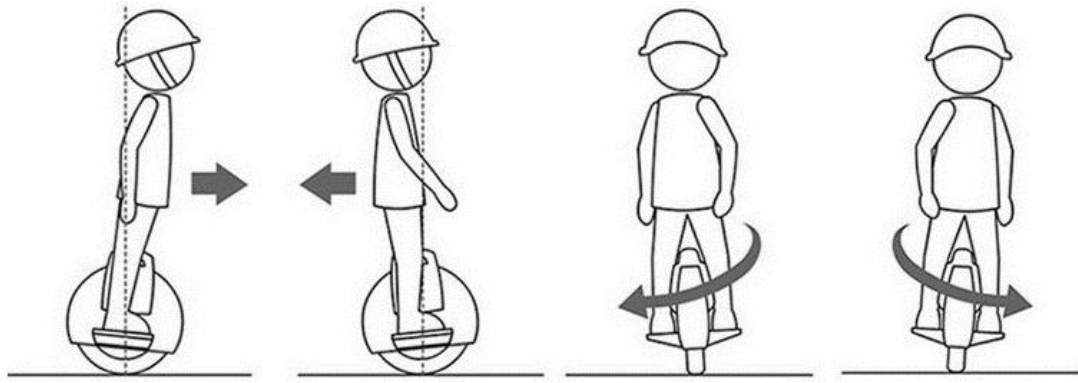


Рисунок 1.2 – Керування моноколесом

Гальмування відбувається при натисканні на підніжку п'ятою, при тривалому натисканні моноколес змінює напрямок і починає рухатися назад, магніти при цьому притягуються і знаходяться в положенні північ/південь (рис. 1.3). Повороти забезпечуються натисканням на підніжку в потрібну сторону.

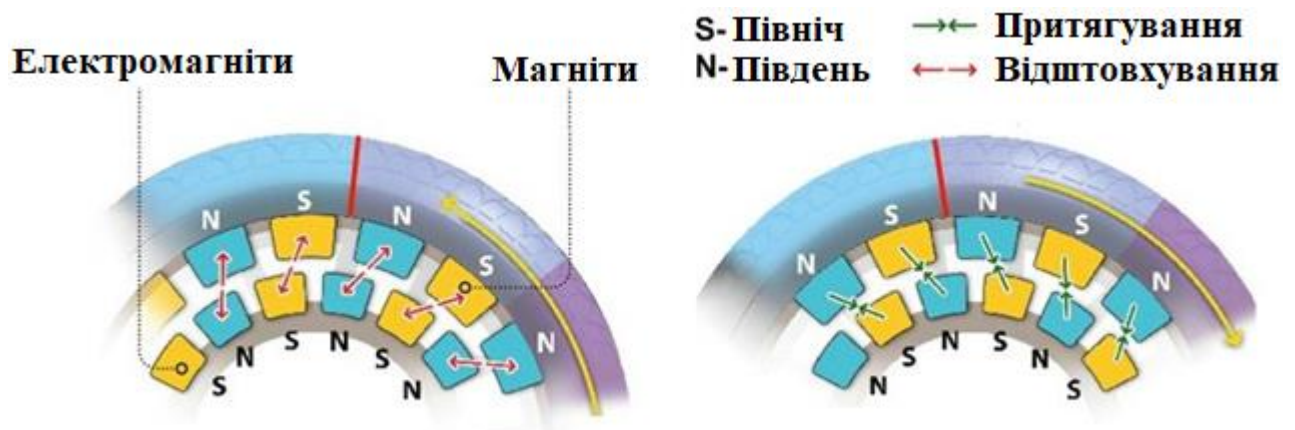


Рисунок 1.3 – Розміщення магнітів в моноколесі

Також передбачений механізм обмеження швидкості, який спрацьовує при перевищенні безпечної швидкості. Піднімаються під час їзди підніжки, запобігають подальшому збільшенню швидкості. Якщо ж швидкість їзди не знижується, лунає на початку переривчастий, а потім суцільний попереджувальний сигнал.

Моноколес добрий і тим, що завдяки своїй стійкості і звичайно ж професійному вмінню людини, дозволяє пересуватися не тільки по рівній дорозі,

але і по ґрунтовій, похилій поверхні, як вниз, так і вгору. Максимальний кут нахилу колеса досягає  $30^\circ$ . При бажанні і обов'язковому ознайомленні з заходами, його можна випробувати навіть під час дощу [1].

## 1.2 Способи керування електромеханічної системи моноколесу

Головна відмінність між синхронним двигуном з постійними магнітами (СДПМ, рис. 1.4) і асинхронним електродвигуном полягає в роторі. Проведені дослідження показують, що СДПМ має ККД приблизно на 2% більше, ніж високоефективний (IE3) асинхронний електродвигун, за умови, що статор має однакову конструкцію, а для управління використовується один і той же частотний перетворювач. При цьому синхронні електродвигуни з постійними магнітами в порівнянні з іншими електродвигунами мають кращі показники: потужність/обсяг, момент/інерція, тощо.

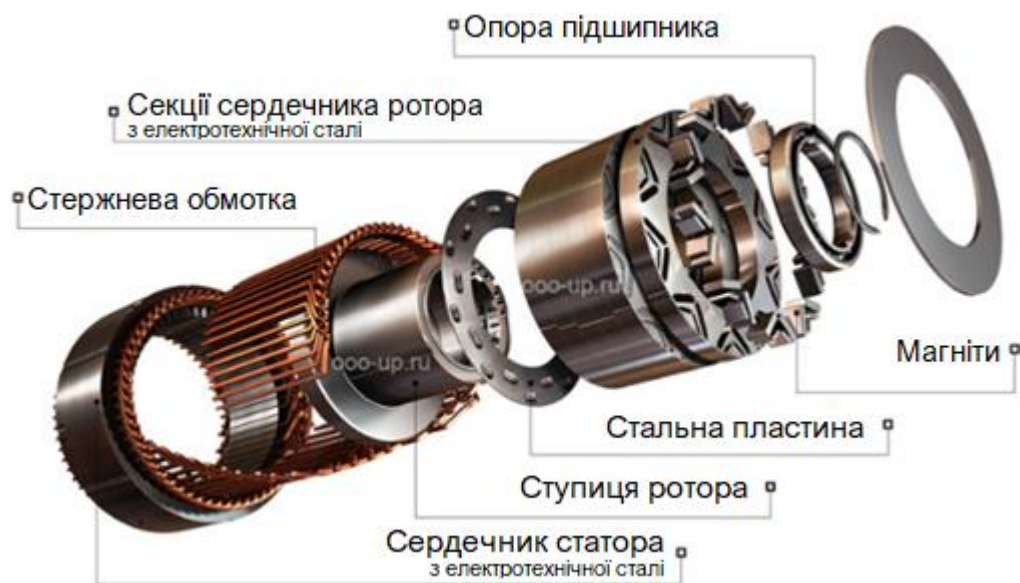


Рисунок 1.4 – Синхронний двигун з постійними магнітами

Статор складається з корпусу і сердечника з обмоткою. Найбільш поширені конструкції з двох- і трифазною обмотками.

Ротор складається з постійних магнітів. В якості постійних магнітів використовуються матеріали з високою коерцитивною силою.



Принцип дії синхронного електродвигуна базується на взаємодії магнітного поля статора і постійного магнітного поля ротора. Концепція обертового магнітного поля статора синхронного електродвигуна така ж, як і у трифазного асинхронного електродвигуна.

Магнітне поле ротора, взаємодіючи з синхронним змінним струмом обмоток статора, відповідно до закону Ампера, створює крутий момент, змушуючи ротор обертатися.

Постійні магніти, розташовані на роторі СДПМ, створюють постійне магнітне поле. При синхронній швидкості обертання ротора з полем статора, полюса ротора зчіплюються з обертовим магнітним полем статора. У зв'язку з цим СДПМ не може сам запуститися при підключенні його безпосередньо до мережі трифазного струму (частота струму в мережі 50Гц).

Для роботи синхронного двигуна з постійними магнітами обов'язково потрібна система керування, наприклад, частотний перетворювач або сервопривод. При цьому існує велика кількість способів управління, що реалізуються системами контролю. Вибір оптимального способу управління, головним чином, залежить від завдання, яке ставиться перед електроприводом.

Для вирішення нескладних завдань, зазвичай, використовується трапецеїдальне керування по датчикам Холла (наприклад, комп'ютерні вентилятори). Для вирішення завдань, які вимагають максимальних характеристик від електроприводу, зазвичай вибирається полеорієнтоване керування.

### **1.2.1 Трапецеїдальне керування**

Одним з найпростіших методів керування синхронним двигуном з постійними магнітами є трапецеїдальне керування (рис. 1.5).

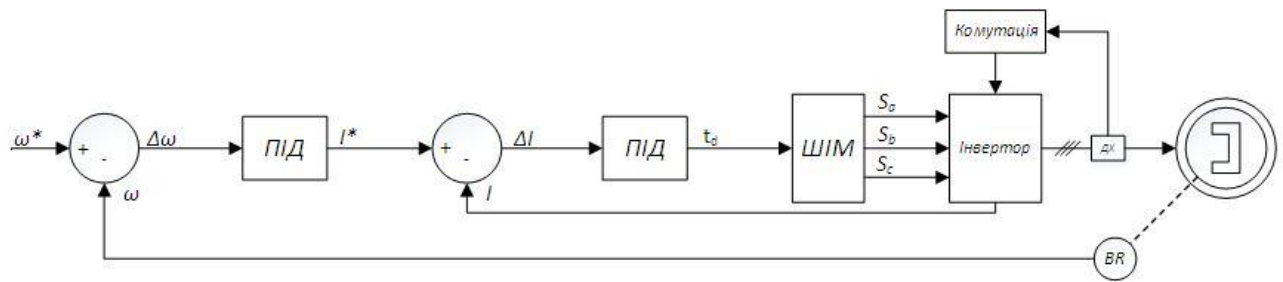


Рисунок 1.5 – Трапецеїдальне керування по датчикам Холла

Трапецеїдальне керування застосовується для керування СДПМ з трапецеїдальною зворотною ЕРС. При цьому цей метод дозволяє також керувати СДПМ з синусоїдальною зворотною ЕРС, але тоді середній момент електроприводу буде нижче на 5%, а пульсації моменту складуть 14% від максимального значення. Існує трапецеїдальне керування без зворотного зв'язку і зі зворотним зв'язком по положенню ротора.

Керування без зворотного зв'язку не оптимальне і може привести до виходу СДПМ з синхронізму, тобто до втрати керованості.

Керування зі зворотним зв'язком можна розділити на:

- а) трапецеїдальне керування по датчику положення (зазвичай – по датчикам Холла);
- б) трапецеїдальне керування без датчика (бездатчикове трапецеїдальне керування).

Як датчик положення ротора при трапецеїдальному керуванні трифазного СДПМ зазвичай використовуються три датчика Холла, вбудовані в електродвигун, які дозволяють визначити кут з точністю  $\pm 30$  градусів. При такому керуванні вектор струму статора приймає тільки шість положень на один електричний період, в результаті чого на виході є пульсації моменту.

## 1.2.2 Полеорієнтоване керування

Полеорієнтоване керування дозволяє плавно, точно і незалежно керувати швидкістю і моментом безщіткового електродвигуна. Для роботи алгоритму

полеорієнтованого керування потрібно знати положення ротора безщіткового електродвигуна.

Існує два способи визначення положення ротора:

- а) по датчику положення;
- б) без датчика - за допомогою обчислення кута системою керування в реальному часі на основі наявної інформації.

### **1.2.3 Полеорієнтоване керування СДПМ по датчику положення**

Як датчик кута використовуються наступні типи датчиків:

- а) індуктивні: синусно-косінусний обертовий трансформатор (СКОТ), редуктосин, індуктосин і ін .;
- б) оптичні;
- в) магнітні: магніторезистивні датчики.

### **1.2.4 Полеорієнтоване керування СДПМ без датчика положення**

Завдяки бурхливому розвитку мікропроцесорів з 1970-х років почали розроблятися бездатчикові векторні методи керування безщітковими електродвигунами змінного струму. Перші бездатчикові методи визначення кута були засновані на властивості електродвигуна генерувати зворотний ЕРС під час обертання. Зворотний ЕРС двигуна містить в собі інформацію про стан ротора, тому обчисливши величину зворотного ЕРС в стаціонарній системі координат можна розрахувати положення ротора. Але, коли ротор нерухомий, зворотна ЕРС відсутня, а на низьких обертах зворотна ЕРС має маленьку амплітуду, яку складно відрізнити від шуму, тому даний метод не підходить для визначення положення ротора двигуна на низьких обертах.

Існує два найпоширеніших варіанти запуску СДПМ:

- а) запуск скалярним методом: запуск за заздалегідь визначеною характеристикою залежності напруги від частоти, але скалярне керування сильно обмежує можливості системи керування і параметри електроприводу в цілому;

б) метод накладення високочастотного сигналу: працює тільки з СДПМ, у якого ротор має явно виражені полюси.

На поточний момент бездатчикове полеорієнтоване керування СДПМ у всьому діапазоні швидкостей можливе тільки для двигунів з ротором з явно вираженими полюсами [2].

### 1.3 Формулювання вимог до електромеханічної системи моноколесу

Показниками регулювання швидкості є:

**Діапазон регулювання** – характеризується співвідношенням значень максимальної швидкості  $\omega_{max}$  до мінімальної  $\omega_{min}$ , які можуть бути реалізовані у даному способі регулюванні.

$$D = \frac{\omega_{max}}{\omega_{min}} = \frac{n_{max}}{n_{min}},$$

де  $D$  – діапазон регулювання.

Максимальна швидкість двигуна – це або швидкість двигуна на природній механічній характеристиці, або максимальна паспортна швидкість.

Звичайно, діапазон регулювання визначається при номінальному моменті двигуна за природною і штучною механічною характеристикою двигуна.

Згідно з формули для діапазону регулювання, можна записати:

$$D = \frac{\omega_{max}}{\omega_{min}} = \frac{\omega_{max}}{\Delta\omega} = \frac{\omega_{max} \cdot \Delta M}{\Delta\omega \cdot \Delta M} = \frac{\omega_{max} \cdot \beta}{M_i};$$

де  $\Delta\omega$  – спад швидкості при номінальному моменті.

$\omega_{min}$  – мінімальна швидкість двигуна, що відповідає межі мінімальної швидкості завданого діапазону регулювання.

Таким чином чим більше жорсткість механічної характеристики тим більший діапазон регулювання.

**Статична похибка регулювання швидкості** – характеризує величину похибки за швидкістю при коливаннях моменту.

У вимогах до системи задаються відносною похибкою при зміні моменту від «0» до номінального значення:

$$\delta = \frac{\Delta\omega}{\omega_{min}} = \frac{\Delta\omega}{\omega_{min.i}} \cdot D.$$

**Плавність регулювання** – характеризує кількість штучних механічних характеристик, можливих при даному діапазоні регулюванні, плавність регулювання тим більша, чим менший коефіцієнт плавності

$$k_p = \frac{\omega_i}{\omega_{i-1}},$$

де  $\omega_i, \omega_{i-1}$  – швидкості на різних штучних механічних характеристиках при одному моменті навантаження.

**Стабільність роботи на заданій швидкості** – стабільність роботи тим більша, чим більша жорсткість на штучній механічній характеристиці.

**Економічність регулювання** – визначається ККД двигуна з одного боку, та втратами на обладнання та експлуатацію з іншого боку. Загальною величиною економічної ефективності є термін – відшкодування втрат.

**Похідна прискорення і уповільнення (ривок)** – визначається необхідністю обмеження динамічних навантажень в механічній передачі під час перехідних процесів і завданням забезпечення необхідного комфорту для водія. Обмеження величин прискорення і ривка повинно забезпечувати високу плавність перехідних процесів і тим самим виключити можливість падіння водія.

Крім того, керування моноколесом має бути зручним та інтуїтивно зрозумілим. Тому не менш важливо визначити залежність кута нахилу водія відносно колеса (двигуна) моноколесу, отриманий графік зображено на рис. 1.6.

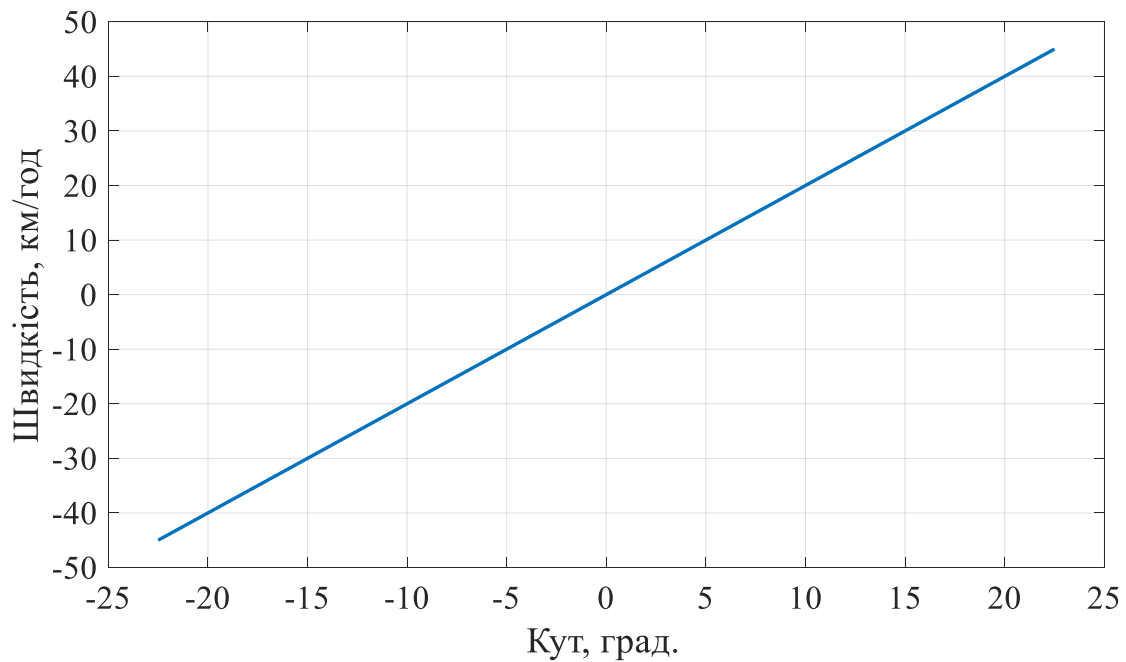


Рисунок 1.6 – Залежність швидкості моноколесу від куту нахилу водія

### Висновки до розділу 1

1. Всі вимоги до моноколесу можна розглядати як базові вимоги до електротранспорту – вимоги щодо безпеки та зручності експлуатації (захисти, блокування, додаткові елементи, тощо) і вимоги до двигуна (діапазон керування, плавність регулювання та руху, перевантажувальна здатність, тощо).

2. Вимоги до двигуна:

- $\omega_{\max} > 34.5 \text{ рад/с}$  ( $v_{\max} > 25 \text{ км/год}$ );
- $a < 1 \text{ м/с}^2$ ;
- можливість реверсу та рекуперації;
- висока перевантажувальна здатність (не менш ніж 2.5).

3. Вимоги до електромеханічної системи:

- захист від розгону вище максимальної швидкості;
- захист АКБ від перезаряду;
- можливість роботи в рекуперативному режимі;
- плавні пуск та гальмування, плавність зміни швидкості руху;
- високий рівень захищеності (мінімум IP42).

## **2 РОЗРАХУНОК ТА ВИБІР ОСНОВНИХ ЕЛЕМЕНТІВ ЕЛЕКТРОМЕХАНІЧНОЇ СИСТЕМИ**

Головним фактором при розробці системи електротранспорту є сили, що діють на нього під час руху. Це тертя об підлогу (поверхню дороги), опір повітря та притискна сила, які, в свою чергу, залежать від типу поверхні дороги, площі транспорту (і, в даному випадку, водія) і швидкості руху та ваги навантаженого транспорту. Крім того, потрібно врахувати вимоги до електротранспорту, такі як необхідна (бажана) відстань, яку можна подолати на одному заряді, що ставить завдання для потужності акумуляторної батареї (АКБ).

В якості швидкості руху взято середні значення швидкості для електротранспорту в умовах міського руху, а також середньостатистичні дані (площа людини, тертя об асфальт, тощо) і параметри схожих гаджетів (потужність двигуна, ємність АКБ, тощо). Виходячи з цих даних, розраховано необхідну потужність для такої швидкості і обрано двигун, який здатний працювати в таких умовах. Крім того, проведено його перевірку на перегрів, що є обов'язковою умовою для двигуна в будь якій системі за будь яких умов роботи, особливо в цьому випадку.

Керуючись потужністю двигуна за цикл було розраховано необхідну потужність АКБ, що здатна забезпечити роботу двигуна протягом всього циклу роботи.

Так як АКБ – джерело постійної напруги, а двигун – трифазний, необхідно організувати генерування трифазної змінної напруги для двигуна при живленні від джерела постійної напруги (АКБ). Для цього використовується автономний інвертор напруги (АІН), керований ШІМ.

