

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та охоронних систем

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**ІНТЕРАКТИВНА МОДЕЛЬ СОНЯЧНОЇ БАТАРЕЇ ДЛЯ
ЖИВЛЕННЯ ІОТ-СИСТЕМИ ЗАСОБАМИ MATLAB**

**INTERACTIVE MODEL OF A SOLAR BATTERY FOR POWERING
AN IoT SYSTEM USING MATLAB**

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи KI-42
Попко Андрій Олександрович

(підпис)

Керівник:
к.т.н., доцент
Пех Петро Антонович

(підпис)

Кваліфікаційну роботу
допущено до захисту
« » червня 2026 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2026 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Т. Терлецький

« 23 » 12 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Попку Андрію Олександровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Інтерактивна модель сонячної батареї для живлення IoT-системи засобами Matlab

Керівник роботи к.т.н., доцент Пех Петро Антонович

затверджені наказом закладу вищої освіти від «20» грудня 2026 року № 536/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 28.05.2026 р.

3. Вихідні дані до роботи Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області, різні інтернет-ресурси технічного спрямування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Вступ

Огляд існуючих рішень

Методи та інструменти вирішення питань дослідження

Розроблення апаратної та програмної частини проекту

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Принцип роботи сонячного елемента на основі p-n переходу

Вольт-амперна характеристика сонячного елемента

Інтерактивний інтерфейс керування параметрами системи ВДЕ

Оптимізована система ВДЕ з акумулятором

Кадр покадрової анімації роботи системи ВДЕ

Результат оптимізації покриття навантаження від ВДЕ

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Огляд існуючих рішень	Пех П.А., доцент		
Методи та інструменти вирішення питань дослідження	Пех П.А., доцент		
Розроблення апаратної та програмної частини проекту	Пех П.А., доцент		
Нормоконтроль	Багнюк Н. В., доцент		
Гарант ОП	Лавренчук С. В., доцент		
Показник запозичень тексту	_____ %		
Академічна доброчесність	Міскевич О. І., ст. викладач		

7. Дата видачі завдання 23.12.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Огляд літератури із досліджуваної проблеми, аналіз предметної області та наявних рішень	до 10.02.2026 р.	
2.	Огляд існуючих рішень	до 02.03.2026 р.	
3.	Методи та інструменти вирішення питань дослідження	до 02.04.2026 р.	
4.	Розроблення апаратної та програмної частини проекту	до 10.04.2026 р.	
5.	Представлення остаточного варіанту кваліфікаційної роботи керівникові	до 01.05.2026 р.	
6.	Нормоконтроль	до 23.05.2026 р.	
7.	Інструментальна перевірка на академічний плагіат	до 26.05.2026 р.	
8.	Здача кваліфікаційної роботи та всіх супровідних документів на кафедру	до 28.05.2026 р.	

Здобувач вищої освіти

(підпис)

Андрій ПОПКО

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Петро ПЕХ

(прізвище, ініціали)

АНОТАЦІЯ

Попко А. О. Інтерактивна модель сонячної батареї для живлення IoT-системи засобами MatLab.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2026.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел.

Перший розділ присвячено огляду предметної області: обґрунтовано актуальність теми, розглянуто фізичні основи перетворення сонячної енергії, проаналізовано однодіодну та дводіодну моделі фотоелектричних перетворювачів, досліджено алгоритми відстеження точки максимальної потужності (P&O, Incremental Conductance, Fuzzy Logic) та розглянуто існуючі IoT-рішення для моніторингу сонячних електростанцій.

У другому розділі здійснено вибір засобів розробки: розглянуто можливості MATLAB/Simulink та бібліотеки Simscape Electrical, обґрунтовано вибір модуля KC200GT як об'єкта моделювання, описано принцип роботи ESP32 та математичний опис однодіодної моделі.

Третій розділ присвячено практичній реалізації восьми MATLAB-програм системи відновлюваної генерації – від базової оптимізації покриття навантаження сонячною панеллю та вітровою турбіною до добової динаміки з акумулятором, анімації та інтерактивних інтерфейсів із графічними повзунками. Наведено результати симуляції у вигляді графіків покриття навантаження, динаміки заряду акумулятора та залишкового дефіциту потужності.

Ключові слова: сонячна панель, вітрова турбіна, відновлювані джерела енергії, MATLAB, оптимізація, акумулятор, анімація, інтерактивна модель.

ANNOTATION

Popko A. Interactive model of a solar battery for powering an IoT system using MATLAB. Qualifying work of a bachelor of EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2026.

The qualification work consists of an introduction, three sections, conclusions, a list of references.

The first section provides an overview of the subject area: the relevance of the topic is substantiated, the physical principles of solar energy conversion are examined, the single-diode and double-diode models of photovoltaic converters are analyzed, maximum power point tracking algorithms (P&O, Incremental Conductance, Fuzzy Logic) are investigated, and existing IoT solutions for solar power plant monitoring are reviewed.

The second section covers the selection of development tools: the capabilities of MATLAB/Simulink and the Simscape Electrical library are examined, the KC200GT module is justified as the object of modeling, the operating principles of the ESP32 microcontroller and the mathematical framework of the single-diode model are described.

The third section is devoted to the practical implementation of eight MATLAB programs for a renewable energy generation system – ranging from basic load coverage optimization using a solar panel and wind turbine to daily dynamics with a battery storage unit, animation, and interactive interfaces with graphical sliders. Simulation results are presented in the form of load coverage graphs, battery charge dynamics, and residual power deficit.

Keywords: solar panel, wind turbine, renewable energy sources, MATLAB, optimization, battery storage, animation, interactive model.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	8
1.1 Принципи роботи фотоелектричних перетворювачів.....	8
1.2 Моделі сонячних батарей у наукових дослідженнях	15
1.3 Методи максимізації енерговіддачі (MPPT-алгоритми).....	17
1.4 Існуючі підходи до моделювання у Matlab/Simulink	20
1.5 Вимоги IoT-систем до джерел живлення	22
РОЗДІЛ 2 МЕТОДИ ТА ІНСТРУМЕНТИ ВИРІШЕННЯ ПИТАНЬ	
ДОСЛІДЖЕННЯ	26
2.1 Постановка задачі моделювання	26
2.2 Математична модель сонячної батареї	28
2.3 Модель споживача (IoT-навантаження)	32
2.4 Алгоритми керування та оптимізації	34
РОЗДІЛ 3 РОЗРОБЛЕННЯ АПАРАТНОЇ ТА ПРОГРАМНОЇ ЧАСТИНИ	
ПРОЕКТУ	37
3.1 Програма моделювання та оптимізації ВДЕ	37
3.2 Програма моделювання системи генерації з акумулятором	39
3.3 Програма динамічного моделювання системи ВДЕ з акумулятором	42
3.4 Програма покадрової анімації роботи системи ВДЕ.....	45
3.5 Програма плавної анімації системи ВДЕ з інтерполяцією	48
3.6 Програма інтерактивної моделі системи ВДЕ з керуванням параметрами.....	50
3.7 Програма інтерактивного моделювання системи ВДЕ з вибором години	54
3.8 Програма плавної анімації ізольованої системи з акумулятором.....	57
ВИСНОВКИ.....	60
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61

ВСТУП

Актуальність теми. Зростання кількості IoT-пристроїв з автономним живленням актуалізує задачу моделювання фотоелектричних систем за змінних умов роботи. Попри широкі можливості MATLAB, типові моделі є статичними, що обмежує їх практичну цінність. Розробка інтерактивної моделі сонячної батареї з динамічною зміною освітленості та навантаження безпосередньо під час симуляції усуває цей недолік.

Метою роботи є розроблення інтерактивної імітаційної моделі сонячної батареї у середовищі MATLAB/Simulink для живлення IoT-системи з можливістю динамічної зміни параметрів освітленості та навантаження під час симуляції.

Об'єкт дослідження – фотоелектричні системи для автономного живлення IoT-пристроїв.

Предмет дослідження – інтерактивна імітаційна модель сонячної батареї, побудована засобами MATLAB/Simulink.

Завдання роботи:

- дослідити принципи роботи фотоелектричних перетворювачів, однодіодну та дводіодну моделі, а також алгоритми відстеження точки максимальної потужності;
- реалізувати інтерактивну модель із можливістю динамічної зміни рівня освітленості та параметрів навантаження безпосередньо під час симуляції;
- візуалізувати результати моделювання у вигляді графіків вихідної напруги, струму та потужності за різних умов освітленості;
- запропонувати серію MATLAB-програм для аналізу роботи системи відновлюваної генерації з акумулятором та оцінки енергетичного балансу.

Методи дослідження: теоретичний аналіз наукової літератури, математичне моделювання фотоелектричного перетворювача, комп'ютерне імітаційне моделювання у MATLAB, порівняльний аналіз MPPT-алгоритмів.

РОЗДІЛ 1

ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Принципи роботи фотоелектричних перетворювачів

В основі роботи фотоелектричних перетворювачів лежить фотоелектричний ефект – фізичне явище, при якому енергія фотонів сонячного випромінювання перетворюється безпосередньо на електричну енергію. Принцип дії сучасних сонячних елементів базується на напівпровідниковому р-n переході, утвореному двома шарами кремнію з різним типом легування. Шар n-типу отримують шляхом введення домішок фосфору, що створює надлишок вільних електронів, тоді як шар р-типу формується легуванням бором, що зумовлює надлишок позитивних носіїв заряду – так званих дірок. На межі цих двох шарів виникає внутрішнє електричне поле, яке є ключовим елементом для перетворення світлової енергії в електричну.

Процес генерації електрики відбувається у декілька послідовних етапів. Коли фотон сонячного випромінювання потрапляє на поверхню напівпровідника, він поглинається матеріалом за умови, що його енергія перевищує значення забороненої зони (bandgap) даного напівпровідника. Для кремнію ця величина становить 1,11 eV за кімнатної температури. Поглинута енергія фотона передається електрону валентної зони, виводячи його до зони провідності та залишаючи після себе дірку – так відбувається генерація електронно-діркової пари. Внутрішнє електричне поле р-n переходу розділяє ці заряди: електрони переміщуються до шару n-типу, а дірки – до шару р-типу. Якщо до зовнішніх контактів сонячного елемента підключити навантаження, утворений потенціал спричинить протікання електричного струму через зовнішній ланцюг (рис. 1.1) [1].

Варто зазначити, що не всі фотони сонячного спектра беруть однакову участь у процесі генерації. Фотони з енергією, меншою за ширину забороненої зони, проходять крізь матеріал без поглинання та не виробляють електричної енергії.

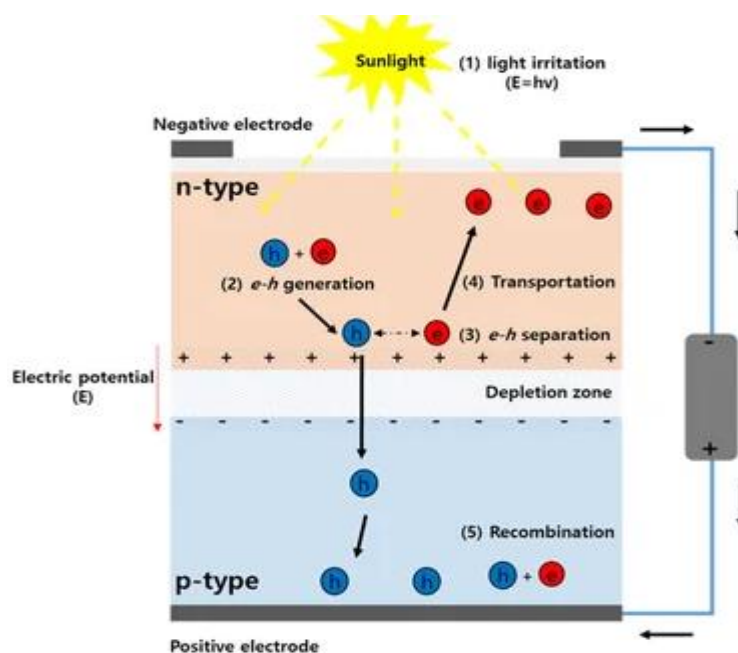


Рисунок 1.1 – Принцип роботи сонячного елемента на основі р-п переходу [2]

Фотони з надмірно високою енергією генерують електронно-діркові пари, проте надлишок енергії розсіюється у вигляді тепла, що знижує загальний ККД перетворення. Саме ця фізична обмеженість однопереходних сонячних елементів зумовила теоретичну межу ефективності для кремнієвих фотоелементів – близько 30 % згідно з межею Шоклі-Квайссера.

Конструктивно сонячний елемент являє собою тонку пластину р-п переходу великої площі з металевими контактами з обох боків. Верхній контакт виконується у вигляді тонкої металевої сітки, що пропускає світло до активного шару, тоді як нижній контакт є суцільною металевою пластиною. Для зменшення відбиття світла від поверхні наносять антивідбивне покриття, яке суттєво підвищує кількість поглинутих фотонів. Окремі сонячні елементи з'єднують послідовно та паралельно, формуючи сонячні модулі (панелі), а з модулів складають масиви (батареї) потрібної потужності. При послідовному з'єднанні елементів підсумовуються напруги, при паралельному – струми, що дозволяє гнучко формувати вихідні електричні характеристики фотоелектричної системи.

Ефективність перетворення сонячної енергії в електричну залежить від цілої низки факторів. Температура є одним із найвагоміших: підвищення температури фотоелемента зменшує ширину забороненої зони, що призводить до зниження

напруги холостого ходу та, відповідно, вихідної потужності. Інтенсивність сонячного випромінювання (освітленість) безпосередньо визначає кількість генерованих носіїв заряду i , отже, величину струму короткого замикання. Розуміння цих фізичних залежностей є фундаментом для побудови математичних моделей сонячних панелей, які дозволяють прогнозувати поведінку фотоелектричної системи в умовах змінних зовнішніх факторів.

На сьогодні у промисловому виробництві та наукових дослідженнях використовуються три основні типи фотоелектричних елементів: монокристалічні, полікристалічні та тонкоплівкові. Кожен із них відрізняється технологією виготовлення, ефективністю перетворення, вартістю та умовами застосування, що визначає їх придатність для різних завдань, зокрема для живлення автономних IoT-систем.

Монокристалічні сонячні елементи виготовляються з кремнію, вирощеного у вигляді єдиного безперервного кристала методом Чохральського. Завдяки однорідній кристалічній структурі такі елементи забезпечують найвищу ефективність серед комерційно доступних типів – від 20 % до 23 % і вище в сучасних зразках. Висока ефективність означає, що для отримання заданої потужності потрібна менша площа панелі, що є суттєвою перевагою при обмеженому просторі. Водночас складність технологічного процесу вирощування монокристала робить такі елементи найдорожчими серед представлених типів. Для малопотужних IoT-систем, де критичною є компактність, монокристалічні панелі є оптимальним вибором [3].