Наступним кроком є розробка силової схеми для двигуна і АКБ, а також схеми керування двигуном.

## 2.1 Енергосиловий розрахунок параметрів циклу руху

Розрахунок ведеться на основі отриманого графіку швидкості моноколесу по міському середньостатистичному циклу [4]. Графіки швидкості по шляху, який він проходить при цьому, зображені на рис. 2.1 – 2.3.

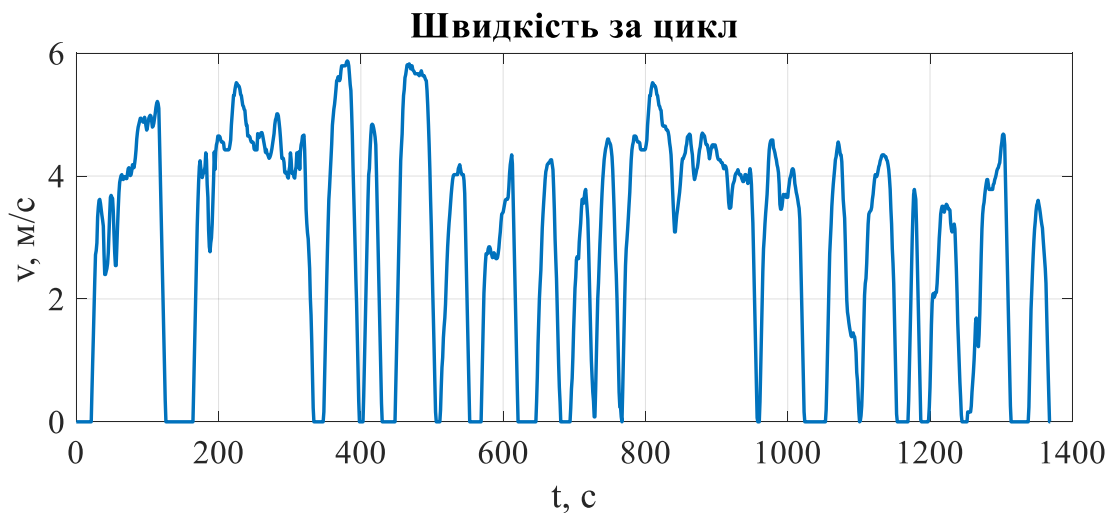


Рисунок 2.1 – Швидкість моноколесу за цикл

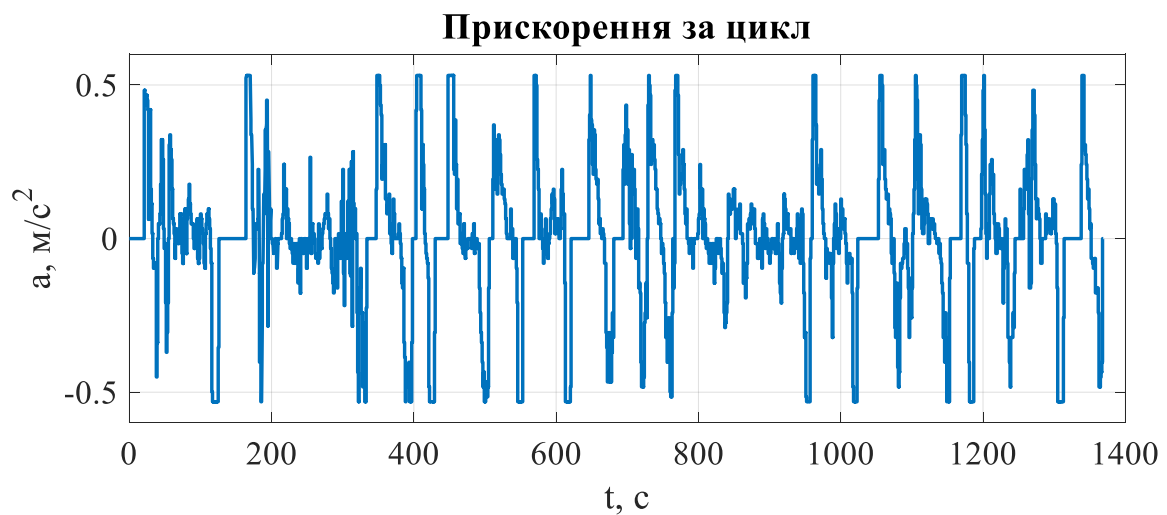


Рисунок 2.2 – Прискорення моноколесу під час руху за один цикл



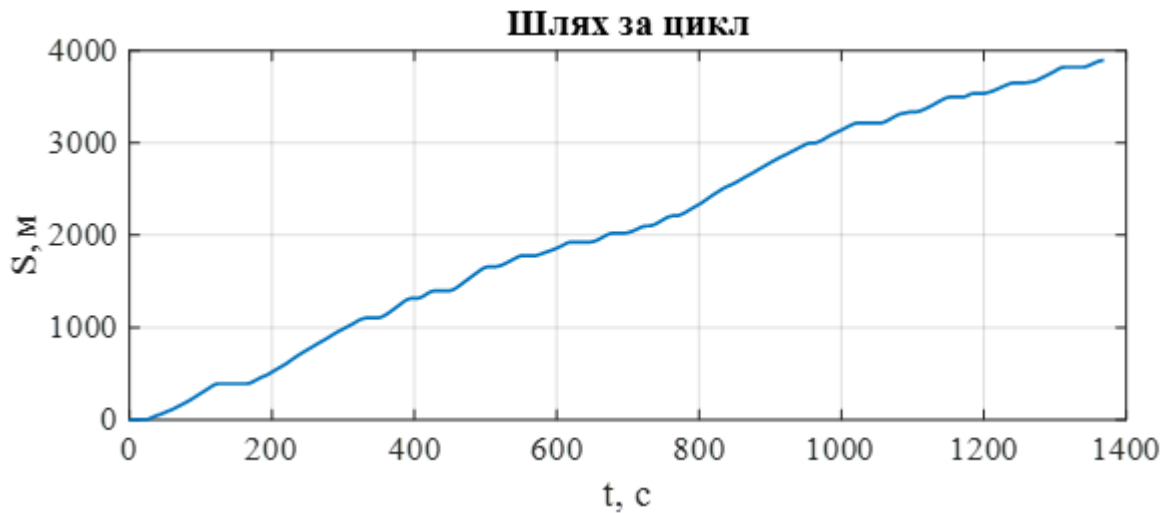


Рисунок 2.3 – Шлях, пройдений моноциклом за один цикл

Середньостатистичний водій моноколесу за день долає відстань 50 км, тому для розрахунку повної потужності батареї будемо вважати, що цикл, зображений на рис. 2.3, повториться близько 13 разів.

## 2.2 Обґрунтування вибору потужності електродвигуна та акумуляторної батареї

Для розрахунку зусиль необхідні параметри моноколесу. За основу беремо існуючий аналог InMotion V8 (IM-V8-BK) [3].

Таблиця 2.1 – Параметри для розрахунку сил, що діють на моноколес

Максимальна маса моноколесу (з водієм) $m$ , кг	120
Радіус колеса $r_k$ , м	0.2032
Аеродинамічний коефіцієнт[5] $C_D$	0.3
Фронтальна площа $A_f$ [6], $m^2$	2
Коефіцієнт тертя кочення коліс $f_t$ [7]	0.013
Щільність повітряного потоку $\rho$ [8]	1.225

Розглянемо сили, які діють на моноколес під час руху (рис. 2.4):

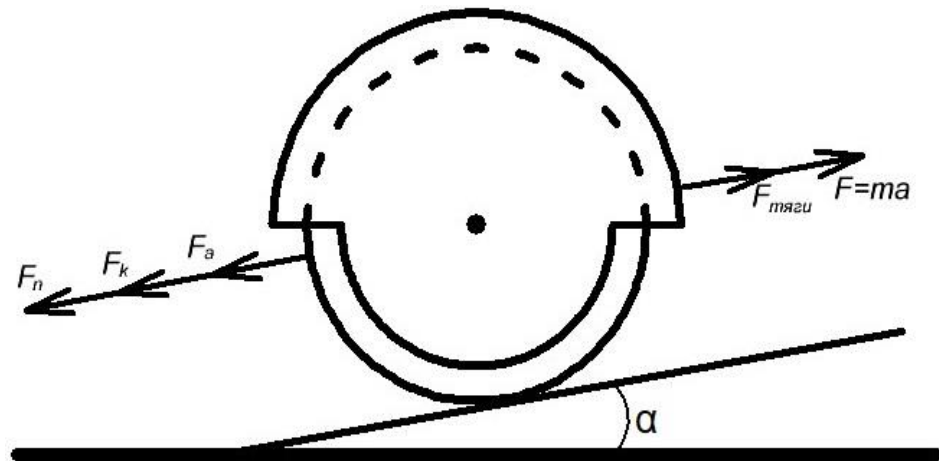


Рисунок 2.4 – Сили, що діють на моноколес під час руху

Для розрахунку сили аеродинамічного опору [9] використовують наступну формулу:

$$F_a = \frac{\rho C_D A_f v^2}{2}, \quad (2.1)$$

де  $\rho$  - щільність повітря (1.225 кг/м<sup>3</sup>, за умов Міжнародної стандартної атмосфери);

$C_D = 0.3$  – аеродинамічний коефіцієнт опору;

$A_f$  - площа лобової поверхні (враховуємо площу моноколесу і водія);

$v$  - швидкість моноколесу (рис. 2.1), м/с.

Тертя кочення визначається притисочною силою і коефіцієнтом тертя кочення:

$$F_k = mgf_m \cos \alpha = mgf_m, \quad (2.2)$$

де  $\alpha$  – кут нахилу поверхні, в нашому випадку  $\alpha=0$ .

Сила підйому розраховується наступним чином:

$$F_n = mg \sin \alpha = 0. \quad (2.3)$$

При визначенні тягового зусилля  $F_t$ , яке має розвивати колесо при русі, було враховано тертя кочення та аеродинамічний опір. Відповідно формула розрахунку тягового зусилля має наступний вигляд:

$$F_t = ma + F_k + F_a. \quad (2.4)$$

Графіки аеродинамічного та тягового зусилля в відповідності до траєкторії руху зображено на рис. 2.5 та рис. 2.6:

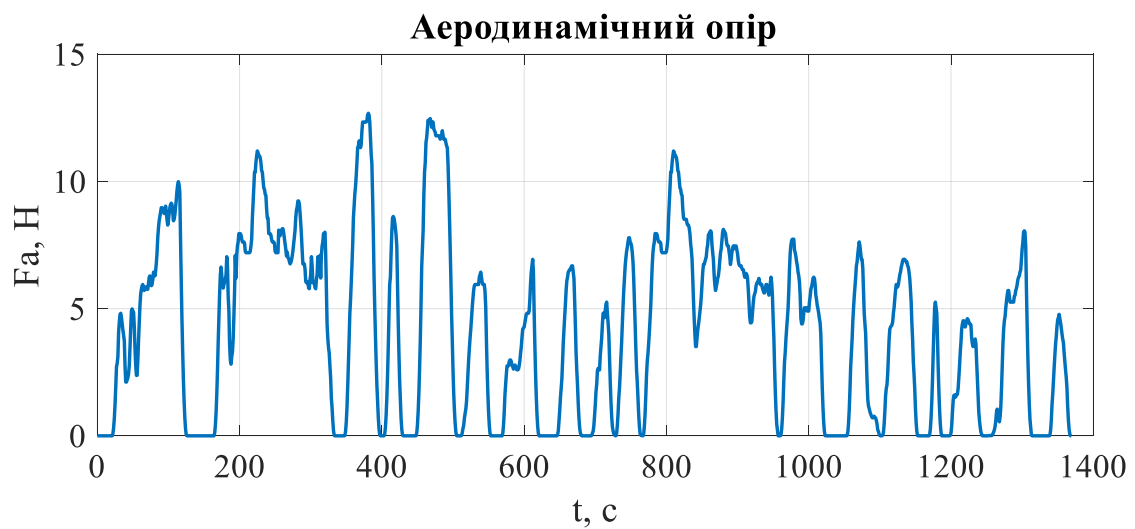


Рисунок 2.5 – Графік аеродинамічного зусилля

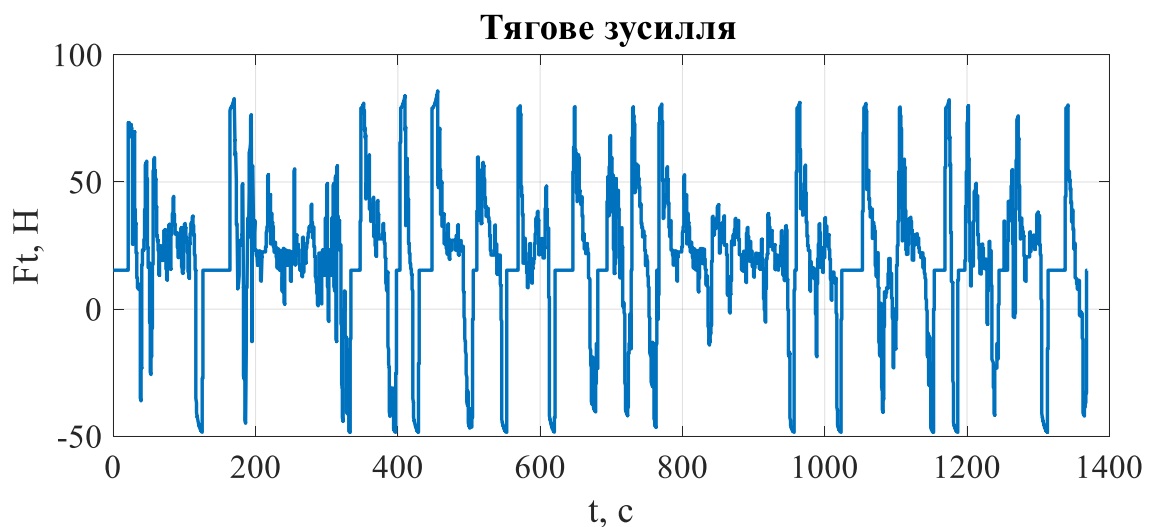


Рисунок 2.6 – Графік тягового зусилля

Для розрахунку сумарного моменту інерції необхідно розрахувати момент інерції колеса і приведений до швидкості колеса момент інерції водія. Так як швидкість залежить від кута нахилу водія, то момент інерції водія залишається незмінним під час руху. Тоді сумарний момент інерції визначається як [10]:

$$J = mr^2 = 120 \cdot 0.2032^2 = 4.955 \text{ кг/м}^2, \quad (2.5)$$

де  $m$  – маса (моноколесу і водія разом);

$r_k$  – радіус колеса.

Для визначення потужності на валу колеса використовується формула

$$P_k = M_k \omega_k = F_t v, \quad (2.6)$$

де  $M_k = F_t r_k$  - момент на валу колеса;

$\omega_k = \frac{v}{r_k}$  - кутова швидкість обертання колеса.

Графіки потужності та моменту на рис. 2.7 та рис. 2.8 відповідно.

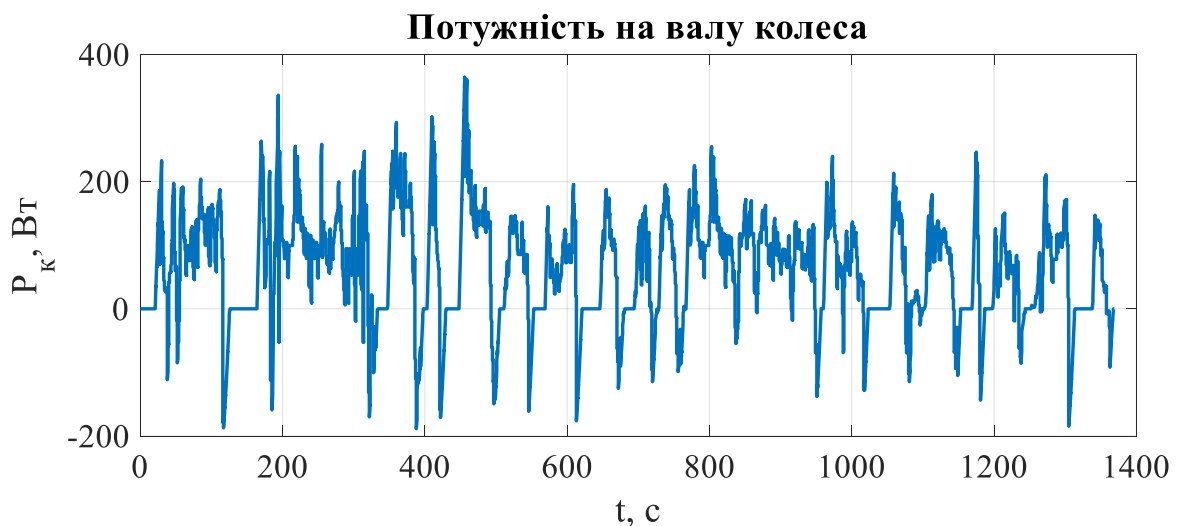


Рисунок 2.7 – Потужність на валу колеса

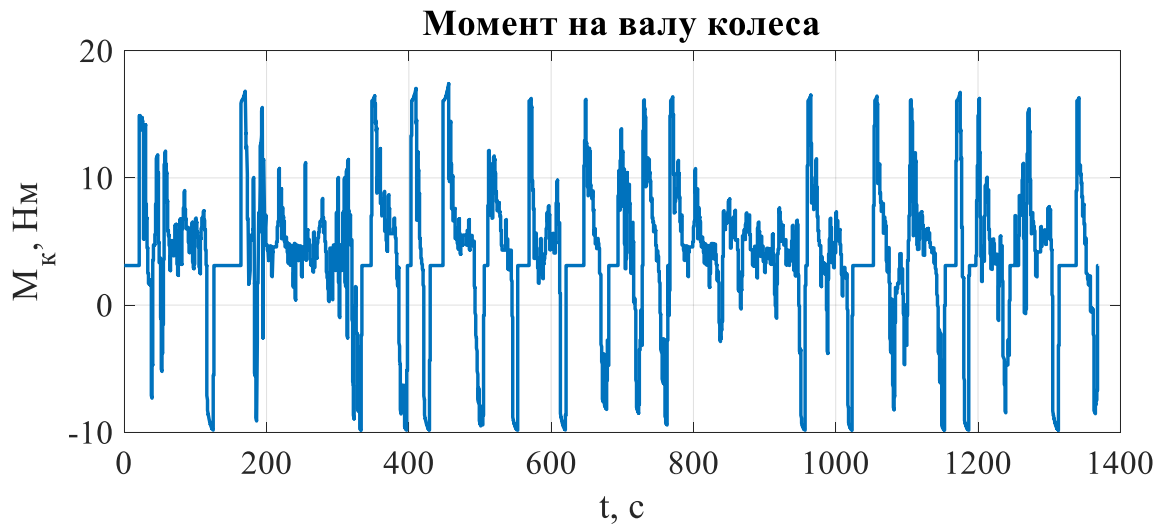


Рисунок 2.8 – Момент на валу колеса

З представлених графіків визначаємо:

$$M_{k.\max} = F_{t.\max} r_k = 85.2252 \cdot 0.2032 = 17.3178 \text{ Нм.} \quad (2.7)$$

При цьому максимальна швидкість обертання колеса:

$$\omega_{k.\max} = \frac{v_{\max}}{r_k} = \frac{8.3}{0.2032} = 40.85 \text{ рад / с,} \quad (2.8)$$

де  $v_{\max} = 30 \frac{\text{км}}{\text{год}} = 8.3 \frac{\text{м}}{\text{с}}$  – максимальна швидкість моноколесу.

Отримавши розраховані значення, знайдемо значення потужності, яку повинен розвивати двигун:

$$P_{\text{дв}} = k_z M_{k.\max} \omega_{k.\max} = 1.3 \cdot 17.3178 \cdot 40.85 = 848.85 \text{ Вт} \quad (2.9)$$

де  $k_z = 1.1 \dots 1.3$  - коефіцієнт запасу (приймаємо  $k_z = 1.3$ ).

Вибрано мотор-колесо «KS-16S» з максимальною потужністю 1200 Вт [11]. Проблема вибору потрібного мотор-колеса в тому, що реальні параметри двигуна знайти доволі складно, тому далі будуть розглядатися параметри схожого двигуна, вони зазначені в табл. 2.2.

Таблиця 2.2 – Параметри мотор-колеса

Найменування	Значення
Номінальна потужність $P$ , Вт	350
Максимальна потужність $P$ , Вт	700
Максимальна швидкість, об/хв	464
Максимальний обертовий момент, Нм	21
Номінальна напруга, В	24
Максимальний ККД, %	87
Максимальна швидкість, км/год	35-40
$I_H/I_{max}$	13.5/29
Маса, кг	2.85

В разі необхідності можна форсувати це мотор-колесо за допомогою контролера і акумулятора на 36 В, що підвищить максимальну швидкість до 40-45 км/год.

Виконаємо перевірку двигуна по нагріву. Для цього знаходимо середньоквадратичні значення моменту та потужності:

$$M_{екв} = k_z \sqrt{\frac{1}{T} \int_0^T M_k^2(t) dt} = 8.2892 \text{ Нм}, \quad (2.10)$$

$$P_{екв} = k_z \sqrt{\frac{1}{T} \int_0^T P_k^2(t) dt} = 127.2529 \text{ Вт}, \quad (2.11)$$

Отримані значення менші тривалої потужності та моменту, тому двигун проходить перевірку по нагріву.

Для розрахунку потужності необхідно використати графік потужності обраного двигуна (рис. 2.9). З рисунку 2.9 видно, що є ділянки з позитивною та негативною потужністю. Ці ділянки відповідають рушійному та гальмівному (рекуперативному) режимам роботи. В рушійному режимі двигун споживає енергію з акумуляторної батареї (АКБ), в гальмівному – генерує в АКБ.

Для розрахунку потужності на клеммах АКБ, яка споживається та генерується на цих ділянках використовують формули:

$$P_{AKB}(t) = \frac{P_{\partial}(t)}{\eta_{\partial}\eta_{inv}} - \text{споживання}, \quad (2.12)$$

$$P_{AKB}(t) = P_{\partial}(t)\eta_{\partial}\eta_{inv} - \text{рекуперація}, \quad (2.13)$$

де  $\eta_{\partial}, \eta_{inv}$  - ККД двигуна і інвертора відповідно.

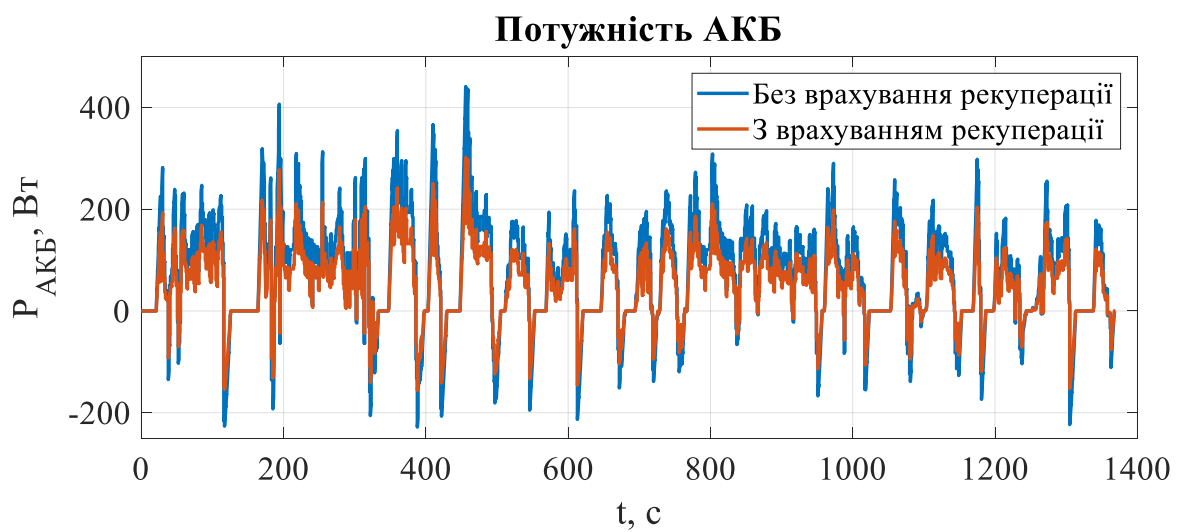


Рисунок 2.9 – Потужність акумуляторної батареї

Щоб вибрати АКБ, необхідно визначити енергію, яку вона має зберігати (рис. 2.10). На основі потужності споживана енергія визначається

$$W_{AKB} = \int_0^T P_{AKB} dt. \quad (2.14)$$

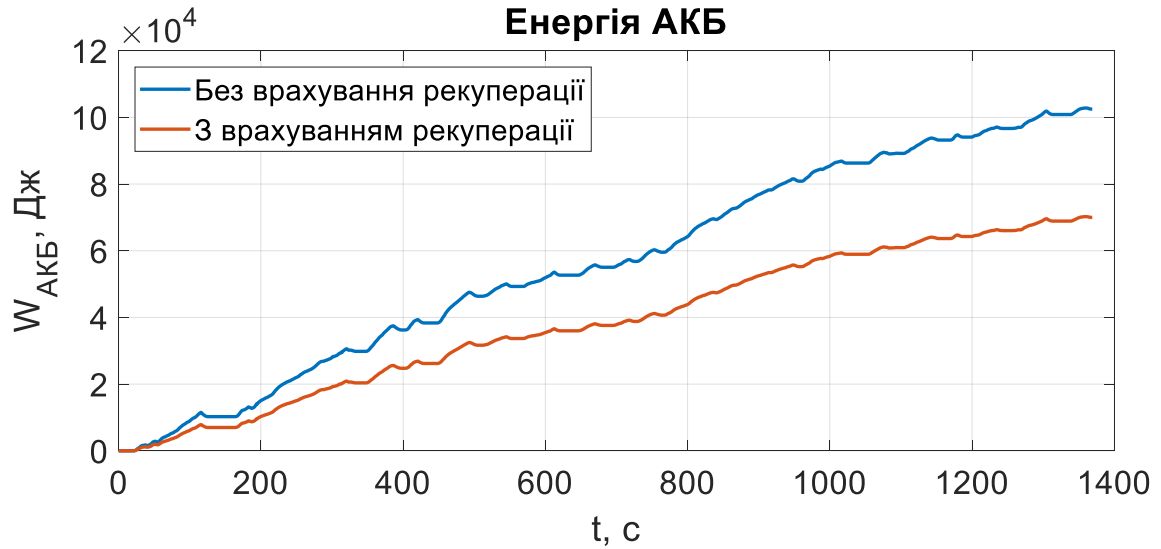


Рисунок 2.10 – Енергія акумуляторної батареї

Виходячи із енергії з врахуванням рекуперації, визначаємо потужність АКБ (рис. 2.11):

$$P_{AKB \text{ год}} = \frac{W_{AKB}}{1 \text{ год}} = \frac{W_{AKB}}{3600 \text{ c}}. \quad (2.15)$$

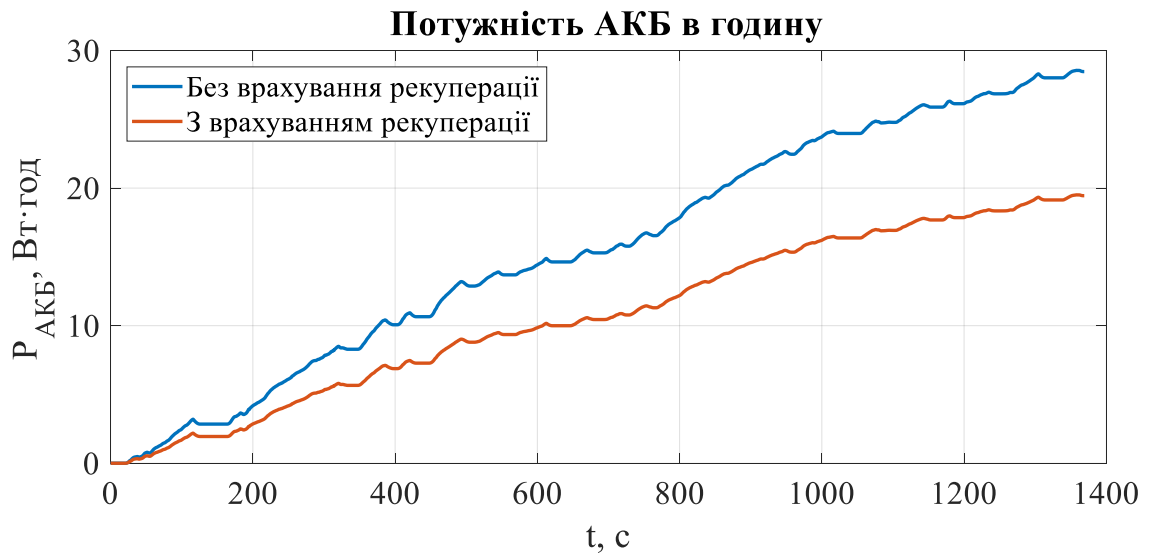


Рисунок 2.11 – Потужність акумуляторної батареї за годину



Так як більшість АКБ складаються із літій-іонних акумуляторів, тому обираємо акумулятори «Rablex 18650 3200Mah Li-Ion 3.7V Original» [12], основні параметри яких наведено в таблиці 2.3.

Таблиця 2.3 – Параметри АКБ

Найменування	Значення
Напруга (номінальна), В	3.7
Робоча напруга, В	2.75-4.2
Ємність (заявлена виробником), мА·год	3200
Ресурс (циклів)	500
Тип батареї	Li-Ion
Вага, г	44
Розміри, мм	70x18

Для розрахунку кількості АКБ в одній паралельній гілці ділимо напругу ланки постійного струму на напругу однієї батареї:

$$N_{\text{гілки}} = \frac{U_{dc}}{U_{\text{АКБ1}}} = \frac{24}{3.7} = 6.4865 \approx 7 \quad (2.16)$$

Сумарний струм, який має видавати АКБ, визначається

$$I_{\text{АКБ}\Sigma} = \frac{\frac{S_{\text{шляху}}}{S_{\text{циклу}}} P_{\text{АКБ}}}{U_{dc}} = \frac{\frac{50}{3.9} \cdot 20}{24} = 10.68 \text{ А.} \quad (2.17)$$

де  $S_{\text{шляху}}$  – відстань, яку має проїхати моноколес на одному повному заряді АКБ;

$S_{\text{циклу}}$  – шлях, пройдений моноколесом за один цикл руху (рис. 2.3).

Визначаємо кількість паралельних гілок для забезпечення сумарного струму

$$N_{\text{гілок}} = \frac{I_{AKB\Sigma}}{I_{AKB1}} = \frac{10.68}{3.2} = 3.34 \approx 4 \quad (2.18)$$

Розраховуємо сумарну кількість акумуляторів

$$N_{\Sigma} = N_{\text{гілки}} N_{\text{гілок}} = 7 \cdot 4 = 28. \quad (2.19)$$

Далі знаходимо площу АКБ та масу:

$$\begin{aligned} m_{AKB\Sigma} &= N_{\Sigma} m_{AKB} = 28 \cdot 44 = 1232 \text{ г} \\ S_{AKB\Sigma} &= N_{\Sigma} l_1 l_2 = 28 \cdot 0.07 \cdot 0.018 = 0.1147 \text{ м}^2 \end{aligned} \quad (2.20)$$

### 2.3 Розрахунок та вибір силового обладнання

Для розрахунку основних елементів перетворювача необхідно знати параметри двигуна, у нашому випадку це мотор-колесо (таблиця 2.2).

Амплітудні значення напруги і струму статора:

$$U_{na} = 24 \text{ В}, \quad I_{na} = 13.5 \text{ А}.$$

Максимальне значення струму двигуна також вказано в паспортних даних:

$$I_{\max} = 29 \text{ А}. \quad (2.21)$$

При живленні перетворювача від джерела напруги  $U_{grid} = 24 \text{ В}$ , напруга в ланці постійного струму при ідеальній фільтрації також буде  $U_{dcn} = U_{grid} = 24 \text{ В}$ , а струм в ланці  $I_{dcn} = I_{na} = 13.5 \text{ А}$ .

При виборі транзисторів для автономних інверторів напруги їх максимально допустима напруга стік-витік (drain-source) має бути більшою за

максимально допустиму напругу ланки постійного струму, тобто  $U_{ds} = 1.5U_{dcn} = 36 \text{ В}$ .

Аналогічні вимоги і до струму, тому  $I_{ds} = 1.5I_{dcn} = 20.25 \text{ А}$ .

Для побудови силової частини перетворювача обрано mosfet-транзистори, так як на них втрати енергії будуть мінімальними.

В результаті для автономного інвертора напруги вибрано транзистори типу IRF540 [13], максимальна напруга якого становить 55 В, а максимальний струм – 33 А.

Розрахуємо значення струму ланки постійного струму з врахуванням перевантаження

$$I_{dc} = \frac{k_{ovl} P_n}{U_{dcn} \eta_{\delta} \eta_{інв}} = \frac{2 \cdot 350}{24 \cdot 0.87 \cdot 0.95} = 35.29 \text{ А}, \quad (2.22)$$

де  $\eta_{інв} = (0.95 \dots 0.97)$  – коефіцієнт ККД інвертора напруги, прийmemo  $\eta_{інв} = 0.95$ ;

$k_{ovl}$  – перевантажувальна здатність, яка, в загальному випадку, рівна (2...3) – для високо динамічних застосувань, таких як приводи подачі, тощо. Приймаємо  $k_{ovl} = 2$ .

Ємність ланки постійного струму визначається за формулою:

$$C = \frac{I_{dc}}{\Delta U f_{sw}} = \frac{35.29}{2 \cdot 10000} = 1764.5 \text{ мкФ}, \quad (2.23)$$

де  $f_{sw} = 10 \text{ кГц}$  – частота комутації ключів;

$\Delta U = 2 \text{ В}$  – пульсації напруги.

Щоб отримати необхідну ємність, використаємо два конденсатори «К50-35» [14] ємністю  $C_1 = C_2 = 1000 \text{ мкФ}$  і увімкнемо їх послідовно, тоді результуюча ємність буде  $C_{\Sigma} = C_1 + C_2 = 2000 \text{ мкФ}$ .

Для організації зворотних зв'язків за струмом необхідно ввести в схему відповідні датчики. Їх здійснюють за максимальним струмом, який необхідно вимірювати, тобто щоб вимірюваний діапазон датчика був більшим від максимального струму в системі. Для вимірювання струмів на виході перетворювача для вибору використовують значення максимального струму (2.22).

Оскільки в розглянутому прикладі цього достатньо, то по розрахованому значенню  $I_{dc}$  з каталогу [15] вибираємо датчики типу ACS712, діапазон вимірювання 100 А.

## 2.4 Розробка і опис схеми силової частини і схеми керування

При розробці схеми керування СДПМ використовувалася стаття [16].

Щоб керувати трифазним двигуном за допомогою синусоїдальних струмів, для кожної фази потрібно генерувати незалежні напруги. Це робиться за допомогою трьох напівмостів, по одному на кожен вивод двигуна. Кожен напівміст складається з двох ключів (верхній і нижній).

Для генерації певного рівня напруги використовується ШІМ (PWM). Залежно від скважності ШІМ можна керувати середньою напругою  $V_{out}$ . Але форма цього сигналу буде повторювати сигнал ШІМ, тобто буде прямокутним як ШІМ сигнал, а не гладким. Якщо цю напругу подати на низькочастотний фільтр, то на виході фільтра буде напруга, пропорційна коефіцієнту заповнення ШІМ верхнього ключа. Керуючи скважністю ШІМ можна отримати довільну форму напруги на виході, в тому числі і необхідну напругу синусоїдальної форми (рис. 2.12).

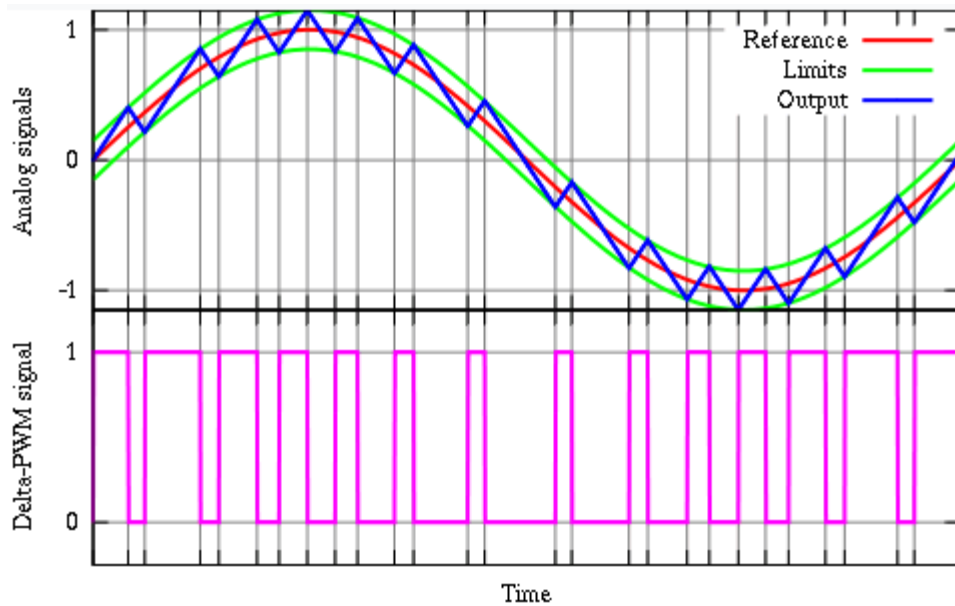


Рисунок 2.12 – Широтно-імпульсна модуляція

З різних причин, в пристроях керування двигуном не використовують фільтри низьких частот. По-перше, сам двигун діє як фільтр. Індуктивність і опір обмоток двигуна створюють RL фільтр. До того ж механічна інерція двигуна і навантаження створюють "механічний" фільтр низьких частот. При досить великій частоті PWM, флуктуації (випадкові відхилення від середнього значення або середньоквадратична похибка) швидкості ротора будуть абсолютно незначні. По-друге, протікання струму через низькочастотний фільтр, наприклад RC-фільтр, призвело б до суттєвих втрат потужності на самому фільтрі.

Оскільки при генерації PWM використовується комплементарна робота двох ключів (верхнього і нижнього), потрібно врахувати той факт, що для закриття ключа потрібен якийсь час. Тому потрібно вимикати відкритий ключ, і через деякий час, необхідний для його закриття, відкривати комплементарний ключ. Цей час описує термін «dead time». У мікроконтролера STM32 це питання вирішує таймер, який займається генерацією PWM.

Знаючи положення ротора в даний момент часу, можна математичним шляхом обчислювати амплітуду (скважність PWM) для кожної з трьох фаз. Однак математичні обчислення займають багато часу і тому чинять інакше.

Можна створити таблицю, в яку помістити заздалегідь обчислену синусоїду для всіх трьох фаз.

Генерувати синусоїди для кожної фази - це прямий метод, але є більш ефективний спосіб. Ми будемо генерувати три синусоїдальних напруги із зсувом фази на 120 градусів між ними. Табл. 2.5 – 2.6 показують, як цього можна досягти, не роблячи повні синусоїди для кожного виводу двигуна.

Таблиця 2.5 – Кінцеві і лінійні напруги, рух вперед

Step	U	V	W	U-V	V-W	W-U
S1-S2	$\sin(\theta)$	0	$-\sin(\theta-120)$	$\sin(\theta)$	$\sin(\theta-120)$	$\sin(\theta-240)$
S3-S4	$-\sin(\theta-240)$	$\sin(\theta-240)$	0	$\sin(\theta)$	$\sin(\theta-120)$	$\sin(\theta-240)$
S5-S6	0	$-\sin(\theta)$	$\sin(\theta-240)$	$\sin(\theta)$	$\sin(\theta-120)$	$\sin(\theta-240)$

Таблиця 2.6 – Кінцеві і лінійні напруги, рух назад

Step	U	V	W	U-V	V-W	W-U
S1-S2	$\sin(\theta)$	$-\sin(\theta-120)$	0	$-\sin(\theta-240)$	$-\sin(\theta-120)$	$-\sin(\theta)$
S3-S4	$-\sin(\theta-240)$	0	$\sin(\theta-120)$	$-\sin(\theta-240)$	$-\sin(\theta-120)$	$-\sin(\theta)$
S5-S6	0	$\sin(\theta-240)$	$-\sin(\theta)$	$-\sin(\theta-240)$	$-\sin(\theta-120)$	$-\sin(\theta)$

На рис. 2.13 наочно зображений цей алгоритм в порівнянні з блочною комутацією (як для BLDC).