Полікристалічні сонячні елементи отримують шляхом плавлення кремнієвої сировини та її наступного охолодження у формах, внаслідок чого утворюється структура з безлічі дрібних кристалів різної орієнтації. Характерна блакитно-строката поверхня є відмінною ознакою таких панелей. Ефективність полікристалічних елементів дещо нижча – як правило, від 15 % до 18 % – межі між кристалами перешкоджають вільному руху носіїв заряду та збільшують рекомбінаційні втрати. Проте спрощена технологія виробництва робить полікристалічні панелі значно дешевшими, що забезпечує їм широке застосування

у великих фотоелектричних установках, де вартість одиниці потужності є пріоритетним критерієм.

Тонкоплівкові сонячні елементи являють собою принципово інший технологічний підхід: надтонкий шар фотоактивного матеріалу наноситься на підкладку зі скла, металу або гнучкого полімеру. Як активні матеріали використовуються аморфний кремній (a-Si), телурид кадмію (CdTe) або мідь-індій-галій-селенід (CIGS). Комерційно доступні тонкоплівкові панелі мають ефективність у діапазоні 10-13 %, що суттєво поступається кристалічним аналогам. Однак їх ключовими перевагами є гнучкість конструкції, менша матеріаломісткість та здатність зберігати відносно стабільну продуктивність за умов дифузного освітлення й підвищених температур. Ці властивості відкривають перспективи для застосування тонкоплівкових елементів у нестандартних конструктивах IoT-пристроїв, де потрібне конформне розміщення джерела живлення.

Порівняння трьох типів сонячних елементів за ключовими параметрами наведено у таблиці 1.1. Як видно з таблиці, для живлення малопотужних IoT-вузлів найбільш обґрунтованим вибором є монокристалічні панелі з огляду на їх високу питому потужність та компактність. Саме цей тип елементів використовується як основа для моделювання у даній роботі, зокрема параметри реальної монокристалічної панелі слугують вхідними даними для побудованої моделі [3].

Електрична поведінка фотоелектричного елемента описується за допомогою двох взаємопов'язаних характеристик: вольт-амперної (I-V) кривої та кривої потужності (P-V).

Вольт-амперна характеристика є графічним відображенням залежності між вихідним струмом і напругою сонячного елемента за конкретних умов освітленості та температури.

Вона містить три ключові параметри: струм короткого замикання (I_{sc}) – максимальний струм при нульовій напрузі, напруга холостого ходу (V_{oc}) – максимальна напруга при нульовому струмі, та точка максимальної потужності (MPP – Maximum Power Point), у якій добуток струму та напруги є найбільшим.

Таблиця 1.1 – Порівняльна характеристика типів сонячних елементів [3]

Параметр	Монокристалічні	Полікристалічні	Тонкоплівкові
ККД, %	20-23	15-18	10-13
Вартість	Висока	Середня	Низька
Технологія	Метод Чохральського	Лиття у форми	Нанесення на підкладку
Поведінка при слабкому світлі	Задовільна	Задовільна	Добра
Гнучкість конструкції	Ні	Ні	Так
Придатність для IoT	Оптимальна	Прийнятна	Обмежена

Крива потужності $P-V$ будується як добуток $I \cdot V$ для кожної точки вольт-амперної характеристики і має характерний максимум, що відповідає точці MPP (рис. 1.2).

Точка максимальної потужності є центральним поняттям при проектуванні фотоелектричних систем. У цій точці сонячна панель віддає найбільшу можливу потужність за даних зовнішніх умов. Координати точки MPP – напруга V_{mpp} та струм I_{mpp} – залежать від поточних значень освітленості та температури і постійно змінюються протягом доби. Саме тому в реальних системах застосовують спеціальні алгоритми відстеження точки максимальної потужності (MPPT), які безперервно коригують робочу точку системи, забезпечуючи максимальну енергетичну ефективність незалежно від умов навколишнього середовища.

Освітленість (інсоляція) чинить визначальний вплив на вихідні характеристики сонячного елемента.

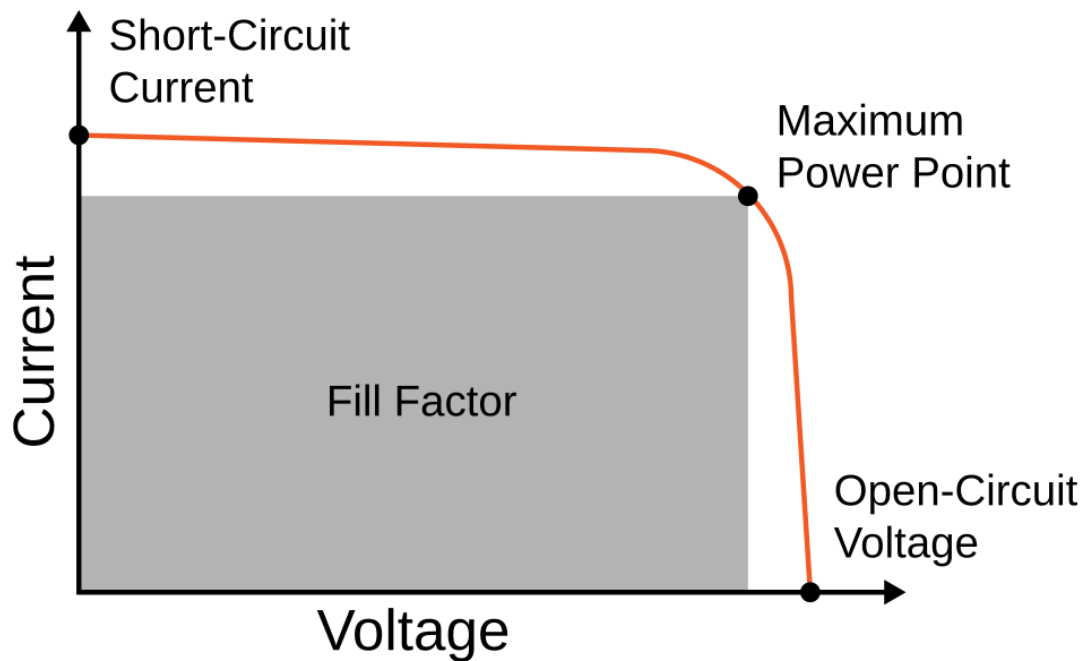


Рисунок 1.2 – Вольт-амперна характеристика сонячного елемента [4]

Освітленість (інсоляція) чинить визначальний вплив на вихідні характеристики сонячного елемента. Зі зростанням інтенсивності сонячного випромінювання струм короткого замикання I_{sc} збільшується практично лінійно (рис. 1.3), оскільки більша кількість фотонів генерує більше електронно-діркових пар. Напруга холостого ходу V_{oc} також зростає зі збільшенням освітленості, але значно повільніше – логарифмічно. В результаті максимальна потужність MPP зростає зі збільшенням освітленості, а крива I-V зміщується вгору по осі струму. Стандартні умови тестування (STC) передбачають освітленість 1000 Вт/м^2 при температурі елемента 25°C – саме за цих умов визначаються паспортні характеристики панелей.