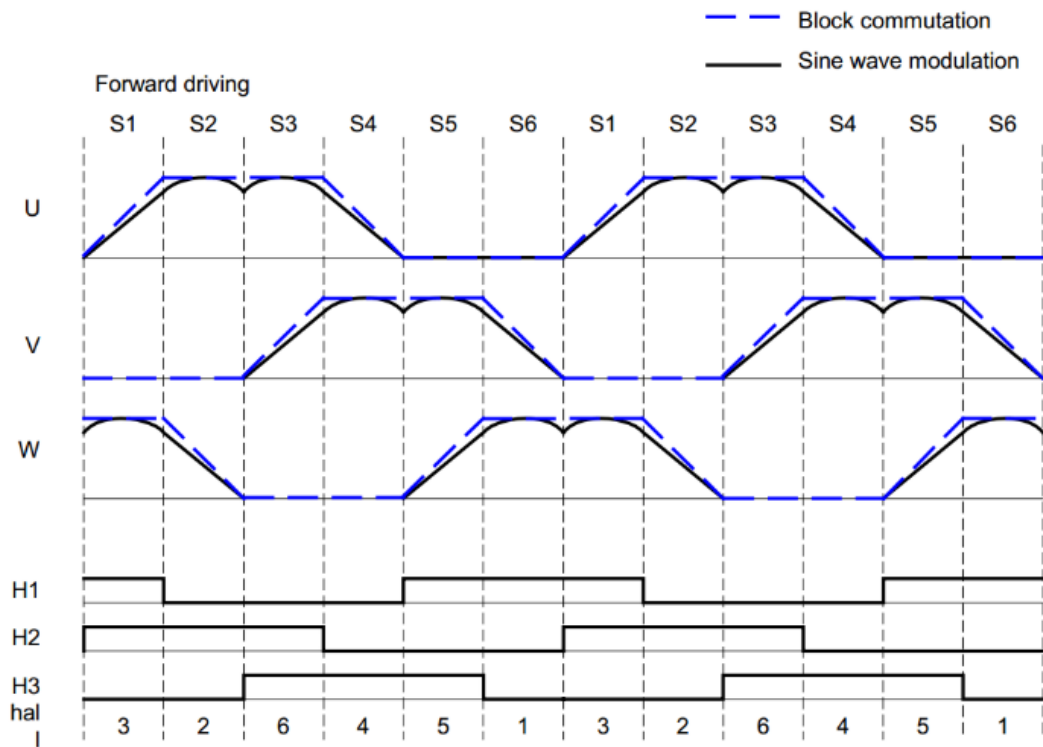


Рисунок 2.13 – Генерація синусоїди за допомогою ШІМ

У такого підходу є дві переваги:

а) максимальна амплітуда згенерованої напруги вище, ніж при генерації чистої синусоїди на кожній фазі. Як наслідок, отримаємо більш високий крутний момент і швидкість.

б) кожен вивід двигуна третину часу підключений до землі, що скорочує втрати потужності на комутацію.

При створенні таблиці значень потрібно знайти компроміс між обсягом таблиці і швидкістю роботи з нею. Оскільки форма хвилі однакова для всіх трьох фаз, і відрізняється тільки зрушенням на 120 градусів, то можна зберігати в таблиці тільки одну хвилю і використовувати її для розрахунку всіх трьох вихідних напруг. Однак, для скорочення часу на обчислення, прийнято рішення зберегти в таблиці значення для всіх трьох фаз. Так забезпечується найбільш швидка обробка даних. Використовувана в прикладі таблиця «синусів» має 192 значення для кожної фази. Тобто генерований сигнал буде розбитий на секції і мати ступінчасту форму.

Таким чином, за один електричний оборот нам доведеться 192 рази обчислювати і змінювати значення PWM для кожної з трьох фаз (рис. 2.14).

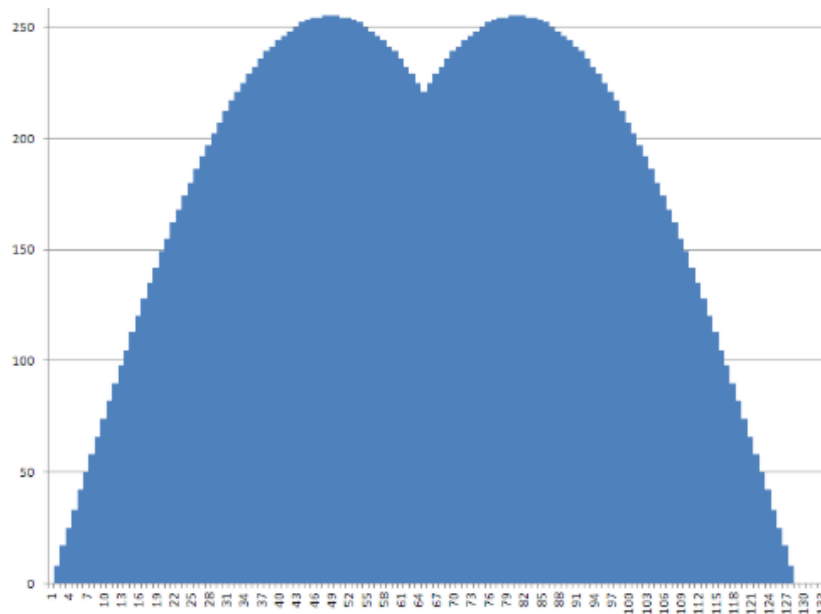


Рисунок 2.14 – Згенерований сигнал

При розрахунку PWM для кожної фази потрібно враховувати і величину потужності, що подається на двигун. При реалізації замкнутої системи, наприклад, зі стабілізацією обертів, цей показник буде обчислюватися математично з урахуванням поточного навантаження, необхідних обертів і параметрів ПД регулятора.

Було б ідеально, якщо датчик положення ротора міг будь-якої миті інформувати мікроконтролер про поточний стан ротора. Три датчика Холла змінюють свій стан 6 раз за один електричний оборот. Між цими подіями положення ротора достовірно не відомо. Для визначення поточного положення ротора між спрацьовуваннями датчиків, доведеться робити обчислення з урахуванням поточної швидкості обертання ротора.

Для того щоб генерувати напругу, що подається на виводи двигуна потрібної частоти, потрібно знати швидкість обертання двигуна. Для визначення швидкості обертання використовуються датчики положення. Обчислюючи час



між двома перемиканнями датчиків положення (будемо вважати, що швидкість обертання ротора за цей час не змінюється), можна розрахувати частоту, з якою нам треба генерувати вихідний сигнал. Таким чином, знаючи швидкість, ми можемо обчислити положення ротора в будь-який момент часу. Швидкість обертання ротора може змінюватися, тому потрібно періодично підлаштовувати вихідну напругу відповідно до реального стану ротора. Таке налаштування називається «фазове налаштування».

Найзручніше фазове налаштування виконувати, коли контролер точно знає реальне положення ротора. У випадку з трьома датчиками Холла – це момент зміни стану датчиків. Саме в цей момент можна дізнатися точний кут ротора. Підлаштування виконується шляхом корекції індексу положення в таблиці вихідних значень (в таблиці синусів).

У момент старту швидкість ротора невідома доти, поки не відбудуться послідовно дві зміни стану датчиків Холла. Щоб забезпечити нормальний запуск двигуна, поки швидкість обертання ротора не відома, застосовується блокова комутація. Тобто контролер управляє двигуном як (BLDC). Карта напруг, яка використовується в режимі блокової комутації, показана на рис. 2.13 (штрихова синя лінія) на тлі з синусоїдальною напругою. Як тільки швидкість обертання ротора буде визначена, відбувається перехід до синусоїдального збудження.

Розроблена схема зображена на рис. 2.15.

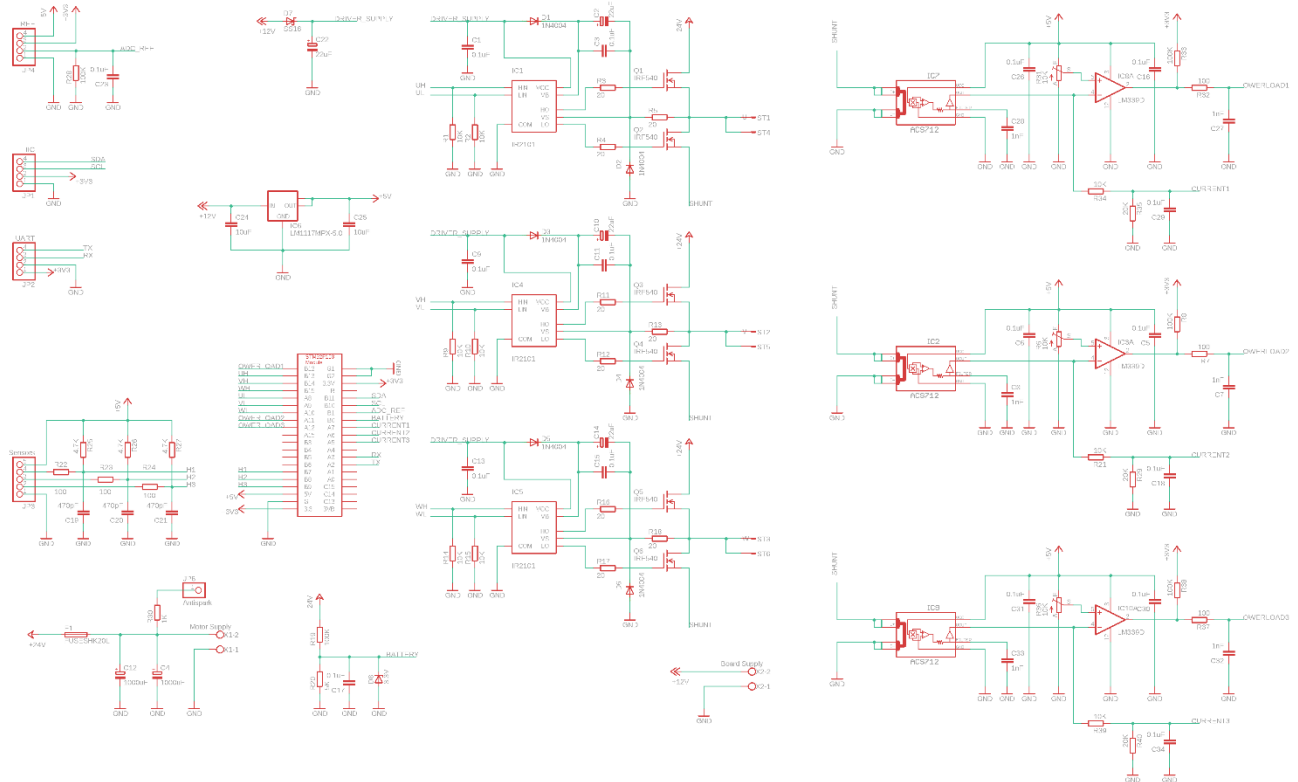


Рисунок 2.15 – Схема електрична керування

## Висновки до розділу 2

1. Використовуючи отримані дані, а також враховуючи висунуті раніше вимоги, розраховано сили, що впливають на моноколес під час руху, на основі яких отримано графіки зміни моменту та потужності двигуна під час руху.
2. По розрахованих значеннях потужності та моменту вибрано спеціальний двигун – мотор-колесо. Подібні двигуни розроблені для використання в таких пристроях, як моноколесо, електровелосипед, електросамокат, тощо.
3. Розраховано потужність акумуляторної батареї, що забезпечить живлення двигуна на весь цикл руху без підзарядки.
4. Для живлення трифазного двигуна від джерела напруги (джерела постійного струму) потрібний автономний інвертор напруги (АІН). Для цього розглянуто принцип роботи АІН та розроблено схему електричну.
5. Знаючи напругу і струм живлення двигуна, розраховано та обрано основні елементи силової схеми пристрою, зокрема силові ключі, драйвери,

давачі струму статора та напруги ланки постійного струму. Давач Холла вбудований в мотор-колесо, тому не потребує вибору.

### **3 ПРОЕКТУВАННЯ СИСТЕМИ КЕРУВАННЯ**

В ході розробки електромеханічної системи електротранспорту (як і будь якої системи з двигуном) необхідно спроектувати систему керування двигуном.

Опираючись на вимоги до двигуна і системи керування в цілому, а також особливості розроблюваного пристрою, визначаються цілі керування. На основі цього можна визначитися з методом керування і синтезувати систему керування.

#### **3.1 Обґрунтування принципу керування електроприводом**

Вибір принципу керування двигуна в першу чергу залежить від вимог, що ставляться до системи, такі як діапазон регулювання швидкості, величини прискорення та ривка, тощо.

В моноколесі керування відбувається за рахунок нахилу водія (тіла водія) в бік руху. По суті, це момент навантаження. Таким чином, керування схоже на автомобіль, де нахил водія аналогічний натисканню педалі газу.

Таким чином, було вирішено обрати векторне керування двигуном при живленні від джерела напруги, так як АКБ – джерело напруги.

Векторне в цьому випадку незамінне, так як від системи вимагається максимальна точність керування, в першу чергу для зручності використання пристрою, а отже і безпеки водія під час руху.

#### **3.2 Структурно-параметричний синтез системи керування**

Розглядаючи рівняння динаміки узагальненої електричної машини, зрозуміло, що рівняння руху механічної частини різних електричних машин однакові відносно електромагнітного моменту. Тому асимптотичне відпрацювання моменту є необхідною умовою асимптотичного керування кутовою швидкістю та кутовим положенням. Специфіка конкретного типу електричної машини як раз і полягає в керуванні її моментом і модулем вектора потокозчеплення. У деяких технологічних використаннях асимптотичне

відпрацювання моменту, як заданої функції часу незалежно від змін механічних координат, є самостійною задачею. При цьому зміна механічних координат електричної машини зумовлюється технологічними процесами функціонування установки. Така задача виникає, наприклад, в системах регулювання натягу, в процесах намотування – розмотування; під час виконання операцій, що пов'язані зі взаємодією із зовнішнім середовищем, в електроприводах екскаваторів, гірничих машин, промислових роботів, у металообробці; різноманітних навантажувальних агрегатах, тощо.

У загальному випадку задача асимптотичного відпрацювання моменту та модуля потокозчеплення формулюється наступним чином. Нехай  $M^*(t)$  та  $\psi^*(t)$  є задані безперервні функції моменту та модуля потокозчеплення. Передбачається, що  $M^*(t)$  – обмежена функція часу, що має обмежену відому першу похідну за часом, а  $\psi^*(t) > 0$  – строго додатна функція часу з обмеженими першою або першою та другою похідними за часом. Кутова швидкість при керуванні моментом розглядається, як обмежена зовнішня функція часу. Розглянемо клас електричних машин, що описуються загальною моделлю:

$$\begin{aligned}\dot{\theta} &= \omega, \\ \dot{\omega} &= \frac{1}{J} [M - v\omega - M_c(t)], \quad M = \frac{1}{2} \bar{i}^T \frac{\partial \bar{L}(\theta)}{\partial \theta} \bar{i}, \\ \dot{\bar{i}} &= \bar{L}^{-1}(\theta) \left( -\bar{R} \bar{i} - \omega \frac{\partial \bar{L}(\theta)}{\partial \theta} \bar{i} + \bar{u} \right),\end{aligned}\tag{3.1}$$

де  $\theta, \omega$  – кутове положення та кутова швидкість ротора відносно статора;

$v > 0$  – коефіцієнт в'язкого тертя;

$M$  – момент двигуна;

$M_c$  – момент навантаження;

$\bar{u} = (\bar{u}_1^T, \bar{u}_2^T)^T$  – вектор керуючих напруг;

$\bar{i} = (\bar{i}_1^T, \bar{i}_2^T)^T$  – вектор струмів;

$\bar{L}(\theta) \in R^{4 \times 4}$  – матриця індуктивностей;

$\bar{R} = \text{diag}(R_1, R_1, R_2, R_2)$  – матриця активних опорів машини;

$R_1, R_2$  – активні опори статора і ротора.

Необхідно сконструювати вектор керуючих дій  $\bar{u}$ , як функцію вимірюваних змінних стану та задаючих дій  $M^*(t)$  та  $\psi^*(t)$ , що гарантує системі:

а) глобальне асимптотичне відпрацювання моменту та модуля вектора потокозчеплення, тобто

$$\lim_{t \rightarrow \infty} \tilde{M} = 0, \quad \lim_{t \rightarrow \infty} \tilde{\psi} = 0, \quad (3.2)$$

де похибки відпрацювання моменту та потокозчеплення визначені як

$$\tilde{M} = M - M^*, \quad \tilde{\psi} = \psi - \psi^*; \quad (3.3)$$

б) розв'язаність процесів керування моментом та потокозчепленням, тобто з умови  $\tilde{\psi}(0) = 0$  слідує  $\tilde{\psi}(t) = 0 \forall t \geq 0$ .

Додатково вимагається гарантувати невиродженість алгоритму керування та обмеженість усіх його внутрішніх змінних.

### 3.2.1 Постановка задачі

Динамічна модель неявнополюсного синхронного двигуна зі збудженням від постійних магнітів (СДПМ), за умов симетричного живлення, визначається рівняннями [14]:

$$\begin{aligned}
\dot{\theta} &= \omega, \\
\dot{\omega} &= \frac{1}{J}(M - M_c), \\
\dot{i}_{1a} &= L_1^{-1}(-R_1 i_{1a} + L_m i_f p_n \omega \sin p_n \theta + u_{1a}), \\
\dot{i}_{1b} &= L_1^{-1}(-R_1 i_{1b} - L_m i_f p_n \omega \cos p_n \theta + u_{1b}),
\end{aligned} \tag{3.4}$$

де  $(i_{1a}, i_{1b})^T$  – компоненти вектору струму статора;

$M = k_m(-i_{1a} \sin p_n \theta + i_{1b} \cos p_n \theta)$  – момент двигуна;

$M_c$  – момент навантаження;

$\theta, \omega$  – кутове положення та кутова швидкість ротора відносно статора;

$i_f = \text{const}$  – сталі значення, що схематизує дію постійних магнітів;

$L_m$  – індуктивність контуру намагнічування;

$L_1$  – індуктивність статорної обмотки;

$R_1$  – активний опір статора;

$k_m$  – коефіцієнт моменту;

$p_n$  – число пар полюсів.

Необхідно синтезувати алгоритм векторного керування моментом СД, який формує двомірний вектор напруг  $\bar{u} = (\bar{u}_1^T, \bar{u}_2^T)^T$  так, що будуть досягнуті цілі керування моментом (3.2).

### 3.2.2 Розв'язання задачі

Щоб позбавитися залежності рівнянь динаміки СДПМ від кутового положення, за допомогою перетворення Парк–Горєва [17] «перейдемо» в систему координат (d-q):

$$\begin{aligned}
\begin{pmatrix} i_{1d} \\ i_{1q} \end{pmatrix} &= e^{-j p_n \theta} \begin{pmatrix} i_{1a} \\ i_{1b} \end{pmatrix}, \\
\begin{pmatrix} u_{1d} \\ u_{1q} \end{pmatrix} &= e^{-j p_n \theta} \begin{pmatrix} u_{1a} \\ u_{1b} \end{pmatrix},
\end{aligned} \tag{3.5}$$

де 
$$e^{j p_n \theta} = \begin{bmatrix} \cos p_n \theta & -\sin p_n \theta \\ \sin p_n \theta & \cos p_n \theta \end{bmatrix}, e^{-j p_n \theta} = \begin{bmatrix} \cos p_n \theta & \sin p_n \theta \\ -\sin p_n \theta & \cos p_n \theta \end{bmatrix}.$$

Після перетворень отримуємо модель СДПМ у наступній формі:

$$\begin{aligned} \dot{\theta} &= \omega, \\ \dot{\omega} &= \frac{1}{J} (M - M_c), M = k_m i_{1q}, \\ \dot{i}_{1d} &= -\frac{R_1}{L_1} i_{1d} + p_n \omega i_{1q} + \frac{1}{L_1} u_{1d}, \\ \dot{i}_{1q} &= -\frac{R_1}{L_1} i_{1q} - p_n \omega i_{1d} - \frac{L_m}{L_1} i_f p_n \omega + \frac{1}{L_1} u_{1q}, \end{aligned} \quad (3.6)$$

де  $i_{1d}, i_{1q}$  – струми статора в системі координат (d-q);

$u_{1d}, u_{1q}$  – струми статора в системі координат (d-q).

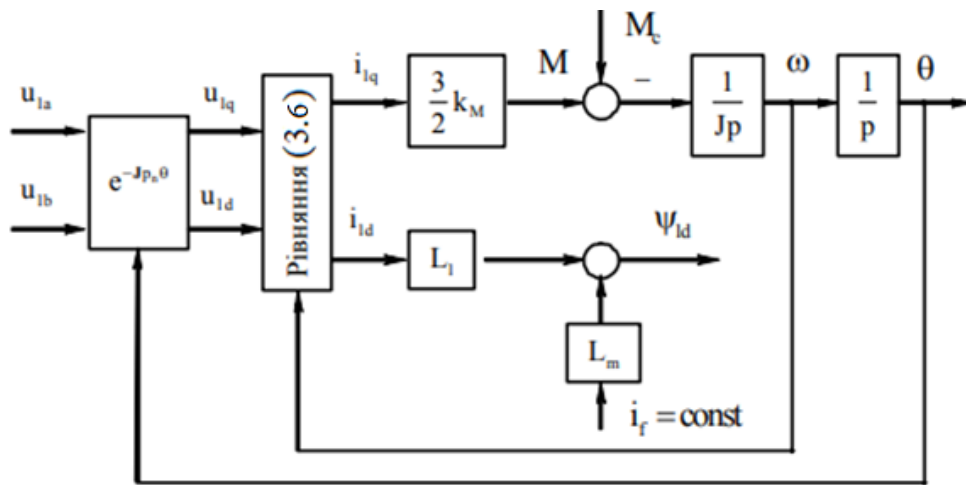


Рисунок 3.1 – Структурна схема СДПМ

Сформувавши вектор напруг статора у вигляді

$$\begin{pmatrix} u_{1d} \\ u_{1q} \end{pmatrix} = \begin{pmatrix} -L_1 p_n \omega i_{1q} + v_d \\ L_1 p_n \omega i_{1d} + v_q \end{pmatrix}, \quad (3.7)$$

отримаємо



$$\begin{aligned}
\dot{\theta} &= \omega, \\
\dot{\omega} &= \frac{1}{J}(M - M_c), M = k_m i_{1q}, \\
\dot{i}_{1d} &= -\frac{R_1}{L_1} i_{1d} + \frac{1}{L_1} v_d, \\
\dot{i}_{1q} &= -\frac{R_1}{L_1} i_{1q} - \frac{L_m}{L_1} i_f p_n \omega + \frac{1}{L_1} v_q.
\end{aligned} \tag{3.8}$$

Отримані рівняння динаміки СДПМ є лінійними.