Температура є другим за значимістю фактором, що впливає на роботу фотоелектричного елемента, і її вплив є переважно негативним. Підвищення температури елемента призводить до суттєвого зниження напруги холостого ходу V_{oc} , тоді як струм короткого замикання I_{sc} змінюється незначно (рис. 1.4). Оскільки зниження напруги є більш вираженим, загальна вихідна потужність зменшується зі зростанням температури. Кількісно цей ефект описується температурним коефіцієнтом напруги, типові значення якого для кремнієвих панелей становить близько $-0,3 \dots -0,5 \%$ на кожен градус Цельсія понад 25°C .

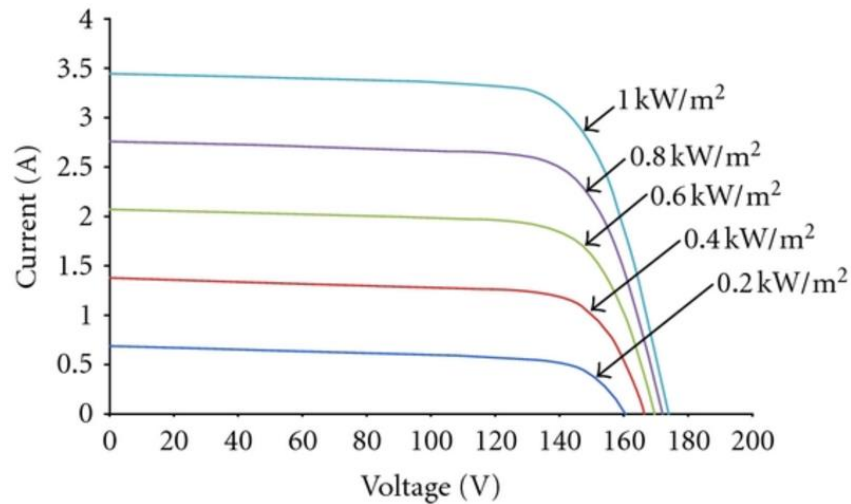


Рисунок 1.3 – I-V характеристики сонячного елемента при різних освітленості [5]

Це означає, що в спекотний літній день реальна потужність панелі може бути на 10-20 % нижчою за паспортну. Для моделювання систем живлення IoT-пристроїв урахування температурних залежностей є обов'язковим, оскільки визначає точність розрахунку енергетичного балансу системи [6].

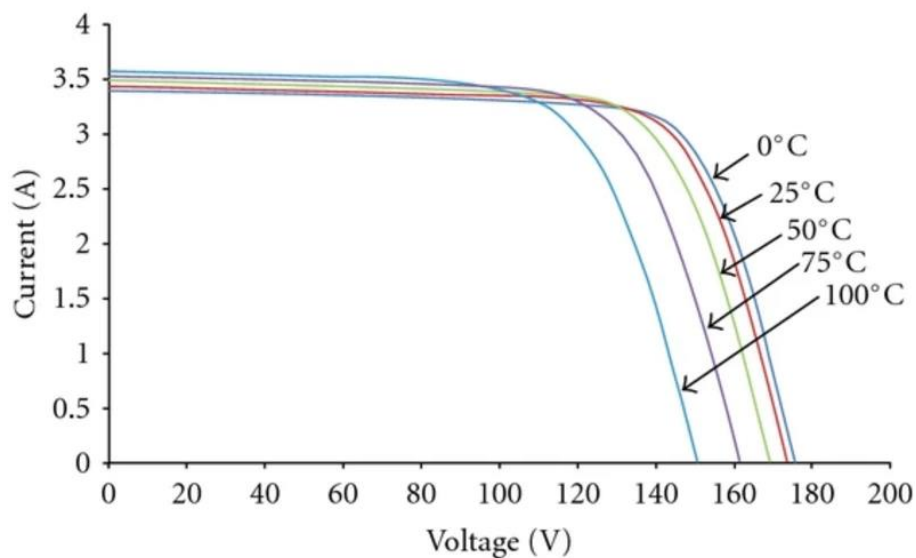


Рисунок 1.4 – I-V характеристики сонячного елемента при різних температурі [5]

Ще одним параметром, що характеризує якість сонячного елемента, є коефіцієнт заповнення (Fill Factor, FF) – відношення максимальної потужності MPP до добутку $V_{oc} \cdot I_{sc}$ і визначається за формулою (1.1):

$$FF = \frac{P_{mpp}}{V_{oc} \cdot I_{sc}}, \quad (1.1)$$

де FF – коефіцієнт заповнення;

P_{mpp} – потужність у точці максимальної потужності, Вт;

V_{oc} – напруга холостого ходу, В;

I_{sc} – струм короткого замикання, А.

Коефіцієнт заповнення відображає «прямокутність» I-V кривої: чим ближче форма характеристики до прямокутника, тим вищий коефіцієнт заповнення і тим ефективніше елемент використовує доступну енергію. Для якісних кремнієвих панелей FF зазвичай знаходиться в діапазоні 0,7-0,85. Сукупність параметрів V_{oc} , I_{sc} , V_{mpp} , I_{mpp} та FF повністю описує електричну поведінку сонячного елемента і є вхідними даними для його математичної моделі.

1.2 Моделі сонячних батарей у наукових дослідженнях

Математичне моделювання фотоелектричних перетворювачів базується на представленні сонячного елемента у вигляді еквівалентної електричної схеми. Найбільш поширеними підходами є однодіодна та дводіодна моделі, які відрізняються між собою кількістю параметрів, точністю відтворення реальних характеристик та обчислювальною складністю. Вибір між цими моделями визначається вимогами до точності симуляції та обчислювальними ресурсами, доступними для розрахунку.

Однодіодна модель (Single Diode Model, SDM) є найбільш застосовуваною у практиці моделювання фотоелектричних систем. Еквівалентна схема цієї моделі складається з керованого джерела фотоструму I_{ph} , діода, що представляє p-n

перехід, послідовного опору R_s , який враховує омичні втрати в контактах і напівпровіднику, та шунтуючого опору R_{sh} , що моделює струми витоку через дефекти структури. Вихідний струм елемента описується нелінійним рівнянням, що містить п'ять невідомих параметрів: I_{ph} , струм насичення діода I_0 , коефіцієнт ідеальності діода n , R_s та R_{sh} . Незважаючи на певні спрощення, однодіодна модель забезпечує достатню точність для більшості інженерних розрахунків і є основою для моделювання у середовищі MATLAB завдяки відносній простоті реалізації. Вихідний струм однодіодної моделі описується формулою (1.2):

$$I = I_{ph} - I_0 \left(\exp \left(\frac{V + IR_s}{nV_T} \right) - 1 \right) - \frac{V + IR_s}{R_{sh}}, \quad (1.2)$$

де I – вихідний струм елемента, А;

I_{ph} – фотострум, А;

I_0 – струм насичення діода, А;

V – вихідна напруга, В;

R_s – послідовний опір, Ом;

n – коефіцієнт ідеальності діода;

V_t – термічна напруга, В;

R_{sh} – шунтуючий опір, Ом [7].

Дводіодна модель (Double Diode Model, DDM) є розширенням однодіодної та враховує два різні механізми рекомбінації носіїв заряду в р-п переході. Перший діод описує дифузійний струм рекомбінації, а другий – рекомбінацію в збідненому шарі.

Це дає моделі сім параметрів замість п'яти, що суттєво ускладнює процедуру їх ідентифікації. Дводіодна модель демонструє вищу точність порівняно з однодіодною, особливо в умовах низької освітленості, коли рекомбінаційні процеси в збідненому шарі стають більш значущими. Еквівалентні схеми однодіодної та дводіодної моделей сонячного елемента наведено на рисунку 1.5.