Алгоритм керування, що перетворює нелінійну модель СДПМ до лінійної повністю керованої форми, задається перетвореннями координат (3.2) та вектором напруг статора (3.4) і має вигляд

$$\begin{aligned}
\begin{pmatrix} u_{1a} \\ u_{1b} \end{pmatrix} &= e^{J p_n \theta} \begin{pmatrix} -L_1 p_n \omega i_{1q} + v_d \\ L_1 p_n \omega i_{1d} + v_q \end{pmatrix}, \\
\begin{pmatrix} i_{1d} \\ i_{1q} \end{pmatrix} &= e^{-J p_n \theta} \begin{pmatrix} i_{1a} \\ i_{1b} \end{pmatrix}.
\end{aligned} \tag{3.9}$$

Цей алгоритм класифікується як лінеаризуючий зворотнім зв'язком, а у відповідності з принципом формування керуючих дій (3.7) в системі координат ротора (d-q), називається векторним керуванням в координатах ротора. Структурна схема лінеаризуючого зворотнім зв'язком регулятора показана на рис. 3.2.

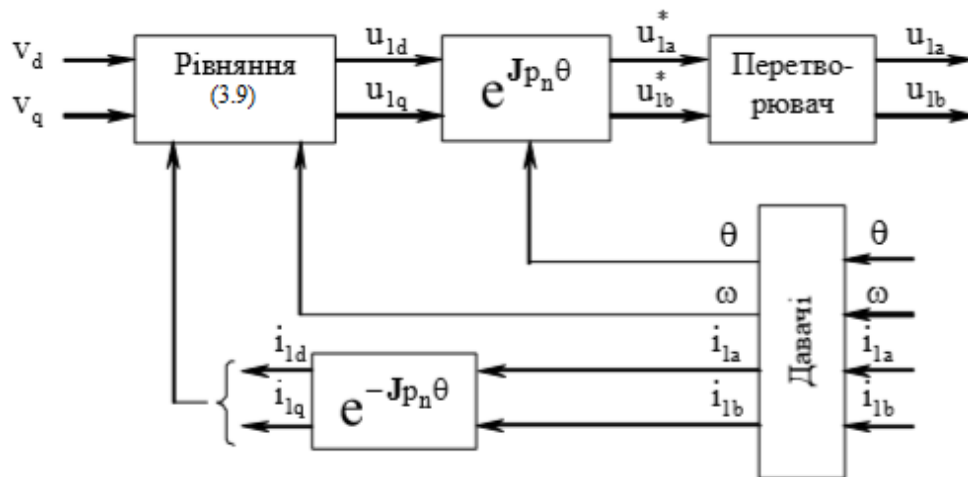


Рисунок 3.2 – Структурна схема лінеаризуючого регулятора СД зі збудженням від постійних магнітів

Похибки відпрацювання цих струмів визначаються так:

$$\begin{pmatrix} \tilde{i}_{1d} \\ \tilde{i}_{1q} \end{pmatrix} = \begin{pmatrix} i_{1d} - i_{1d}^* \\ i_{1q} - i_{1q}^* \end{pmatrix}. \quad (3.10)$$

Для проектування алгоритму відпрацювання моменту шляхом керування струмом  $i_{1q}$  керуюча дія  $v_q$  формується як вихід регулятора струму

$$\begin{aligned} v_q &= R_1 i_{1q}^* + L_m i_f p_n \omega + L_1 (\dot{i}_{1q} - k_{1q1} \tilde{i}_{1q} - x_q), \\ \dot{x}_q &= k_{i1q} \tilde{i}_{1q}, \end{aligned} \quad (3.11)$$

де  $x_q$  - інтегральна складова по осі (q);

$(k_{i1q}, k_{i1q}) > 0$  – коефіцієнти пропорційної та інтегральної складових регулятора струму.

Керуюча дія регулятора струму  $v_d$  має вигляд

$$\begin{aligned} v_d &= R_1 i_{1d}^* + L_1 (\dot{i}_{1d} - k_{i1d} \tilde{i}_{1d} - x_d), \\ \dot{x}_d &= k_{i1d} \tilde{i}_{1d}, \end{aligned} \quad (3.12)$$

де  $x_d$  - інтегральна складова по осі (d);

$(k_{id1}, k_{iid}) > 0$  – коефіцієнти пропорційної та інтегральної складових регулятора струму.

Підставивши отримані рівняння керуючих дій (3.8, 3.9) в рівняння динаміки СДПМ (3.5), отримуємо наступні рівняння похибок відпрацювання:

$$\begin{aligned}\dot{\tilde{i}}_{1d} &= -k_{1d}\tilde{i}_{1d} - x_d, \\ \dot{x}_d &= k_{iid}\tilde{i}_{1d}, \\ \dot{\tilde{i}}_{1q} &= -k_{1q}\tilde{i}_{1q} - x_q, \\ \dot{x}_q &= k_{iiq}\tilde{i}_{1q},\end{aligned}\tag{3.13}$$

де  $k_{id} = \frac{R_1}{L_1} + k_{id1};$

$$k_{iq} = \frac{R_1}{L_1} + k_{iq1}.$$

Повний алгоритм відпрацювання моменту містить:

а) регулятор моменту

$$i_{1q}^* = \frac{M^*}{k_m}, k_m = \frac{3}{2} L_m i_f p_n;$$

б) регулятор потокозчеплення

$$i_{1d}^* = L_1^{-1}(\psi_1^* - L_m i_f), \quad \psi_1^* = L_m i_f \quad \text{при } |\omega| < \omega_n;$$

в) регулятори струму

по осі d

$$\begin{aligned}u_{1d} &= R_1 i_{1d}^* + L_1 \left( -p_n \omega i_{1q} + \dot{i}_{1d}^* - k_{id1} \tilde{i}_{1d} - x_d \right), \\ \dot{x}_d &= k_{iid} \tilde{i}_{1d}\end{aligned}$$

та осі q

$$u_{1q} = R_1 i_{1q}^* + L_m i_f p_n \omega + L_1 (p_n \omega i_{1d} + i_{1d}^* - k_{iq1} \tilde{i}_{1q} - x_q),$$

$$\dot{x}_q = k_{iiq} \tilde{i}_{1q}.$$

На рис. 3.3 представлена структурна схема системи відпрацювання моменту, яка вміщує лінеаризуючий регулятор (3.9), регулятори поточзчеплення та моменту, а також регулятори струмів (3.11),(3.12).

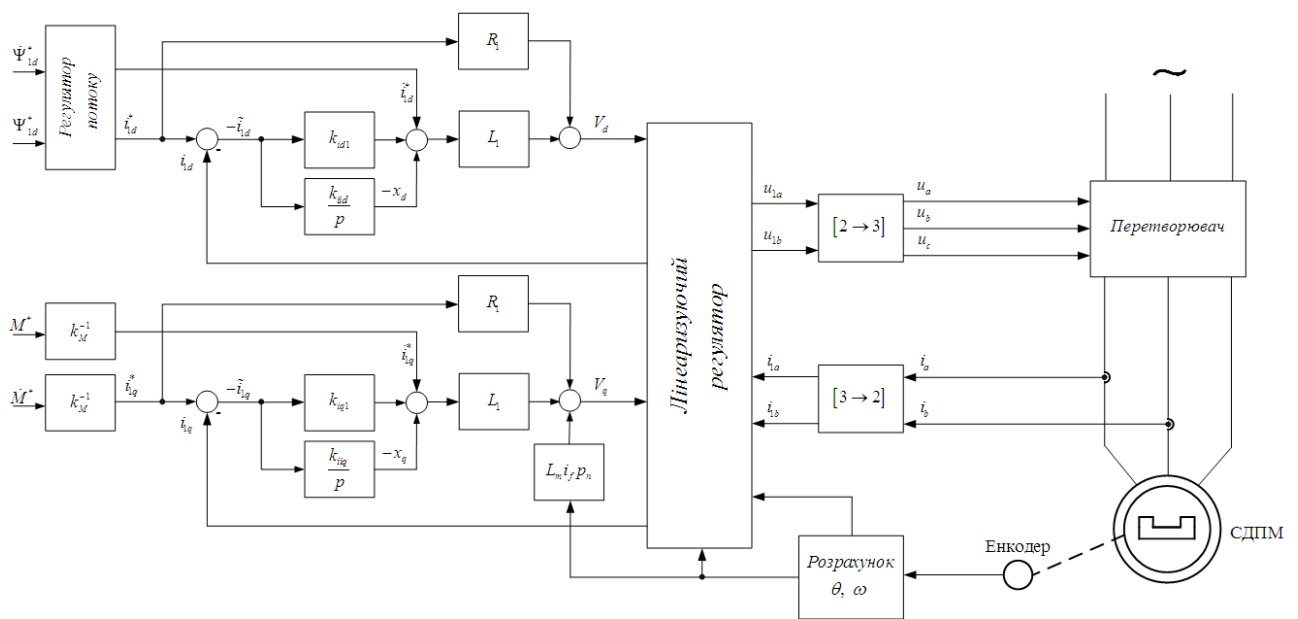


Рисунок 3.3 – Структурна схема системи відпрацювання моменту та потокозчеплення статора СД

### 3.3 Розрахунок параметрів регуляторів і зворотних зв'язків

Оптимізація динаміки похибок відпрацювання моменту (струмів статора) забезпечується за рахунок вибору коефіцієнтів пропорційної  $k_{iq}$ ,  $k_{id}$  та інтегральної  $k_{iiq}$ ,  $k_{iid}$  дії регуляторів струму щодо стандартної лінійної системи другого порядку.

Для стандартних налаштувань з коефіцієнтами демпфування  $\xi = 0.707$  та  $\xi = 1$  використовується розрахунок згідно співвідношень:

$$k_{iiq} = \frac{k_{iq}^2}{2} (\xi = 0.707),$$

$$k_{iiq} = \frac{k_{iq}^2}{4} (\xi = 1).$$
(3.14)

Приймаємо  $k_{iq} = k_{id} = 1000$ , коефіцієнт демпфування приймемо рівним 1.

### **Висновки до розділу 3**

1. Математична модель синхронного двигуна з збудженням від постійних магнітів складається з чотирьох диференційних рівнянь. За допомогою перетворення Парка-Горєва виконано перехід із системи координат (a-b) в систему координат (d-q).
2. Синтезовано алгоритм векторного керування моментом синхронного двигуна з збудженням від постійних магнітів, що забезпечує асимптотичне відпрацювання заданої траєкторії моменту двигуна.
3. Для забезпечення відпрацювання заданих струмів статора по осях d та q використано контури регулювання струму з від'ємним зворотним зв'язком.
4. Для дослідження перехідних процесів системи керування, а також перевірки виконання поставлених цілей, було розроблено математичну модель двигуна і системи керування, які було реалізовано в середовищі MatLab Simulink.
5. Для налаштування системи необхідно задати коефіцієнт пропорційної складової регулятора струму та коефіцієнт демпфування, від якого залежить величина інтегральної складової та якість перехідного процесу.

#### 4 ДОСЛІДЖЕННЯ СТАТИЧНИХ ТА ДИНАМІЧНИХ РЕЖИМІВ ЕЛЕКТРОМЕХАНІЧНОЇ СИСТЕМИ

Для дослідження динамічних характеристик використаємо розроблений алгоритм векторного керування моментом і відпрацюємо розрахований момент на валу колеса (рис. 2.8).

Для цього, опираючись на розрахунки підпункту 2.1, в середовищі MatLab Simulink розроблено підсистему, що генерує потрібний момент. Але є деякі сили, що діють на моноколес не постійно. Наприклад, аеродинамічний опір відсутній, якщо швидкість рівна нулю. В цей же час відсутній опір тертя, на моноколес діє лише сила тяжіння. Таким чином сформовано величину статичного моменту, який також залежить від швидкості руху.

Вихідні сигнали підсистем можна розглянути прямо в моделі, але зручніше вивести їх на робочий простір MatLab, де ці параметри можна вивести у вигляді графіків і оформити.

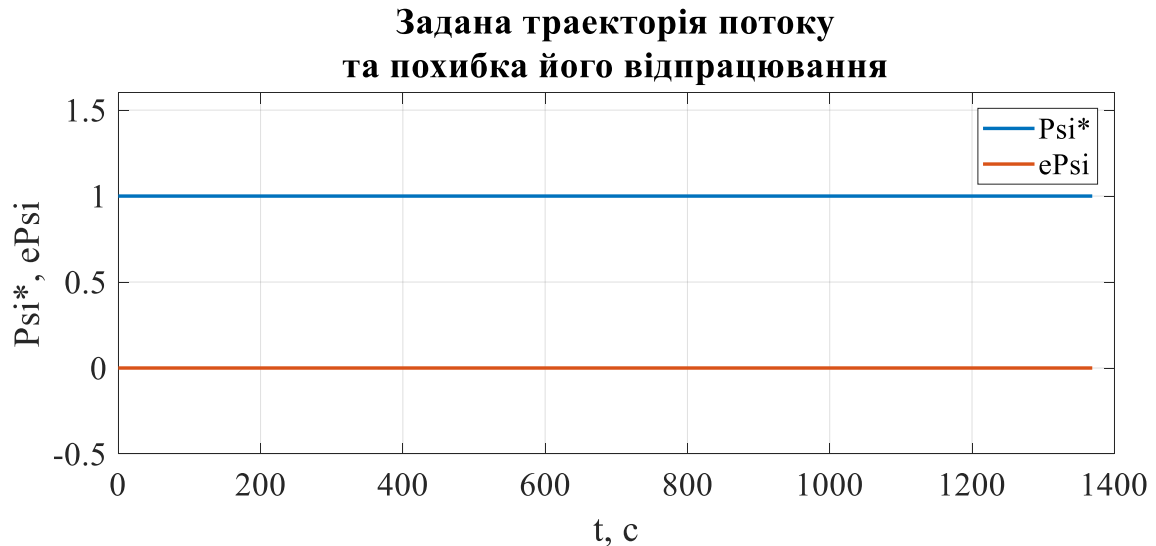
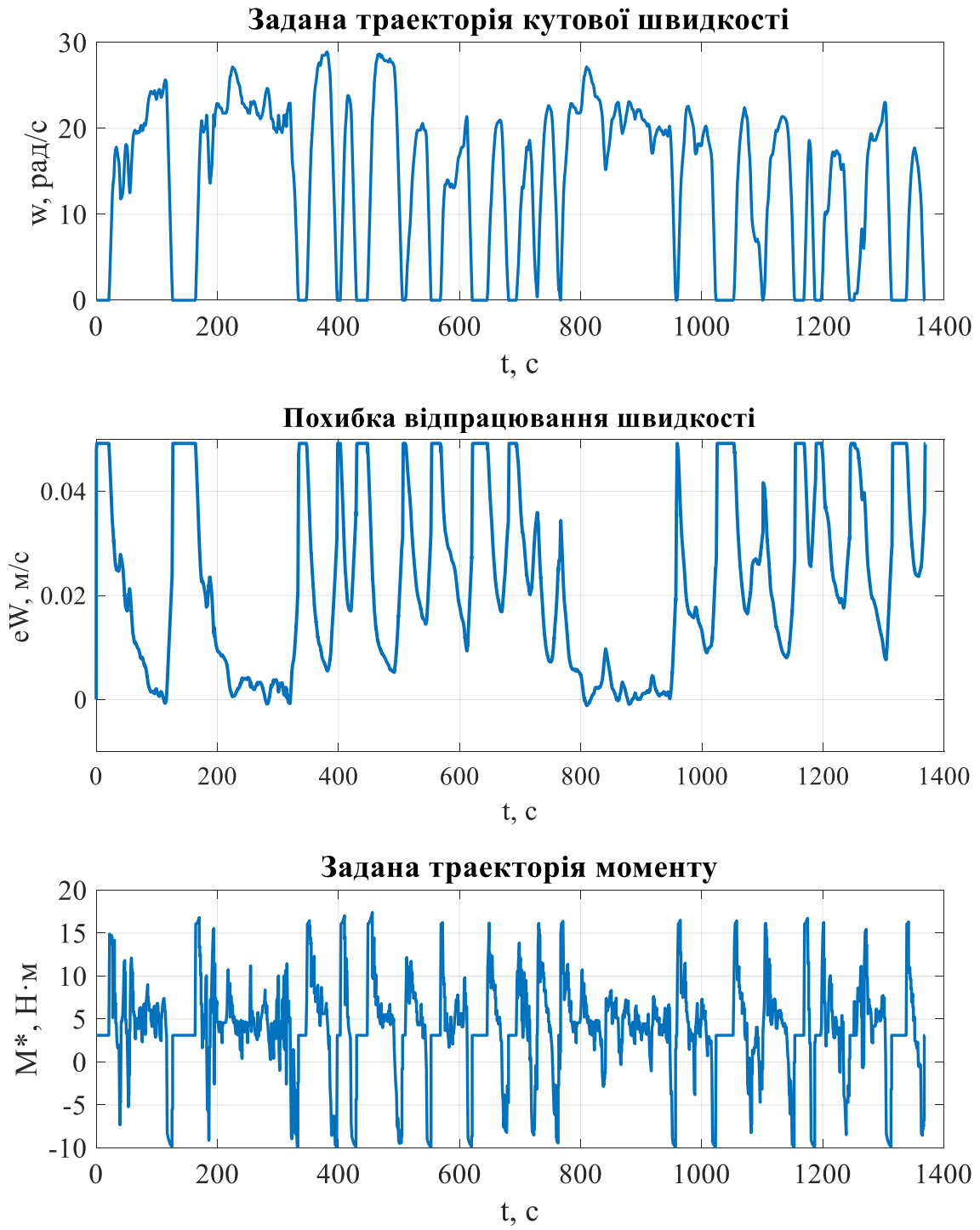