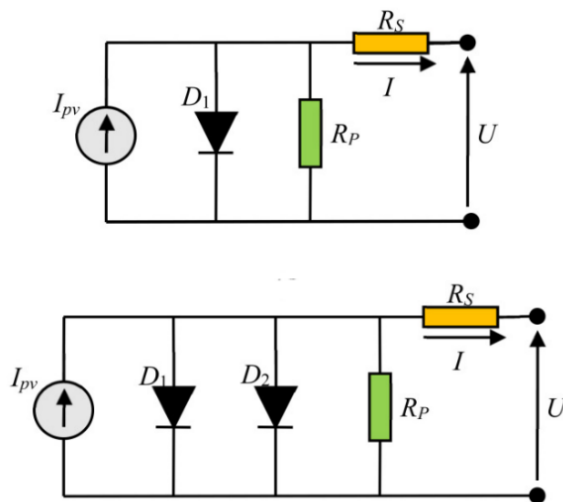


Рисунок 1.5 – Еквівалентні схеми однодіодної та дводіодної моделей сонячного елемента [8]

Однак підвищена точність досягається ціною зростання обчислювальних витрат і складності алгоритму розв’язання нелінійної системи рівнянь.

Вибір між однодіодною та дводіодною моделями залежить від конкретного завдання моделювання. Для задач планування енергосистем та інженерного проектування, де важлива швидкість розрахунку, однодіодна модель є пріоритетною завдяки меншій кількості параметрів та нижчим обчислювальним витратам. Дводіодна модель доцільна у наукових дослідженнях, де потрібна висока точність відтворення характеристик панелі за широкого діапазону умов освітленості, зокрема при слабкому розсіяному світлі. У контексті даної роботи, де моделюється живлення малопотужного IoT-вузла і пріоритетом є практична реалізованість інтерактивної моделі, обрано однодіодну модель як оптимальний баланс між точністю та обчислювальною ефективністю [7].

1. 3 Методи максимізації енерговіддачі (MPPT-алгоритми)

Алгоритми відстеження точки максимальної потужності (Maximum Power Point Tracking, MPPT) є невід’ємною складовою будь-якої ефективної фотоелектричної системи. Оскільки координати точки MPP постійно змінюються залежно від освітленості та температури, система керування повинна безперервно

коригувати робочу точку панелі, утримуючи її якомога ближче до максимуму потужності. На сьогодні існує велика кількість МРРТ-алгоритмів, що відрізняються складністю реалізації, швидкістю відстеження та ефективністю за мінливих умов навколишнього середовища.

Алгоритм збурення та спостереження (Perturb & Observe, P&O) є найпростішим і найбільш поширеним у практичних застосуваннях (рис. 1.6). Принцип його роботи полягає у періодичному внесенні малого збурення робочої напруги панелі та спостереженні за зміною вихідної потужності. Якщо потужність після збурення зростає – наступне збурення виконується в тому ж напрямку, якщо зменшилась – у протилежному. Алгоритм простий у реалізації та не вимагає знання параметрів панелі, проте має два суттєві недоліки: постійні коливання навколо точки МРР в усталеному режимі та можливість хибного відстеження при різких змінах освітленості. Для малопотужних IoT-систем з відносно стабільними умовами освітлення алгоритм P&O є цілком прийнятним рішенням завдяки мінімальним вимогам до обчислювальних ресурсів [5].

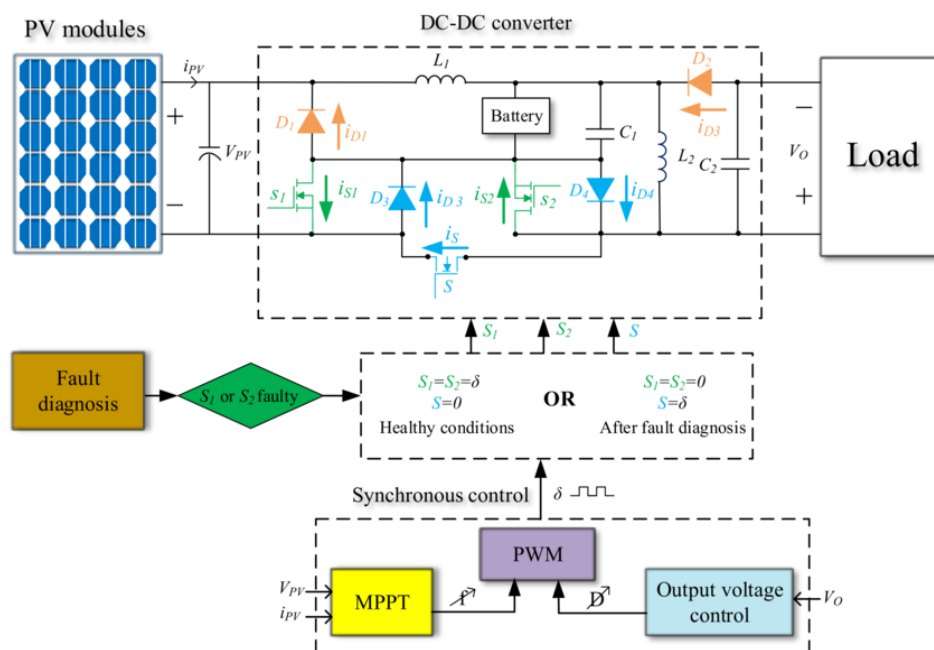


Рисунок 1.6 – Блок-схема алгоритму збурення та спостереження (P&O) [9]

Алгоритм інкрементальної провідності (Incremental Conductance, InC) базується на математичній умові досягнення точки МРР: похідна потужності за

напругою дорівнює нулю в точці максимуму. Це еквівалентно умові рівності миттєвої провідності та її інкременту, взятого зі зворотним знаком. Алгоритм InC точніше визначає положення MPP порівняно з P&O і краще справляється зі швидкими змінами освітленості, оскільки може точно зупинитись у точці максимуму без постійних коливань.

Водночас він вимагає точнішого вимірювання струму та напруги і є дещо складнішим у програмній реалізації. За результатами моделювання алгоритм InC демонструє вищу ефективність відстеження порівняно з P&O, особливо за динамічних умов роботи. Це пояснюється тим, що метод InC безпосередньо аналізує похідну потужності за напругою і здатний точно визначити, з якого боку від точки максимальної потужності перебуває робоча точка. Завдяки цьому за різкої зміни освітленості алгоритм рухається в правильному напрямку без характерних для P&O помилкових кроків. Крім того, у стаціонарному режимі InC забезпечує менші коливання робочої точки навколо MPP, що зменшує втрати потужності на усталеному рівні генерації. Платою за ці переваги є більший обсяг обчислень та підвищені вимоги до точності вимірювальних кіл, що слід враховувати під час вибору алгоритму для конкретної системи.

Алгоритм InC зручно подати у вигляді блок-схеми (рис. 1.7). На кожній ітерації вимірюються напруга і струм панелі та обчислюються прирости ΔV і ΔI . Спочатку перевіряється умова $\Delta V = 0$: якщо вона не виконується, алгоритм порівнює інкрементальну провідність $\Delta I / \Delta V$ з $-I/V$ – рівність означає, що робоча точка перебуває в MPP і напруга лишається незмінною, а знак різниці вказує напрямок її зміщення на крок ϵ . Якщо ж $\Delta V = 0$, рішення приймається за знаком ΔI : за $\Delta I = 0$ напруга не змінюється, інакше її коригують залежно від того, додатний чи від'ємний приріст струму. Після прийняття рішення цикл повертається до наступного вимірювання.

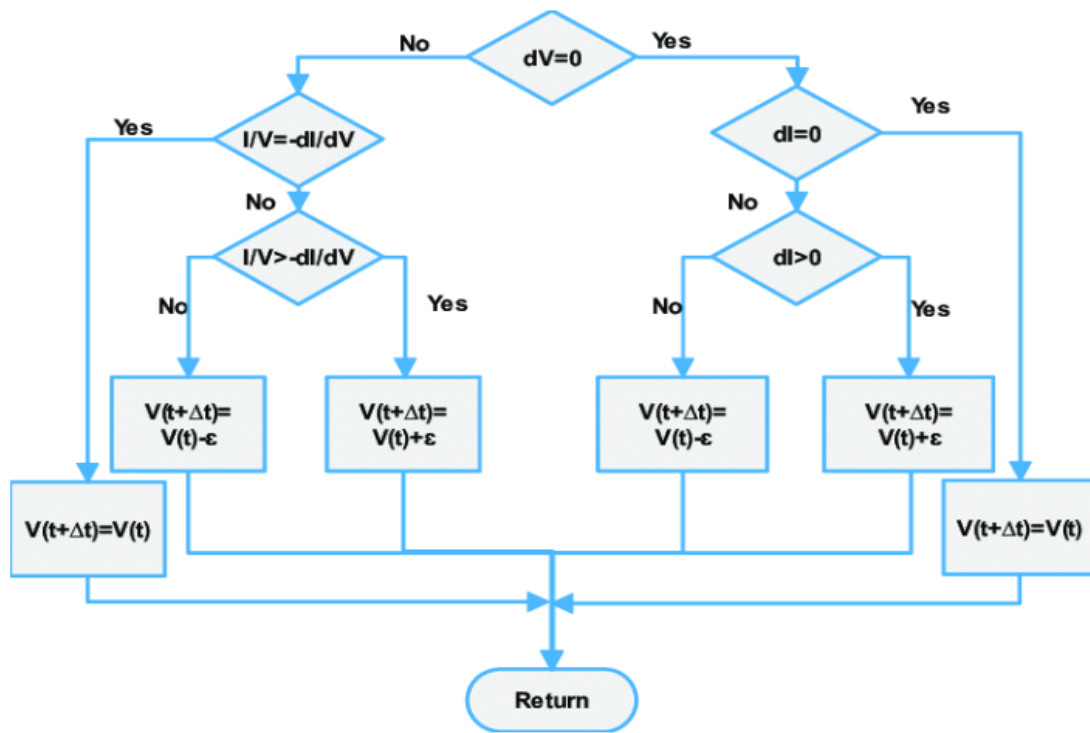


Рисунок 1.7 – Блок-схема алгоритму інкрементальної провідності (InC) [10]

Алгоритми на основі нечіткої логіки (Fuzzy Logic MPPT) та нейронних мереж (Neural MPPT) відносяться до інтелектуальних методів відстеження. Fuzzy Logic контролер не потребує точної математичної моделі системи і може враховувати нелінійні залежності через набір лінгвістичних правил, що робить його стійким до шумів вимірювання та різких змін умов. Нейромережеві методи навчаються на історичних даних і здатні прогнозувати оптимальну робочу точку без ітераційного пошуку. Обидва підходи забезпечують вищу ефективність порівняно з класичними алгоритмами, однак вимагають значно більших обчислювальних ресурсів та складнішої процедури налаштування, що обмежує їх застосування у вбудованих системах з мікроконтролерами низького класу.

1.4 Існуючі підходи до моделювання у Matlab/Simulink

Середовище MATLAB/Simulink є одним із найбільш поширених інструментів для моделювання фотоелектричних систем у наукових дослідженнях та інженерній практиці. Ключову роль у цьому відіграє бібліотека Simscape Electrical, що входить до складу пакету Specialized Power Systems. Вона містить

готовий блок PV Array, який реалізує п'ятипараметрову однодіодну модель сонячної панелі з урахуванням залежностей від освітленості та температури. Блок дозволяє обирати параметри з бази даних реальних панелей Національної лабораторії відновлюваної енергетики США (NREL) або задавати власні характеристики вручну на основі паспортних даних виробника. Додатково бібліотека містить блоки DC-DC перетворювачів, вимірювання струму та напруги, акумуляторних батарей і різноманітних типів навантажень, що забезпечує можливість побудови повноцінної системи живлення в єдиному середовищі моделювання [1, 11, 12].

Окрім стандартних блоків Simscape Electrical, для моделювання PV-систем у MATLAB/Simulink широко застосовуються підсистеми на основі математичних рівнянь, реалізовані за допомогою функціональних блоків Fcn, MATLAB Function та блоків алгебраїчних операцій. Такий підхід дає розробнику повний контроль над параметрами моделі та забезпечує прозорість математичного опису, що є важливим у навчальних та дослідницьких цілях. Бібліотека MATLAB Central File Exchange містить значну кількість відкритих моделей сонячних елементів і панелей різного рівня складності, серед яких широко цитована модель Saad Motahhir, що реалізує однодіодну модель трьома різними способами і слугує відправною точкою для багатьох досліджень [7, 13].

У науковій літературі представлено широкий спектр готових Simulink-моделей фотоелектричних систем різного призначення. Найпростіші моделі обмежуються відтворенням I-V та P-V характеристик окремого елемента або модуля за змінних умов освітленості та температури. Більш складні моделі включають MPPT-контролер на основі алгоритму P&O або InC, DC-DC boost-конвертер та резистивне навантаження, що дозволяє досліджувати ефективність відстеження максимальної потужності за динамічних умов. Найбільш комплексні рішення, релевантні для даної роботи, включають повний ланцюг від PV-масиву до специфічного навантаження з урахуванням акумулятора та контролера заряду. Зокрема, в роботі з моделювання IoT-орієнтованої PV-системи засобами MATLAB реалізовано модель з

ESP32-подібним інтерфейсом моніторингу, що охоплює PV-масив, MPPT-контролер, акумуляторну батарею та однофазний інвертор.

Незважаючи на широкі можливості MATLAB/Simulink для моделювання фотоелектричних систем, більшість існуючих моделей орієнтовані на великі промислові або побутові установки і не враховують специфіки малопотужних IoT-навантажень. Зокрема, стандартні моделі не відтворюють імпульсний характер споживання енергії сенсорними вузлами, не моделюють чергування режимів активності та сну, а також не містять інтерактивного інтерфейсу для оперативного дослідження впливу параметрів на поведінку системи. Саме ці прогалини визначають напрям розробки власної моделі в даній роботі: створення інтерактивної Simulink-моделі з графічним інтерфейсом на базі App Designer, яка специфічно адаптована до задачі живлення IoT-вузла з циклічним режимом роботи.

1.5. Вимоги IoT-систем до джерел живлення

Проектування джерела живлення для IoT-системи неможливе без детального розуміння енергетичного профілю навантаження. На відміну від традиційних споживачів з відносно стабільним споживанням, IoT-вузли характеризуються різко імпульсним характером енергоспоживання, що визначається чергуванням режимів активної роботи та глибокого сну. Типовий IoT-вузол на базі мікроконтролера ESP32 у активному режимі з увімкненим Wi-Fi споживає від 80 до 260 мА залежно від інтенсивності радіообміну, тоді як у режимі глибокого сну споживання знижується до 10-30 мкА. Середня потужність за цикл роботи, що включає пробудження, зчитування сенсора, передачу даних та повернення у сон, становить близько 6-8 мВт при типовому циклі вимірювань кожні 5-15 хвилин [14].