Рисунок 4.1 – Перехідні процеси в системі керування



#### 4.2 – Перехідні процеси в системі керування

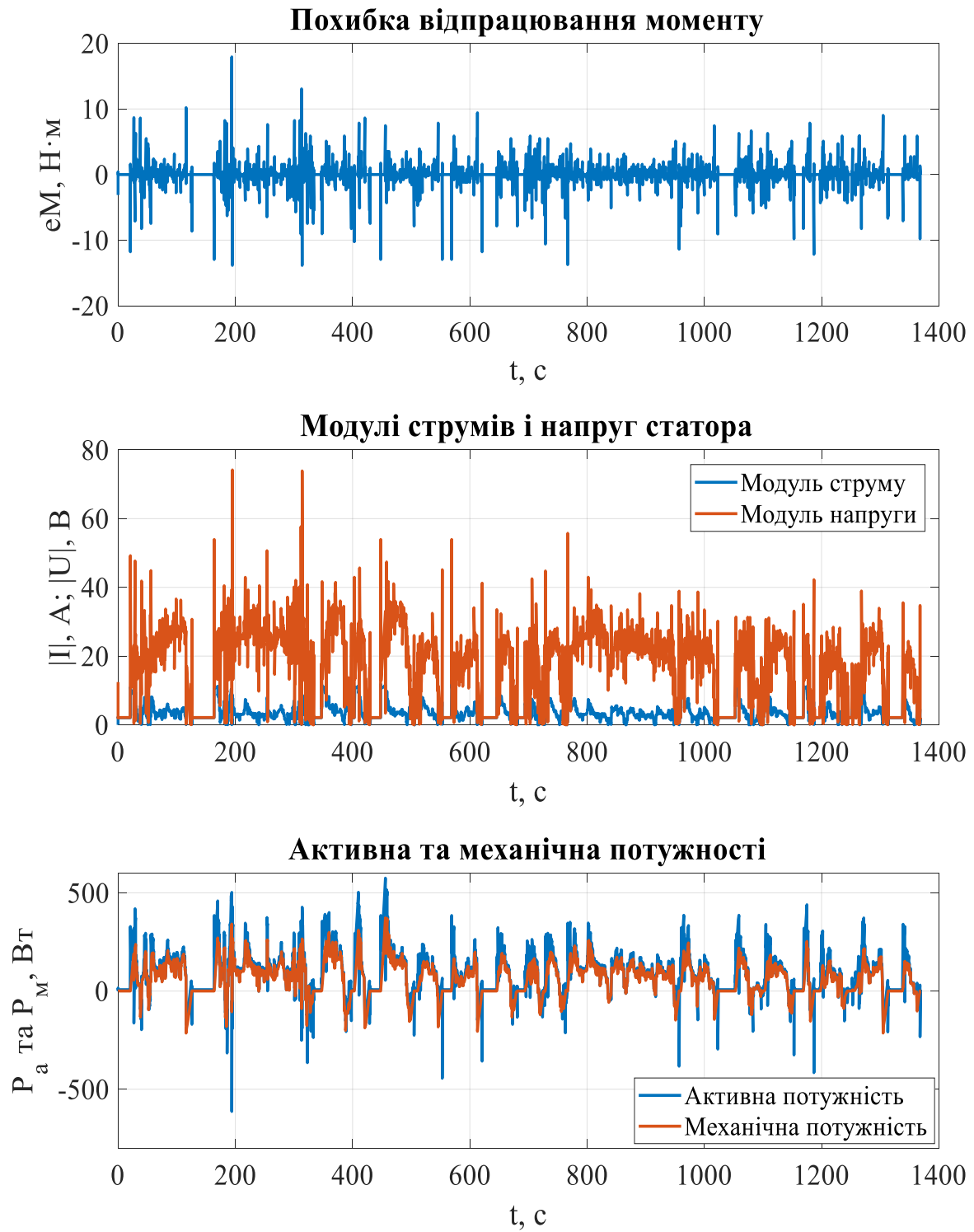


Рисунок 4.3 – Перехідні процеси в системі керування



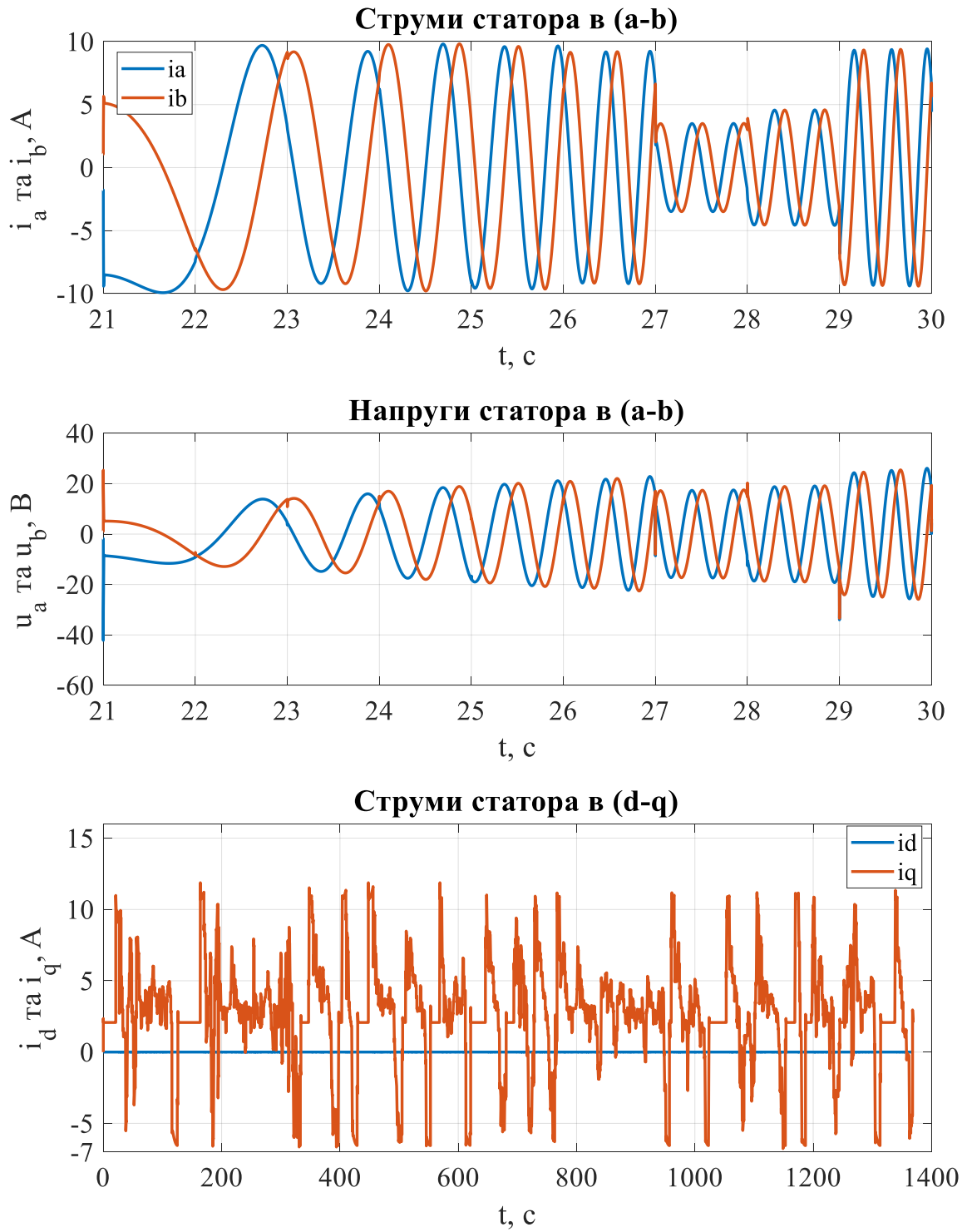


Рисунок 4.4 – Перехідні процеси в системі керування

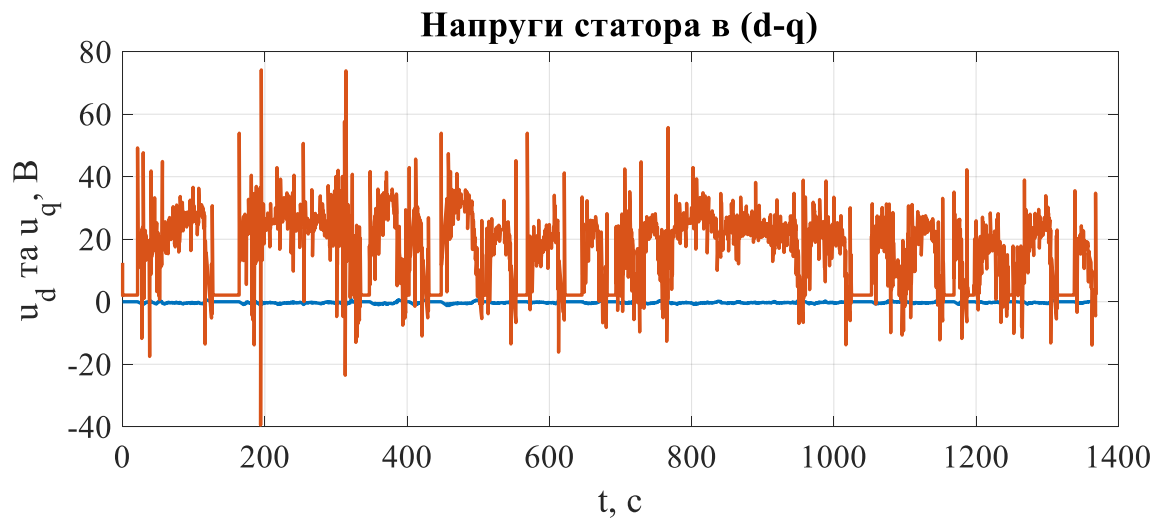


Рисунок 4.5 – Перехідні процеси в системі керування

#### Висновки до розділу 4

1. З результатів математичного моделювання встановлено, що  $\lim_{t \rightarrow \infty} \tilde{M} \approx 0$ ,  $\lim_{t \rightarrow \infty} \tilde{\psi} = 0$ , а отже цілі керування виконуються – момент і потокозчеплення відпрацьовуються асимптотично.
2. При відпрацюванні заданого моменту отримана траєкторія кутової швидкості відповідає заданій тахограмі руху. Тобто результати моделювання співпадають з результатами, які розраховані у розділі 2.

## **5 СПЕЦЧАСТИНА. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ КЕРУВАННЯ ДВИГУНА**

Для того щоб мікроконтролер міг вирішувати завдання і виконувати певні функції, його потрібно запрограмувати – записати в нього програму або ж код програми.

Насамперед, перш ніж приступити до написання будь-якої програми, а точніше коду програми, слід чітко уявляти, які функції виконуватиме мікроконтролер. Тому спочатку потрібно визначити кінцеву мету програми. Коли вона визначена і повністю зрозуміла, тоді складається алгоритм роботи програми. Алгоритм – це послідовність виконання команд. Застосування алгоритмів дозволяє більш чітко структурувати процес написання коду, а при написанні складних програм часто дозволяє скоротити час, що витрачається на їх розробку і налагодження.

Наступним етапом після складання алгоритму є безпосереднє написання коду програми. Програми для мікроконтролерів пишуться на мові Сі або Асемблері. Тільки Асемблер більше відноситься до набору інструкцій, ніж до мови програмування і є мовою низького рівня. Програми на Сі пишуться набагато швидше в порівнянні з аналогічними на Асемблері. До того ж всі складні програми пишуться переважно на Сі.

Сам код програми можна писати в будь-якому стандартному текстовому редакторі, наприклад в Блокноті. Однак на практиці користуються більш зручними редакторами.

Написаний код на Сі ще зовсім не зрозумілий мікроконтролеру (МК), оскільки МК розуміє команди тільки в двійковій (або шістнадцятковій) системі, яка представляє собою набір нулів і одиниць. Тому Сі-шний код потрібно перетворити в нулі і одиниці. Для цього застосовується спеціальна програма – компілятор, а сам процес перетворення коду називається компіляція.

Далі відкомпільований готовий код потрібно помістити в мікроконтролер, а точніше записати його в пам'ять мікроконтролера або, простіше кажучи, прошити мікроконтролер.

Для прошивки мікроконтролера застосовується пристрій, який називається програматор.

Залежно від типу програматора вхід його підключається до COM або USB порту, а вихід до певних виводів мікроконтролера.

Після того, як мікроконтролер прошитий, виконується налагодження і тестування програми на реальному пристрої.

Отже, підсумуємо етапи програмування мікроконтролерів:

- 1) визначення цілей програми;
- 2) складання алгоритму;
- 3) написання коду;
- 4) компіляція коду;
- 5) «прошивка» програми в мікроконтролер;
- 6) налаштування і тестування програми.

Слід зауважити, що налагодження і тестування програми також виконують до прошивки мікроконтролера.

Існує безліч корисних і зручних програм для програмування МК. Вони бувають як платні, так і безкоштовні. Серед них можна виділити три основних:

- 1) Atmel Studio;
- 2) CodeVisionAVR;
- 3) WinAVR.

Всі ці програми відносяться до IDE – Integrated Development Environment – інтегроване середовище розробки. У них можна писати код, компілювати і налагоджувати його.

На початковому етапі програмування всі програми краще прописувати вручну, без будь-яких спрощень. Це допоможе швидко набути необхідних

навичок, а в подальшому добре розуміти і редагувати під свої потреби коди, написані кимось іншим.

Прошивати мікроконтролери ми будемо за допомогою додаткової програми AVRDUDE.

Якщо мікроконтролера в наявності немає, то його роботу можна емітувати за допомогою програми Proteus. Вона значно спрощує процес налагодження програми навіть при наявності мікроконтролера, щоб його часто вже не перепрошивати, адже будь-який мікроконтролер має кінцеве число перезаписів, хоча це число і досить велике.

При прошивці і налагодженні мікроконтролера його зручно розташовувати на макетній платі, але це зовсім не обов'язково. Тому для більшої зручності стане в нагоді і макетна плата. Існує великий вибір макетних плат, однак краще брати ту, яка має по можливості більше число отворів – на такій платі можна розмістити більше елементів [17].

У вигляді прикладу розглянемо код прошивки для системи керування СДПМ на базі мікроконтролера STM32F103.

Даний алгоритм дозволяє програмно встановити кут випередження комутації, щоб налаштувати фазу синусоїди і змусити її випереджати ротор. Підбором кута випередження можна домогтися максимальної швидкості або ефективності двигуна. У прикладі цей кут не змінюється протягом всі роботи. Але, він програмно доступний і при необхідності його можна коригувати на ходу, наприклад, в залежності від поточної швидкості обертання ротора двигуна.

Таймер TIM1 займається генеруванням PWM сигналів для 6 ключів. Так само TIM1 забезпечує «dead time» і відключає виходи таймера, що генерують PWM сигнали, при перевантаженні по струму. Сигнал про перевантаження по струму подається на вхід таймера TIM1\_BKIN.

Генерування вихідних напруг заданої форми для трьох фаз двигуна виконується шляхом зміни шпаруватості PWM. У таблиці PMSM\_SINTABLE зберігатися 192 значень. Тобто за один електричний оборот потрібно 192 рази

змінювати шпаруватість PWM для кожної з трьох фаз. Таймер TIM4 відміряє необхідний час і викликає обробник переривання TIM4\_IRQHandler, де і виконуються необхідні обчислення і установка нових значень PWM для всіх трьох фаз, і виконується приріст індексу положення в таблиці синусів PMSM\_SinTableIndex.

Для того щоб повідомити таймером TIM4 необхідний час, потрібно знати з якою швидкістю обертається ротор. Таймер TIM3 використовується для визначення швидкості обертання. Таймери TIM3 і TIM4 тактуються однаково. Це спрощує роботу з ними. Таймер TIM4 повинен "крокувати" по таблиці синусів (PMSM\_SINTABLE) з такою швидкістю, щоб пройти всі 192 пункту таблиці за один повний електричний оборот.

Оскільки у нас є датчики положення, ми можемо в певному положенні запустити таймер лічильник TIM3 і вважати з нього лічильник при завершенні повного обороту. Потім значення цього лічильника розділити на 192 і отримати число, яке потрібно задати таймером TIM4. Оскільки TIM3 і TIM4 налаштовані на однакову частоту, переривання TIM4 буде спрацьовувати 192 рази за електричний оборот. Що і потрібно.

Оскільки датчики положення змінюють свій стан 6 раз за електричний оборот, ми можемо вимірювати час, необхідний для проходження неповного обороту, а 1/6 обороту. І ділити значення лічильника на 32 ( $192/6 = 32$ ). Таким чином, вимірювання швидкості буде виконуватися 6 разів за один електричний оборот.

Сигнали датчиків Холла генерують переривання. Оброблювач переривання визначає поточний актуальний стан ротора, зчитує дані з таймера TIM3, коригує швидкість роботи TIM4, і виконує фазову підстроювання шляхом зміщення індексу положення в таблиці синусів PMSM\_SinTableIndex. Оскільки частота обертання ротора змінюється, робота таймера TIM4 може викликати зсув вихідної напруги по фазі. Тобто вихідна синусоїда буде відставати або випереджати актуальний стан ротора. Для усунення цього ефекту застосовується

фазова підстроювання. Оскільки в момент зміни стану датчиків Холла ми точно знаємо актуальний стан ротора, можна скорегувати поточний стан в таблиці синусів шляхом корекції PMSM\_SinTableIndex. Відповідність даних датчиків Холла і положення в таблиці синусів визначено в таблицях PMSM\_STATE\_TABLE\_INDEX\_FORWARD і PMSM\_STATE\_TABLE\_INDEX\_BACKWARD для прямого і зворотного напрямку обертання ротора.

Файл «main.c»:

```
#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_rcc.h"
#include "stm32f10x_tim.h"
#include "stm32f10x_exti.h"
#include "stm32f10x_adc.h"
#include "stm32f10x_dma.h"
#include "misc.h"
#include "string.h"
#include "stdio.h"

#include "sysclk.h"
#include "adc_dma.h"
#include "bldc.h"

int main(void)
{
    // Set clock
    SetSysClockTo72();
    // ADC Init
    ADC_DMA_init();
    // TIM1, outputs, inputs, interrupts, etc. Init
    BLDC_Init();

    // Reverse pin Init
    GPIO_InitTypeDef GPIO_InitStructure;
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_2MHz;
    GPIO_Init(GPIOB, &GPIO_InitStructure);
    GPIO_SetBits(GPIOB, GPIO_Pin_0);

    while(1)
    {
        if (ADCBuffer[0] > BLDC_ADC_START) {
            if (BLDC_MotorGetSpin() == BLDC_STOP) {
```

```

        // Check Reverse pin
        if (GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_0) != 0) {
            // Forward
            BLDC_MotorSetSpin(BLDC_CW);
        }
        else {
            // Backward
            BLDC_MotorSetSpin(BLDC_CCW);
        }
        BLDC_MotorCommutation(BLDC_HallSensorsGetPosition());
    }
    BLDC_SetPWM(BLDC_ADCToPWM(ADCBuffer[0]));
}
else {
    if (BLDC_MotorGetSpin() != BLDC_STOP) {
        if (ADCBuffer[0] < BLDC_ADC_STOP) {
            BLDC_MotorStop();
        }
    }
}
}
}
}

```

Файл "bldc.c":

```

#include "stm32f10x_rcc.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_tim.h"
#include "stm32f10x_exti.h"
#include "misc.h"
#include "string.h"
#include "stdio.h"

#include "bldc.h"

uint8_t BLDC_MotorSpin = 0;
uint8_t BLDC_STATE[6] = {0,0,0,0,0,0};

#ifndef BLDC_PWMCOMPLEMENTARYMODE
uint8_t BLDC_STATE_PREV[6] = {0,0,0,0,0,0};
#endif

// BLDC motor steps tables
static const uint8_t BLDC_BRIDGE_STATE_FORWARD[8][6] = // Motor steps
{
    //      UH,UL          VH,VL          WH,WL
    { 0,0,    0,0    ,    0,0 }, // 0 //000
    { 0,1,    0,0    ,    1,0 },
    { 1,0,    0,1    ,    0,0 },
    { 0,0,    0,1    ,    1,0 },

```



```

    { 0,0,      1,0    ,      0,1 },
    { 0,1,      1,0    ,      0,0 },
    { 1,0,      0,0    ,      0,1 },
    { 0,0,      0,0    ,      0,0 }, // 0 //111
};

```

```

static const uint8_t BLDC_BRIDGE_STATE_BACKWARD[8][6] = // Motor steps
{
//      UH,UL          VH,VL      WH,WL
{ 0,0,      0,0    ,      0,0 }, // //000
{ 1,0,      0,0    ,      0,1 },
{ 0,1,      1,0    ,      0,0 },
{ 0,0,      1,0    ,      0,1 },
{ 0,0,      0,1    ,      1,0 },
{ 1,0,      0,1    ,      0,0 },
{ 0,1,      0,0    ,      1,0 },
{ 0,0,      0,0    ,      0,0 }, // //111
};

```

```

void BLDC_Init(void) {
    BLDC_HallSensorsInit();
    BLDC_PWMTimerInit();
    TIM_SelectOCxM(TIM1, TIM_Channel_1, TIM_OCMode_PWM1);
    TIM_SelectOCxM(TIM1, TIM_Channel_2, TIM_OCMode_PWM1);
    TIM_SelectOCxM(TIM1, TIM_Channel_3, TIM_OCMode_PWM1);
    BLDC_MotorStop();
}

```

```

void BLDC_HallSensorsInit(void) {
    GPIO_InitTypeDef GPIO_InitStructure;
    EXTI_InitTypeDef EXTI_InitStructure;
    NVIC_InitTypeDef NVIC_InitStructure;

    // Enable clock
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB
RCC_APB2Periph_AFIO, ENABLE);

    // Init GPIO
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7 | GPIO_Pin_8 | GPIO_Pin_9;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_2MHz;
    GPIO_Init(GPIOB, &GPIO_InitStructure);

    // Init NVIC
    NVIC_InitStructure.NVIC_IRQChannel = EXTI9_5_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0x00;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0x00;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);

    // Tell system that you will use EXTI_Lines */

```

```

GPIO_EXTILineConfig(GPIO_PortSourceGPIOB, GPIO_PinSource7);
GPIO_EXTILineConfig(GPIO_PortSourceGPIOB, GPIO_PinSource8);
GPIO_EXTILineConfig(GPIO_PortSourceGPIOB, GPIO_PinSource9);

// EXTI
EXTI_InitStruct.EXTI_Line = EXTI_Line7 | EXTI_Line8 | EXTI_Line9;
EXTI_InitStruct.EXTI_LineCmd = ENABLE;
EXTI_InitStruct.EXTI_Mode = EXTI_Mode_Interrupt;
EXTI_InitStruct.EXTI_Trigger = EXTI_Trigger_Rising_Falling;
EXTI_Init(&EXTI_InitStruct);
}

void EXTI9_5_IRQHandler(void) {
    if ((EXTI_GetITStatus(EXTI_Line7) | EXTI_GetITStatus(EXTI_Line8) |
EXTI_GetITStatus(EXTI_Line9)) != RESET) {
        // Clear interrupt flags
        EXTI_ClearITPendingBit(EXTI_Line7);
        EXTI_ClearITPendingBit(EXTI_Line8);
        EXTI_ClearITPendingBit(EXTI_Line9);

        // Commutation
        BLDC_MotorCommutation(BLDC_HallSensorsGetPosition());
    }
}

void BLDC_PWMTimerInit(void) {
    GPIO_InitTypeDef GPIO_InitStructure;
    TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
    TIM_OCInitTypeDef TIM_OCInitStructure;
    TIM_BDTRInitTypeDef TIM_BDTRInitStructure;

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA |
RCC_APB2Periph_GPIOB, ENABLE);

    //initialize Tim1 PWM outputs
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_13 | GPIO_Pin_14 | GPIO_Pin_15;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
    GPIO_Init(GPIOB, &GPIO_InitStructure);

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 | GPIO_Pin_9 | GPIO_Pin_10;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_TIM1, ENABLE);

    // Time Base configuration
    TIM_TimeBaseStructure.TIM_Prescaler = 0;
    TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
    TIM_TimeBaseStructure.TIM_Period = BLDC_CHOPPER_PERIOD;