Сенсорні вузли IoT-системи складаються з кількох функціональних блоків, кожен з яких вносить свій внесок у загальне енергоспоживання. Мікроконтролер або мікропроцесор є центральним елементом і відповідає за обробку даних та керування периферією. Радіомодуль для бездротової передачі даних є найбільш

енергоємним компонентом системи: передавач Wi-Fi споживає до 350 мА в піковому режимі під час встановлення з'єднання. Датчики температури, вологості, тиску, освітленості – як правило, споживають від одиниць до десятків мікроампер і вносять незначний внесок у загальний енергетичний бюджет. Аналого-цифровий перетворювач та інші периферійні блоки також активні лише під час вимірювального циклу, що додатково обґрунтовує доцільність використання режимів сну для мінімізації середнього споживання. Зведені дані щодо енергоспоживання основних компонентів IoT-вузла на базі ESP32 наведено в таблиці 1.2 [15].

Головною вимогою до джерела живлення IoT-системи на базі сонячної батареї є забезпечення енергетичного балансу між генерацією та споживанням з урахуванням добової та сезонної нерівномірності інсоляції.

Таблиця 1.2 – Енергоспоживання компонентів IoT-вузла ESP32 [14]

Компонент	Режим	Споживання
ESP32 + Wi-Fi	Активний (передача)	80-260 мА
ESP32 + Wi-Fi	Пікове з'єднання	до 350 мА
ESP32	Глибокий сон	10-30 мкА
Датчики	Активний	1-50 мкА
АЦП	Вимірювання	1-2 мА

Оскільки сонячна панель генерує енергію лише протягом світлового дня, обов'язковим елементом системи є акумулятор, що накопичує надлишок денної генерації для живлення вузла в нічний час та в періоди хмарності. Розмір акумулятора визначається добовим енергетичним балансом та вимогами до автономності системи – кількістю днів, протягом яких система повинна функціонувати без сонячного заряджання. Контролер заряду з функцією MPPT забезпечує максимально ефективне заряджання акумулятора та захист від перезаряду і глибокого розряду (рис. 1.9).

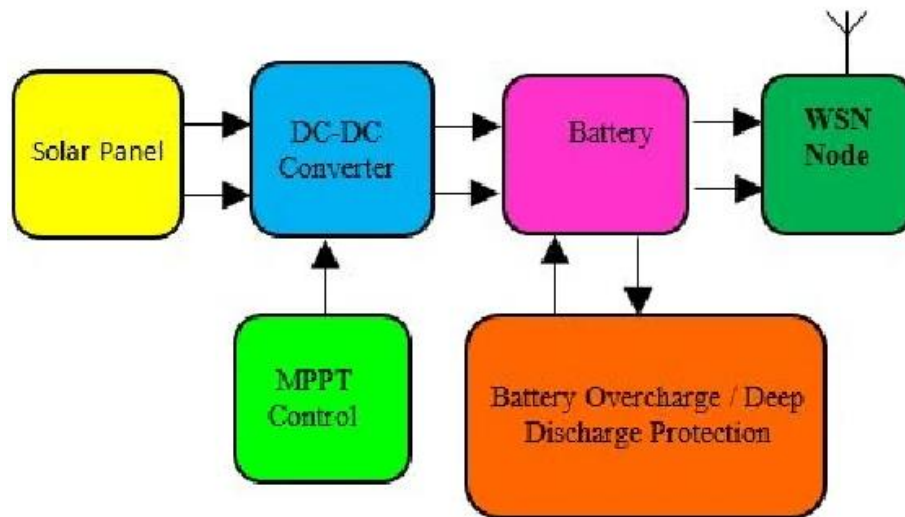


Рисунок 1.9 – Структурна схема системи живлення IoT-вузла на основі сонячної панелі [16]

Напруга живлення є ще одним критичним параметром. Більшість сучасних IoT-мікроконтролерів, зокрема ESP32, працюють від напруги 3,3 В, тоді як акумуляторні батареї забезпечують 3,7-4,2 В (літій-іонні) або 12 В (свинцево-кислотні). Відповідно система живлення повинна включати DC-DC перетворювач, що стабілізує вихідну напругу незалежно від рівня заряду акумулятора та миттєвих піків споживання під час радіопередачі. Стабільність напруги живлення безпосередньо впливає на точність вимірювань аналогових датчиків, тому якість регулювання є важливою технічною вимогою [14].

Практичне застосування сонячних батарей для живлення IoT-систем охоплює широкий спектр задач. Системи моніторингу навколишнього середовища – метеостанції, датчики якості повітря, гідрологічні пости – є найбільш поширеним прикладом, де автономне живлення від сонячної панелі є єдиним практичним рішенням через розміщення в польових умовах. Системи точного землеробства використовують мережі IoT-вузлів для моніторингу вологості ґрунту, температури та інших агрометеорологічних параметрів на великих площах, де прокладання кабельних мереж живлення є економічно недоцільним. Системи моніторингу інфраструктури – мостів, трубопроводів, ліній електропередач – також активно впроваджують автономні IoT-вузли з сонячним живленням для забезпечення

тривалої безперебійної роботи без технічного обслуговування. У всіх перелічених випадках точне моделювання енергетичного балансу системи на етапі проектування є запорукою надійної та ефективної роботи розгорнутої IoT-мережі.

Уваги заслуговують системи відеоспостереження та охорони віддалених об'єктів, де сонячна енергія дозволяє розгортати камери та сенсори руху в місцях без доступу до електромережі. Такі застосування характеризуються нерівномірним споживанням енергії: тривалі періоди очікування з мінімальним навантаженням чергуються з короткочасними піками під час спрацювання та передачі даних. Це висуває підвищені вимоги до ємності акумулятора та точності прогнозування енергетичного балансу, оскільки недооцінка пікового споживання призводить до передчасного розрядження системи в нічний час [17].

Спільною рисою всіх розглянутих застосувань є залежність працездатності системи від співвідношення між генерованою та спожитою енергією протягом доби. Сонячна панель виробляє енергію лише в світлий час, причому її потужність суттєво залежить від погодних умов, пори року та географічного розташування об'єкта. Споживання IoT-вузла відбувається цілодобово, тому надлишок денної генерації повинен накопичуватися в акумуляторі для покриття нічного навантаження та компенсації похмурих днів. Порушення цього балансу – через недостатню потужність панелі, малу ємність акумулятора або неточне врахування реальних умов експлуатації – призводить до періодичних відмов системи, які складно діагностувати та усувати на вже розгорнутій мережі.

РОЗДІЛ 2

МЕТОДИ ТА ІНСТРУМЕНТИ ВИРІШЕННЯ ПИТАНЬ ДОСЛІДЖЕННЯ

2.1 Постановка задачі моделювання

За результатами аналізу, проведеного в першому розділі, встановлено, що існуючі підходи до моделювання фотоелектричних систем у середовищі MATLAB/Simulink не враховують специфіки малопотужних IoT-навантажень і не забезпечують можливості оперативного дослідження поведінки системи в інтерактивному режимі. Метою даного розділу є формулювання задачі моделювання та обґрунтування вибору методів і інструментів для її вирішення. Розроблювана модель повинна відтворювати повний енергетичний ланцюг від фотоелектричного перетворювача до споживача, імітувати циклічний режим роботи IoT-вузла та надавати користувачеві можливість в реальному часі змінювати параметри системи і спостерігати за відповідною реакцією вихідних характеристик.