```

```

TIM_TimeBaseStructure.TIM_ClockDivision = 0;
TIM_TimeBaseStructure.TIM_RepetitionCounter = 0;
TIM_TimeBaseInit(TIM1, &TIM_TimeBaseStructure);

// Channel 1, 2, 3 – set to PWM mode - all 6 outputs
TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_Timing;
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_OutputNState = TIM_OutputNState_Enable;
TIM_OCInitStructure.TIM_Pulse = 0; // initialize to zero output

TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
TIM_OCInitStructure.TIM_OCNPolarity = TIM_OCNPolarity_High;
TIM_OCInitStructure.TIM_OCIdleState = TIM_OCIdleState_Reset; /// !!!
TIM_OCInitStructure.TIM_OCNIdleState = TIM_OCNIdleState_Reset; /// !!!

TIM_OC1Init(TIM1, &TIM_OCInitStructure);
TIM_OC2Init(TIM1, &TIM_OCInitStructure);
TIM_OC3Init(TIM1, &TIM_OCInitStructure);

TIM_BDTRInitStructure.TIM_OSSRState = TIM_OSSRState_Enable;
TIM_BDTRInitStructure.TIM_OSSIState = TIM_OSSIState_Enable;
TIM_BDTRInitStructure.TIM_LOCKLevel = TIM_LOCKLevel_OFF;

// DeadTime[ns] = value * (1/SystemCoreFreq) (on 72MHz: 7 is 98ns)
TIM_BDTRInitStructure.TIM_DeadTime = BLDC_NOL;

TIM_BDTRInitStructure.TIM_AutomaticOutput = TIM_AutomaticOutput_Enable;

// Break functionality (overload input)
TIM_BDTRInitStructure.TIM_Break = TIM_Break_Enable;
TIM_BDTRInitStructure.TIM_BreakPolarity = TIM_BreakPolarity_Low;

TIM_BDTRConfig(TIM1, &TIM_BDTRInitStructure);

TIM_Cmd(TIM1, ENABLE);
// enable motor timer main output (the bridge signals)
TIM_CtrlPWMOutputs(TIM1, ENABLE);
}

uint8_t BLDC_HallSensorsGetPosition(void) {
    return (uint8_t) ((GPIO_ReadInputData(GPIOB) & (GPIO_Pin_7 | GPIO_Pin_8 |
GPIO_Pin_9)) >> 7);
}

uint8_t BLDC_MotorGetSpin(void) {
    return BLDC_MotorSpin;
}

void BLDC_MotorSetSpin(uint8_t spin) {
    BLDC_MotorSpin = spin;
}

```

```
}
```

Далі описана функція зупинки (гальмування) двигуна:

```
void BLDC_MotorStop(void)
{
    BLDC_SetPWM(0);

    TIM_CCxCmd(TIM1, TIM_Channel_1, TIM_CCx_Disable);
    TIM_CCxCmd(TIM1, TIM_Channel_2, TIM_CCx_Disable);
    TIM_CCxCmd(TIM1, TIM_Channel_3, TIM_CCx_Disable);

    TIM_CCxNCmd(TIM1, TIM_Channel_1, TIM_CCxN_Disable);
    TIM_CCxNCmd(TIM1, TIM_Channel_2, TIM_CCxN_Disable);
    TIM_CCxNCmd(TIM1, TIM_Channel_3, TIM_CCxN_Disable);

    BLDC_MotorSpin = 0;
    #ifndef BLDC_PWMCOMPLEMENTARYMODE
        memset(BLDC_STATE_PREV, 0, sizeof(BLDC_STATE_PREV));
    #endif
}

#ifdef BLDC_PWMTOPBOTTOMKEYS
void BLDC_MotorCommutation(uint16_t hallpos)
{
    if (BLDC_MotorSpin == BLDC_CW) {
        memcpy(BLDC_STATE, BLDC_BRIDGE_STATE_FORWARD[hallpos],
sizeof(BLDC_STATE));
    }
    else {
        memcpy(BLDC_STATE, BLDC_BRIDGE_STATE_BACKWARD[hallpos],
sizeof(BLDC_STATE));
    }

    // Disable if need
    if (!BLDC_STATE[UH])    TIM_CCxCmd(TIM1,    TIM_Channel_1,
TIM_CCx_Disable);
    if (!BLDC_STATE[UL])    TIM_CCxNCmd(TIM1,    TIM_Channel_1,
TIM_CCxN_Disable);
    if (!BLDC_STATE[VH])    TIM_CCxCmd(TIM1,    TIM_Channel_2,
TIM_CCx_Disable);
    if (!BLDC_STATE[VL])    TIM_CCxNCmd(TIM1,    TIM_Channel_2,
TIM_CCxN_Disable);
    if (!BLDC_STATE[WH])    TIM_CCxCmd(TIM1,    TIM_Channel_3,
TIM_CCx_Disable);
    if (!BLDC_STATE[WL])    TIM_CCxNCmd(TIM1,    TIM_Channel_3,
TIM_CCxN_Disable);

    // Enable if need. If previous state is Enabled then not enable again. Else output do
flip-flop.
    if (BLDC_STATE[UH] & !BLDC_STATE[UL] & !BLDC_STATE_PREV[UH])
TIM_CCxCmd(TIM1, TIM_Channel_1, TIM_CCx_Enable);
```

```

        if (BLDC_STATE[UL] & !BLDC_STATE[UH] & !BLDC_STATE_PREV[UL])
TIM_CCxNCmd(TIM1, TIM_Channel_1, TIM_CCxN_Enable);
        if (BLDC_STATE[VH] & !BLDC_STATE[VL] & !BLDC_STATE_PREV[VH])
TIM_CCxCmd(TIM1, TIM_Channel_2, TIM_CCx_Enable);
        if (BLDC_STATE[VL] & !BLDC_STATE[VH] & !BLDC_STATE_PREV[VL])
TIM_CCxNCmd(TIM1, TIM_Channel_2, TIM_CCxN_Enable);
        if (BLDC_STATE[WH] & !BLDC_STATE[WL] & !BLDC_STATE_PREV[WH])
TIM_CCxCmd(TIM1, TIM_Channel_3, TIM_CCx_Enable);
        if (BLDC_STATE[WL] & !BLDC_STATE[WH] & !BLDC_STATE_PREV[WL])
TIM_CCxNCmd(TIM1, TIM_Channel_3, TIM_CCxN_Enable);

        memcpy(BLDC_STATE_PREV, BLDC_STATE, sizeof(BLDC_STATE));
    }
#endif

#ifdef BLDC_PWMTOPKEYS

```

Функція комутації ключів:

```

void BLDC_MotorCommutation(uint16_t hallpos)
{
    if (BLDC_MotorSpin == BLDC_CW) {
        memcpy(BLDC_STATE, BLDC_BRIDGE_STATE_FORWARD[hallpos],
sizeof(BLDC_STATE));
    }
    else {
        memcpy(BLDC_STATE, BLDC_BRIDGE_STATE_BACKWARD[hallpos],
sizeof(BLDC_STATE));
    }

    // Disable if need
    if (!BLDC_STATE[UH]) TIM_CCxCmd(TIM1, TIM_Channel_1,
TIM_CCx_Disable);
    if (!BLDC_STATE[UL]) TIM_CCxNCmd(TIM1, TIM_Channel_1,
TIM_CCxN_Disable);
    if (!BLDC_STATE[VH]) TIM_CCxCmd(TIM1, TIM_Channel_2,
TIM_CCx_Disable);
    if (!BLDC_STATE[VL]) TIM_CCxNCmd(TIM1, TIM_Channel_2,
TIM_CCxN_Disable);
    if (!BLDC_STATE[WH]) TIM_CCxCmd(TIM1, TIM_Channel_3,
TIM_CCx_Disable);
    if (!BLDC_STATE[WL]) TIM_CCxNCmd(TIM1, TIM_Channel_3,
TIM_CCxN_Disable);

    // Enable if need. If previous state is Enabled then not enable again. Else output do
flip-flop.
    if (BLDC_STATE[UH] & !BLDC_STATE[UL] & !BLDC_STATE_PREV[UH]) {
        TIM_SelectOCxM(TIM1, TIM_Channel_1, TIM_OCMode_PWM1);
        TIM_CCxCmd(TIM1, TIM_Channel_1, TIM_CCx_Enable);
    }
}

```

```

    if (BLDC_STATE[UL] & !BLDC_STATE[UH] & !BLDC_STATE_PREV[UL]) {
        TIM_SelectOCxM(TIM1, TIM_Channel_1, TIM_ForcedAction_Active);
        TIM_CCxNCmd(TIM1, TIM_Channel_1, TIM_CCxN_Enable);
    }
    if (BLDC_STATE[VH] & !BLDC_STATE[VL] & !BLDC_STATE_PREV[VH]) {
        TIM_SelectOCxM(TIM1, TIM_Channel_2, TIM_OCMode_PWM1);
        TIM_CCxCmd(TIM1, TIM_Channel_2, TIM_CCx_Enable);
    }
    if (BLDC_STATE[VL] & !BLDC_STATE[VH] & !BLDC_STATE_PREV[VL]) {
        TIM_SelectOCxM(TIM1, TIM_Channel_2, TIM_ForcedAction_Active);
        TIM_CCxNCmd(TIM1, TIM_Channel_2, TIM_CCxN_Enable);
    }
    if (BLDC_STATE[WH] & !BLDC_STATE[WL] & !BLDC_STATE_PREV[WH]) {
        TIM_SelectOCxM(TIM1, TIM_Channel_3, TIM_OCMode_PWM1);
        TIM_CCxCmd(TIM1, TIM_Channel_3, TIM_CCx_Enable);
    }
    if (BLDC_STATE[WL] & !BLDC_STATE[WH] & !BLDC_STATE_PREV[WL]) {
        TIM_SelectOCxM(TIM1, TIM_Channel_3, TIM_ForcedAction_Active);
        TIM_CCxNCmd(TIM1, TIM_Channel_3, TIM_CCxN_Enable);
    }

    memcpy(BLDC_STATE_PREV, BLDC_STATE, sizeof(BLDC_STATE));
}
#endif

#ifdef BLDC_PWMBOTTOMKEYS
void BLDC_MotorCommutation(uint16_t hallpos)
{
    if (BLDC_MotorSpin == BLDC_CW) {
        memcpy(BLDC_STATE, BLDC_BRIDGE_STATE_FORWARD[hallpos],
sizeof(BLDC_STATE));
    }
    else {
        memcpy(BLDC_STATE, BLDC_BRIDGE_STATE_BACKWARD[hallpos],
sizeof(BLDC_STATE));
    }

    // Disable if need
    if (!BLDC_STATE[UH]) TIM_CCxCmd(TIM1, TIM_Channel_1,
TIM_CCx_Disable);
    if (!BLDC_STATE[UL]) TIM_CCxNCmd(TIM1, TIM_Channel_1,
TIM_CCxN_Disable);
    if (!BLDC_STATE[VH]) TIM_CCxCmd(TIM1, TIM_Channel_2,
TIM_CCx_Disable);
    if (!BLDC_STATE[VL]) TIM_CCxNCmd(TIM1, TIM_Channel_2,
TIM_CCxN_Disable);
    if (!BLDC_STATE[WH]) TIM_CCxCmd(TIM1, TIM_Channel_3,
TIM_CCx_Disable);
    if (!BLDC_STATE[WL]) TIM_CCxNCmd(TIM1, TIM_Channel_3,
TIM_CCxN_Disable);

```

```

flip-flop.
// Enable if need. If previous state is Enabled then not enable again. Else output do
if (BLDC_STATE[UH] & !BLDC_STATE[UL] & !BLDC_STATE_PREV[UH]) {
    TIM_SelectOCxM(TIM1, TIM_Channel_1, TIM_ForcedAction_Active);
    TIM_CCxCmd(TIM1, TIM_Channel_1, TIM_CCx_Enable);
}
if (BLDC_STATE[UL] & !BLDC_STATE[UH] & !BLDC_STATE_PREV[UL]) {
    TIM_SelectOCxM(TIM1, TIM_Channel_1, TIM_OCMode_PWM1);
    TIM_CCxNCmd(TIM1, TIM_Channel_1, TIM_CCxN_Enable);
}
if (BLDC_STATE[VH] & !BLDC_STATE[VL] & !BLDC_STATE_PREV[VH]) {
    TIM_SelectOCxM(TIM1, TIM_Channel_2, TIM_ForcedAction_Active);
    TIM_CCxCmd(TIM1, TIM_Channel_2, TIM_CCx_Enable);
}
if (BLDC_STATE[VL] & !BLDC_STATE[VH] & !BLDC_STATE_PREV[VL]) {
    TIM_SelectOCxM(TIM1, TIM_Channel_2, TIM_OCMode_PWM1);
    TIM_CCxNCmd(TIM1, TIM_Channel_2, TIM_CCxN_Enable);
}
if (BLDC_STATE[WH] & !BLDC_STATE[WL] & !BLDC_STATE_PREV[WH]) {
    TIM_SelectOCxM(TIM1, TIM_Channel_3, TIM_ForcedAction_Active);
    TIM_CCxCmd(TIM1, TIM_Channel_3, TIM_CCx_Enable);
}
if (BLDC_STATE[WL] & !BLDC_STATE[WH] & !BLDC_STATE_PREV[WL]) {
    TIM_SelectOCxM(TIM1, TIM_Channel_3, TIM_OCMode_PWM1);
    TIM_CCxNCmd(TIM1, TIM_Channel_3, TIM_CCxN_Enable);
}

memcpy(BLDC_STATE_PREV, BLDC_STATE, sizeof(BLDC_STATE));
}
#endif

#ifdef BLDC_PWMCOMPLEMENTARYMODE
Далі описано роботу ШІМ з врахуванням мертвого часу:
void BLDC_MotorCommutation(uint16_t hallpos) {
    if (BLDC_MotorSpin == BLDC_CW) {
        memcpy(BLDC_STATE, BLDC_BRIDGE_STATE_FORWARD[hallpos],
sizeof(BLDC_STATE));
    }
    else {
        memcpy(BLDC_STATE, BLDC_BRIDGE_STATE_BACKWARD[hallpos],
sizeof(BLDC_STATE));
    }

    // PWM at low side FET of bridge U
    // active freewheeling at high side FET of bridge U
    // if low side FET is in PWM off mode then the high side FET
    // is ON for active freewheeling. This mode needs correct definition
    // of dead time otherwise we have shoot-through problems

```

```

if (BLDC_STATE[UH]) {
    TIM_SelectOCxM(TIM1, TIM_Channel_1, TIM_OCMode_PWM1);
    TIM_CCxCmd(TIM1, TIM_Channel_1, TIM_CCx_Enable);
    TIM_CCxNCmd(TIM1, TIM_Channel_1, TIM_CCxN_Enable);
} else {
    // Low side FET: OFF
    TIM_CCxCmd(TIM1, TIM_Channel_1, TIM_CCx_Disable);
    if (BLDC_STATE[UL]){
        // High side FET: ON
        TIM_SelectOCxM(TIM1,
                        TIM_Channel_1,
TIM_ForceAction_Active);
        TIM_CCxNCmd(TIM1, TIM_Channel_1, TIM_CCxN_Enable);
    } else {
        // High side FET: OFF
        TIM_CCxNCmd(TIM1, TIM_Channel_1, TIM_CCxN_Disable);
    }
}

if (BLDC_STATE[VH]) {
    TIM_SelectOCxM(TIM1, TIM_Channel_2, TIM_OCMode_PWM1);
    TIM_CCxCmd(TIM1, TIM_Channel_2, TIM_CCx_Enable);
    TIM_CCxNCmd(TIM1, TIM_Channel_2, TIM_CCxN_Enable);
} else {
    // Low side FET: OFF
    TIM_CCxCmd(TIM1, TIM_Channel_2, TIM_CCx_Disable);
    if (BLDC_STATE[VL]){
        // High side FET: ON
        TIM_SelectOCxM(TIM1,
                        TIM_Channel_2,
TIM_ForceAction_Active);
        TIM_CCxNCmd(TIM1, TIM_Channel_2, TIM_CCxN_Enable);
    } else {
        // High side FET: OFF
        TIM_CCxNCmd(TIM1, TIM_Channel_2, TIM_CCxN_Disable);
    }
}

if (BLDC_STATE[WH]) {
    TIM_SelectOCxM(TIM1, TIM_Channel_3, TIM_OCMode_PWM1);
    TIM_CCxCmd(TIM1, TIM_Channel_3, TIM_CCx_Enable);
    TIM_CCxNCmd(TIM1, TIM_Channel_3, TIM_CCxN_Enable);
} else {
    // Low side FET: OFF
    TIM_CCxCmd(TIM1, TIM_Channel_3, TIM_CCx_Disable);
    if (BLDC_STATE[WL]){
        // High side FET: ON
        TIM_SelectOCxM(TIM1,
                        TIM_Channel_3,
TIM_ForceAction_Active);
        TIM_CCxNCmd(TIM1, TIM_Channel_3, TIM_CCxN_Enable);
    } else {
        // High side FET: OFF
        TIM_CCxNCmd(TIM1, TIM_Channel_3, TIM_CCxN_Disable);
    }
}

```



```

    }
}

}
#endif

uint16_t BLDC_ADCToPWM(uint16_t ADC_VALUE) {
    uint32_t tmp;

    if (ADC_VALUE < BLDC_ADC_STOP) {
        return 0;
    }

    if (ADC_VALUE > BLDC_ADC_MAX) {
        return BLDC_CHOPPER_PERIOD+1;
    }

    tmp = (uint32_t)(ADC_VALUE-BLDC_ADC_STOP) *
(uint32_t)BLDC_CHOPPER_PERIOD / (uint32_t)(BLDC_ADC_MAX - BLDC_ADC_START);

    return (uint16_t) tmp;
}

```

```

void BLDC_SetPWM(uint16_t PWM)
{
    TIM1->CCR1 = PWM;
    TIM1->CCR2 = PWM;
    TIM1->CCR3 = PWM;
}

```

Даний файл зберігає оголошення всіх змінних:

Файл "bldc.h":

```

#ifndef _BLDC_LIB_H_
#define _BLDC_LIB_H_

// PWM Frequency = 72000000/BLDC_CHOPPER_PERIOD
#define BLDC_CHOPPER_PERIOD 4500
// Dead time = BLDC_NOL/72000000 (on 72MHz: 7 is 98ns)
// (on 72MHz: 72 is 1000ns)
// #define BLDC_NOL 72
#define BLDC_NOL 72

// #define BLDC_PWMTOPKEYS
// #define BLDC_PWMBOTTOMKEYS
// #define BLDC_PWMTOPBOTTOMKEYS
#define BLDC_PWMCOMPLEMENTARYMODE

#define BLDC_ADC_START 200
#define BLDC_ADC_STOP 50

```

```

#define BLDC_ADC_MAX 4000

////////////////////////////////////
#define BLDC_STOP      0
#define BLDC_CW        1
#define BLDC_CCW       2

#define UH   0
#define UL   1
#define VH   2
#define VL   3
#define WH   4
#define WL   5

void BLDC_Init(void);
void BLDC_HallSensorsInit(void);
void BLDC_PWMTimerInit(void);
uint8_t BLDC_HallSensorsGetPosition(void);
void BLDC_MotorSetSpin(uint8_t spin);
uint8_t BLDC_MotorGetSpin(void);
void BLDC_MotorStop(void);
void BLDC_MotorCommutation(uint16_t hallpos);
uint16_t BLDC_ADCToPWM(uint16_t ADC_VALUE);
void BLDC_SetPWM(uint16_t PWM);

#endif

```

## Висновки до розділу 5

1. Описано основні кроки та обладнання для програмування мікроконтролера.
2. Розроблено алгоритм керування безколекторним двигуном постійного струму (brushless DC).
3. Створено програмний код для «прошивки» контролера. Замість самого контролера використано його віртуальний образ.

## ВИСНОВКИ

В даній бакалаврській роботі було розроблено електромеханічну систему моноколесу на основі безколекторного двигуна постійного струму.

1. Розглянуто безколекторні двигуни постійного струму (БДПС; також їх називають синхронні двигуни з постійними магнітами, СДПМ), їх основні переваги перед іншими видами двигунів.
2. Розглянуто методи керування СДПМ та їх особливості.
3. Висунуто основні вимоги до двигунів, що використовуються в електротранспорті і до самого електротранспорту.
4. Розраховано сили, що діють на електротранспорт під час руху, розглянуто середньостатистичний графік швидкості руху в умовах міського циклу.
5. За графіками потужності і моменту підібрано двигун, проведена його перевірка по перегріву.
6. Проведено розрахунок акумуляторної батареї, її потужності і енергії, необхідних для живлення двигуна на весь цикл руху.
7. Синтезовано алгоритм векторного керування моменту та потокозчеплення при живленні від джерела напруги.
8. Розраховано та розроблено схему силової частини та схему керування двигуна.
9. Проведено дослідження перехідних процесів двигуна з векторним керуванням.
10. Описано порядок дій і необхідного обладнання для програмування мікроконтролерів. Для прикладу створено код керування СМПМ на базі використаного на схемі мікроконтролера.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Що собою являє моноколесо? URL: <https://onewheel.com.ua/ua/blog/391-chto-predstavlyaet-soboy-monokoleso> (дата звернення 20.03.2020 р.);
2. Синхронный двигатель с постоянными магнитами, управление синхронным двигателем с постоянными магнитами. URL: <https://engineering-solutions.ru/motorcontrol/pmsm/> (дата звернення 22.03.2020 р.);
3. Моноколесо InMotion V8 Black (IM-V8-BK). URL: <https://rozetka.com.ua/145956491/p145956491/> (дата звернення 25.03.2020 р.);
4. Daniel J. Auger (2020). Driving Cycle (Simulink Block). URL: <https://uk.mathworks.com/matlabcentral/fileexchange/46777-driving-cycle-simulink-block> (дата звернення 25.03.2020 р.);
5. Al-Obaidi, Abdulkareem Sh, Sun, L.C.. Calculation and optimization of the aerodynamic drag of an open-wheel race car. 2014. 1-15;
6. Mosteller RD. Simplified calculation of body surface area. N Engl J Med 1987;
7. Онищенко О. Г., Коробко Б. А., Ващенко К. М. Структура, кинематика и динамика механизмов. ПолтНТУ, 2010. — 274 с.
8. Плотность воздуха. URL: [https://ru.wikipedia.org/wiki/Плотность\\_воздуха](https://ru.wikipedia.org/wiki/Плотность_воздуха) (дата звернення 27.03.2020 р.);
9. Anderson J. D. Fundamentals of Aerodynamics. 6th ed. New York : McGraw-Hill Education, 2016. 1152 p.
10. Ferdinand P. Beer & E. Russell Johnston Jr. Vector Mechanics for Engineers (Dynamics) McGraw-Hill Book Company Inc. 1977. – 563 p.
11. Электродвигун (мотор-колесо) KS-16S . URL: <https://kingsong.com.ua/p1086467779-elektrodivgun-motor-koleso.html> (дата звернення 07.04.20);
12. Аккумулятор Rablex 18650 3200 Mah Li-Ion 3.7 V Original. URL: <https://rozetka.com.ua/ua/208378021/p208378021/> (дата звернення 01.04.20);

13. Сайт виробника «Infeneon». URL: <https://www.infineon.com/cms/en/> (дата звернення 01.05.20);
14. ЕСАР (K50-35), 1000 мкФ, 100 В, 105°C, 20% snap in 22x35, Конденсатор электролитический алюминевый. URL: <https://www.chipdip.ru/product0/9000565974> (дата звернення 25.04.20);
15. Сайт виробника «Allegro Microsystems». URL: <https://www.allegromicro.com/en/> (дата звернення 26.04.20);
16. STM32 - PMSM Control - Управление PMSM с помощью STM32. URL: [https://blog.avislabs.com/stm32-pmsm\\_ru/](https://blog.avislabs.com/stm32-pmsm_ru/) (дата звернення 28.04.20);
17. М. М. Желінський, Я. С. Крячко. Метод обчислення функцій синуса і косинуса для перетворень координат в системах векторного керування: матеріали конференції ВНТУ, Молодь в науці: дослідження, проблеми, перспективи (МН-2020)
18. Что нужно для программирования микроконтроллеров? URL: <https://diodov.net/chto-nuzhno-dlya-programmirovaniya-mikrokontrollerov/> (дата звернення 26.05.20);

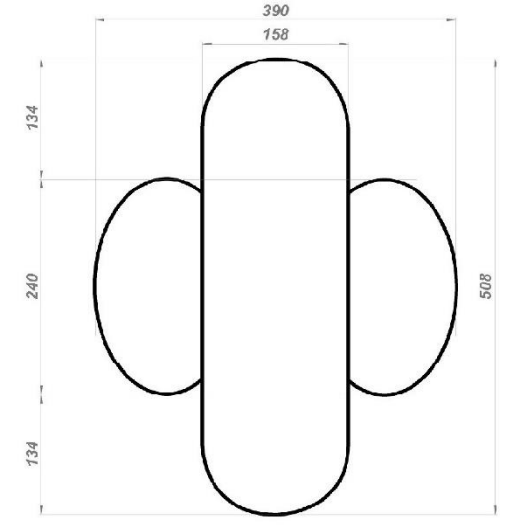
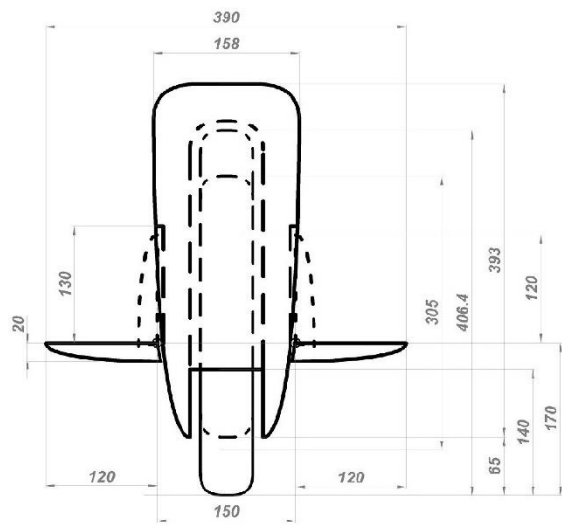
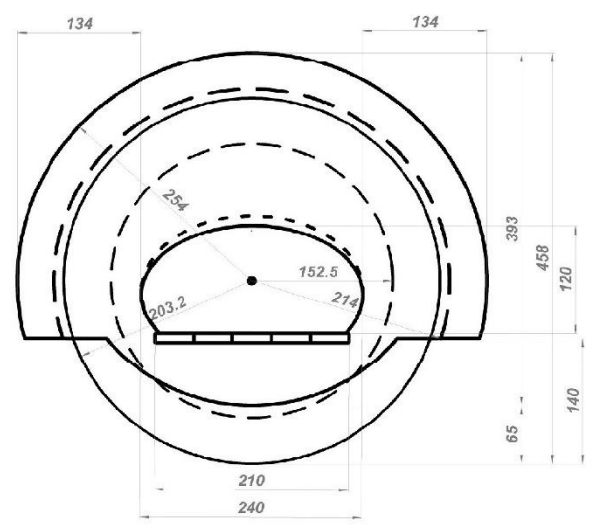
**ДОДАТОК А**  
**Плакати графічної частини**

Таблиця А1 - Параметри АКБ

Найменування	Значення
Вихідна напруга (номінальна), В	24
Вихідна напруга (робоча), В	17.9-27.3
Ресурс (циклів)	500
Потужність, Вт·год	366
Ємність, мА·год	22400
Вага, кг	1.5

Таблиця А2 - Параметри мотор-колеса

Найменування	Значення
Номінальний запас ходу, км	24
Максимальна швидкість, км/год	17.9-27.3
Потужність (номінальна/максимальна), Вт	500/1200
Потужність, Вт·год	366

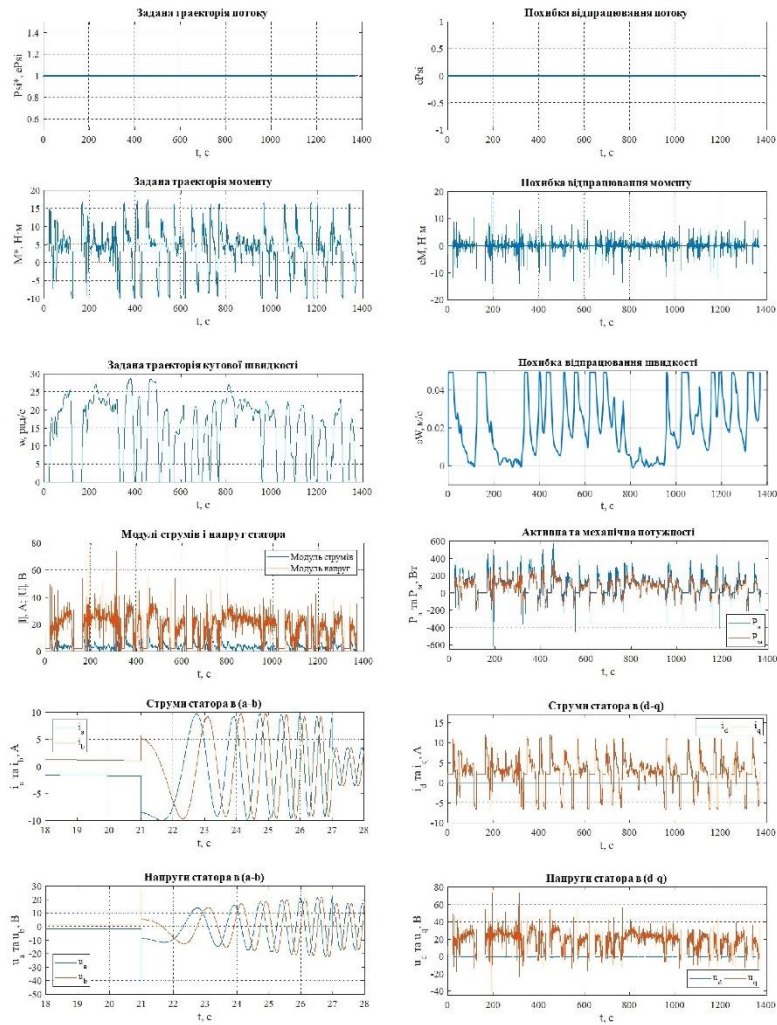


141.0007.006 БР									
Вн.	Док.	М. Розв.	Г. Розв.	Л. Розв.	Специфікація на модуль мотоколеса з функцією ABS (система антиблокування гальмівних коліс)				
Розроб.	Кондрат Р. С.				Висновок експертної групи проєктування автомобіля				
Перевір.	Мельничук Н. М.				Автори: 1. Демченко О. 2. Кіт О. 3. Іванов С. 4. Петренко В. 5. Сидоренко О.				
Розроб.	Мельничук Н. М.				КІТ на тему: Система ABS (система антиблокування гальмівних коліс)				
Н. Розв.	Тетюх В. І.				Формат				
Затв.	Мельничук Н. М.								

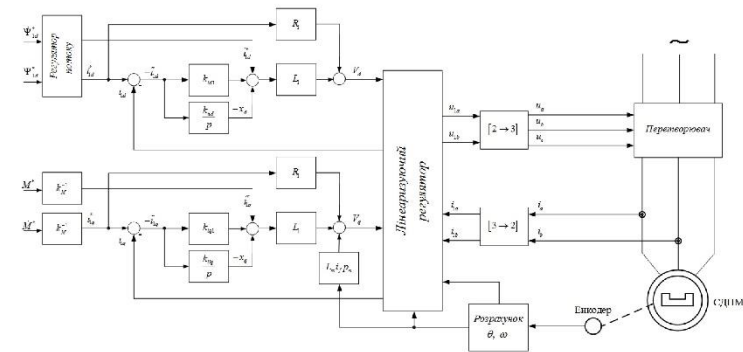




## Результати дослідження перехідних процесів



## Структурна схема системи керування



141.0007.006 БР					Лист		
Вн.	Док.	Мі. Вироб.	Листів	Листів	Лист	Місяц	Місяць
Розроб.	М. В. Влас	Кондрат Р. С.					
Перевір.	Михайленко Н. М.						
Т. Вектор							
Рухомі	Михайленко С. А.						
Н. Вектор	Тетюха В. І.						
Затв.	Михайленко С. М.						

Автори: 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1070, 1071, 1072, 1073, 1074, 1075, 1076, 1077, 1078, 1079, 1080, 1081, 1082, 1083, 1084, 1085, 1086, 1087, 1088, 1089, 1090, 1091, 1092, 1093, 1094, 1095, 1096, 1097, 1098, 1099, 1100, 1101, 1102, 1103, 1104, 1105, 1106, 1107, 1108, 1109, 1110, 1111, 1112, 1113, 1114, 1115, 1116, 1117, 1118, 1119, 1120, 1121, 1122, 1123, 1124, 1125, 1126, 1127, 1128, 1129, 1130, 1131, 1132, 1133, 1134, 1135, 1136, 1137, 1138, 1139, 1140, 1141, 1142, 1143, 1144, 1145, 1146, 1147, 1148, 1149, 1150, 1151, 1152, 1153, 1154, 1155, 1156, 1157, 1158, 1159, 1160, 1161, 1162, 1163, 1164, 1165, 1166, 1167, 1168, 1169, 1170, 1171, 1172, 1173, 1174, 1175, 1176, 1177, 1178, 1179, 1180, 1181, 1182, 1183, 1184, 1185, 1186, 1187, 1188, 1189, 1190, 1191, 1192, 1193, 1194, 1195, 1196, 1197, 1198, 1199, 1200, 1201, 1202, 1203, 1204, 1205, 1206, 1207, 1208, 1209, 1210, 1211, 1212, 1213, 1214, 1215, 1216, 1217, 1218, 1219, 1220, 1221, 1222, 1223, 1224, 1225, 1226, 1227, 1228, 1229, 1230, 1231, 1232, 1233, 1234, 1235, 1236, 1237, 1238, 1239, 1240, 1241, 1242, 1243, 1244, 1245, 1246, 1247, 1248, 1249, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1258, 1259, 1260, 1261, 1262, 1263, 1264, 1265, 1266, 1267, 1268, 1269, 1270, 1271, 1272, 1273, 1274, 1275, 1276, 1277, 1278, 1279, 1280, 1281, 1282, 1283, 1284, 1285, 1286, 1287, 1288, 1289, 1290, 1291, 1292, 1293, 1294, 1295, 1296, 1297, 1298, 1299, 1300, 1301, 1302, 1303, 1304, 1305, 1306, 1307, 1308, 1309, 1310, 1311, 1312, 1313, 1314, 1315, 1316, 1317, 1318, 1319, 1320, 1321, 1322, 1323, 1324, 1325, 1326, 1327, 1328, 1329, 1330, 1331, 1332, 1333, 1334, 1335, 1336, 1337, 1338, 1339, 1340, 1341, 1342, 1343, 1344, 1345, 1346, 1347, 1348, 1349, 1350, 1351, 1352, 1353, 1354, 1355, 1356, 1357, 1358, 1359, 1360, 1361, 1362, 1363, 1364, 1365, 1366, 1367, 1368, 1369, 1370, 1371, 1372, 1373, 1374, 1375, 1376, 1377, 1378, 1379, 1380, 1381, 1382, 1383, 1384, 1385, 1386, 1387, 1388, 1389, 1390, 1391, 1392, 1393, 1394, 1395, 1396, 1397, 1398, 1399, 1400, 1401, 1402, 1403, 1404, 1405, 1406, 1407, 1408, 1409, 1410, 1411, 1412, 1413, 1414, 1415, 1416, 1417, 1418, 1419, 1420, 1421, 1422, 1423, 1424, 1425, 1426, 1427, 1428, 1429, 1430, 1431, 1432, 1433, 1434, 1435, 1436, 1437, 1438, 1439, 1440, 1441, 1442, 1443, 1444, 1445, 1446, 1447, 1448, 1449, 1450, 1451, 1452, 1453, 1454, 1455, 1456, 1457, 1458, 1459, 1460, 1461, 1462, 1463, 1464, 1465, 1466, 1467, 1468, 1469, 1470, 1471, 1472, 1473, 1474, 1475, 1476, 1477, 1478, 1479, 1480, 1481, 1482, 1483, 1484, 1485, 1486, 1487, 1488, 1489, 1490, 1491, 1492, 1493, 1494, 1495, 1496, 1497, 1498, 1499, 1500, 1501, 1502, 1503, 1504, 1505, 1506, 1507, 1508, 1509, 1510, 1511, 1512, 1513, 1514, 1515, 1516, 1517, 1518, 1519, 1520, 1521, 1522, 1523, 1524, 1525, 1526, 1527, 1528, 1529, 1530, 1531, 1532, 1533, 1534, 1535, 1536, 1537, 1538, 1539, 1540, 1541, 1542, 1543, 1544, 1545, 1546, 1547, 1548, 1549, 1550, 1551, 1552, 1553, 1554, 1555, 1556, 1557, 1558, 1559, 1560, 1561, 1562, 1563, 1564, 1565, 1566, 1567, 1568, 1569, 1570, 1571, 1572, 1573, 1574, 1575, 1576, 1577, 1578, 1579, 1580, 1581, 1582, 1583, 1584, 1585, 1586, 1587, 1588, 1589, 1590, 1591, 1592, 1593, 1594, 1595, 1596, 1597, 1598, 1599, 1600, 1601, 1602, 1603, 1604, 1605, 1606, 1607, 1608, 1609, 1610, 1611, 1612, 1613, 1614, 1615, 1616, 1617, 1618, 1619, 1620, 1621, 1622, 1623, 1624, 1625, 1626, 1627, 1628, 1629, 1630, 1631, 1632, 1633, 1634, 1635, 1636, 1637, 1638, 1639, 1640, 1641, 1642, 1643, 1644, 1645, 1646, 1647, 1648, 1649, 1650, 1651, 1652, 1653, 1654, 1655, 1656, 1657, 1658, 1659, 1660, 1661, 1662, 1663, 1664, 1665, 1666, 1667, 1668, 1669, 1670, 1671, 1672, 1673, 1674, 1675, 1676, 1677, 1678, 1679, 1680, 1681, 1682, 1683, 1684, 1685, 1686, 1687, 1688, 1689, 1690, 1691, 1692, 1693, 1694, 1695, 1696, 1697, 1698, 1699, 1700, 1701, 1702, 1703, 1704, 1705, 1706, 1707, 1708, 1709, 1710, 1711, 1712, 1713, 1714, 1715, 1716, 1717, 1718, 1719, 1720, 1721, 1722, 1723, 1724, 1725, 1726, 1727, 1728, 1729, 1730, 1731, 1732, 1733, 1734, 1735, 1736, 1737, 1738, 1739, 1740, 1741, 1742, 1743, 1744, 1745, 1746, 1747, 1748, 1749, 1750, 1751, 1752, 1753, 1754, 1755, 1756, 1757, 1758, 1759, 1760, 1761, 1762, 1763, 1764, 1765, 1766, 1767, 1768, 1769, 1770, 1771, 1772, 1773, 1774, 1775, 1776, 1777, 1778, 1779, 1780, 1781, 1782, 1783, 1784, 1785, 1786, 1787, 1788, 1789, 1790, 1791, 1792, 1793, 1794, 1795, 1796, 1797, 1798, 1799, 1800, 1801, 1802, 1803, 1804, 1805, 1806, 1807, 1808, 1809, 1810, 1811, 1812, 1813, 1814, 1815, 1816, 1817, 1818, 1819, 1820, 1821, 1822, 1823, 1824, 1825, 1826, 1827, 1828, 1829, 1830, 1831, 1832, 1833, 1834, 1835, 1836, 1837, 1838, 1839, 1840, 1841, 1842, 1843, 1844, 1845, 1846, 1847, 1848, 1849, 1850, 1851, 1852, 1853, 1854, 1855, 1856, 1857, 1858, 1859, 1860, 1861, 1862, 1863, 1864, 1865, 1866, 1867, 1868, 1869, 1870, 1871, 1872, 1873, 1874, 1875, 1876, 1877, 1878, 1879, 1880, 1881, 1882, 1883, 1884, 1885, 1886, 1887, 1888, 1889, 1890, 1891, 1892, 1893, 1894, 1895, 1896, 1897, 1898, 1899, 1900, 1901, 1902, 1903, 1904, 1905, 1906, 1907, 1908, 1909, 1910, 1911, 1912, 1913, 1914, 1915, 1916, 1917, 1918, 1919, 1920, 1921, 1922, 1923, 1924, 1925, 1926, 1927, 1928, 1929, 1930, 1931, 1932, 1933, 1934, 1935, 1936, 1937, 1938, 1939, 1940, 1941, 1942, 1943, 1944, 1945, 1946, 1947, 1948, 1949, 1950, 1951, 1952, 1953, 1954, 1955, 1956, 1957, 1958, 1959, 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057,