Як базовий фотоелектричний модуль для моделювання розглянемо монокристалічну панель Kyocera KC200GT потужністю 200 Вт. Вибір обумовлений декількома факторами: параметри цієї панелі присутні у вбудованій базі даних блоку PV Array бібліотеки Simscape Electrical середовища MATLAB/Simulink, що виключає необхідність ручного введення та верифікації параметрів; панель широко використовується у наукових дослідженнях з моделювання фотоелектричних систем, що забезпечує можливість порівняння отриманих результатів з опублікованими даними; потужність 200 Вт є достатньою для забезпечення енергетичних потреб автономного IoT-вузла з урахуванням втрат у системі та необхідності заряджання акумулятора [18]. Паспортні характеристики панелі Kyocera KC200GT за стандартних умов тестування наведено в таблиці 2.1.

Вхідними параметрами моделі є зовнішні умови та характеристики компонентів системи. До зовнішніх умов належать рівень сонячного випромінювання та температура панелі, а характеристики компонентів задаються паспортними параметрами фотомодуля, ємністю акумулятора та профілем споживання IoT-вузла. Вихідними параметрами моделі є струм, напруга й

потужність панелі, а також стан заряду акумулятора, що визначають працездатність системи за заданих умов [19].

Таблиця 2.1 – Паспортні характеристики панелі Kyocera KC200GT [1]

Параметр	Позначення	Значення
Номінальна потужність	P_{max}	200 Вт
Напруга в точці MPP	V_{mpp}	26,3 В
Струм в точці MPP	I_{mpp}	7,61 А
Напруга холостого ходу	V_{oc}	32,9 В
Струм короткого замикання	I_{sc}	8,21 А
Кількість елементів	N_s	54
Температурний коефіцієнт потужності	γ	-0,485 %/° C

Інтенсивність сонячного випромінювання G варіюється в діапазоні 100-1000 Вт/м², температура фотоелектричного елемента T – в діапазоні 0-75° C. Параметри IoT-навантаження визначаються характеристиками ESP32. Споживання в активному режимі з увімкненим Wi-Fi становить 80-260 мА при напрузі живлення 3,3 В, у режимі глибокого сну – близько 10 мкА. Типовий робочий цикл вузла передбачає 3-5 секунд активності для зчитування сенсора та передачі даних і 5-15 хвилин сну між вимірюваннями [20].

Розроблювана модель складається з семи функціональних блоків: фотоелектричної панелі Kyocera KC200GT, DC-DC boost конвертера на основі MOSFET транзистора з діодом Шоткі та LC-фільтром, MPPT контролера що реалізує алгоритм P&O для керування ШІМ сигналом конвертера, акумуляторної батареї Li-Ion для буферного накопичення енергії, контролера заряду для захисту акумулятора від перезаряду та глибокого розряду, IoT-навантаження на базі ESP32 що моделюється керованим джерелом струму з імпульсним профілем, та інтерактивного інтерфейсу з елементами керування та візуалізації [1].

Вихідними параметрами моделі є: вихідний струм I та напруга V сонячної панелі, вихідна потужність P , стан заряду акумулятора (State of Charge, SoC) та енергетичний баланс системи за заданий період симуляції. Вирішення задачі моделювання в інтерактивному режимі дозволяє інженеру-проектувальнику швидко оцінити різні конфігурації системи без необхідності проведення фізичних експериментів [1].

2.2 Математична модель сонячної батареї

Математичний опис фотоелектричного елемента в даній роботі базується на однодіодній моделі, яка є найбільш поширеною у практиці інженерного моделювання завдяки оптимальному балансу між точністю та обчислювальною складністю. Еквівалентна схема заміщення складається з керованого джерела фотоструму I_{ph} , діода D , послідовного опору R_s та шунтуючого опору R_{sh} . Вихідний струм фотоелектричного елемента I визначається як різниця між фотострумом, струмом через діод та струмом витоку через шунтуючий опір і описується таким рівнянням (2.1):

$$I = I_{ph} - I_0 \cdot \left[\exp\left(\frac{V + I \cdot R_s}{n \cdot V_t}\right) - 1 \right] - \frac{V + I \cdot R_s}{R_{sh}}, \quad (2.1)$$

де I – вихідний струм елемента, А;

I_{ph} – фотострум, А;

I_0 – струм насичення діода, А;

V – вихідна напруга елемента, В;

R_s – послідовний опір, Ом;

n – коефіцієнт ідеальності діода;

V_t – термічна напруга, В;

R_{sh} – шунтуючий опір, Ом.

Термічна напруга визначається за формулою (2.2):

$$V_t = \frac{k \cdot T}{q}, \quad (2.2)$$

де k – стала Больцмана, $1,381 \cdot 10^{-23}$ Дж/К;

T – абсолютна температура елемента, К;

q – заряд електрона, $1,602 \cdot 10^{-19}$ Кл.

Наведене рівняння є нелінійним та трансцендентним відносно вихідного струму I , що унеможлиблює його аналітичне розв'язання і вимагає застосування ітераційних чисельних методів.

Температурні залежності параметрів моделі. Для коректного відтворення поведінки панелі Kyocera KC200GT за змінних умов навколишнього середовища необхідно врахувати температурні залежності ключових параметрів моделі. Фотострум I_{ph} лінійно залежить від освітленості G та коригується з урахуванням відхилення температури від стандартного значення 25°C через температурний коефіцієнт струму короткого замикання $\mu_{I_{sc}}$, що описується виразом (2.3):

$$I_{ph} = [I_{sc} + \mu_{I_{sc}} \cdot (T - T_{ref})] \cdot \frac{G}{G_0}, \quad (2.3)$$

де I_{sc} – струм короткого замикання за STC, А;

$\mu_{I_{sc}}$ – температурний коефіцієнт струму, А/К;

T – поточна температура, К;

T_{ref} – референсна температура (298,15 К);

G – поточна освітленість, Вт/м²;

G_0 – референсна освітленість (1000 Вт/м²).

За стандартних умов тестування фотострум практично рівний струму короткого замикання I_{sc} . Струм насичення діода I_0 є сильно залежним від температури і визначається через значення за стандартних умов та температурний

коефіцієнт напруги холостого ходу μV_{oc} . Температурна залежність струму насичення описується виразом (2.4):

$$I_0 = I_{0,ref} \cdot \left(\frac{T}{T_{ref}} \right)^3 \cdot \exp \left[\frac{q \cdot E_g}{n \cdot k} \cdot \left(\frac{1}{T_{ref}} - \frac{1}{T} \right) \right], \quad (2.4)$$

де $I_{0,ref}$ – струм насичення за STC, А;

E_g – ширина забороненої зони (1,11 еВ для Si);

n – коефіцієнт ідеальності діода.

Зі зростанням температури I_0 збільшується, що спричиняє зниження напруги холостого ходу V_{oc} та, відповідно, зменшення максимальної вихідної потужності панелі. Для панелі Kyocera KC200GT температурний коефіцієнт потужності становить -0,485 %/°C, що означає зниження потужності приблизно на 1 Вт при підвищенні температури елемента на 1° C понад 25° C.

Реальна сонячна панель є масивом послідовно та паралельно з'єднаних елементів. Панель Kyocera KC200GT складається з 54 послідовно з'єднаних елементів, що визначає вихідну напругу модуля як суму напруг окремих елементів. При послідовному з'єднанні N_s елементів напруга модуля збільшується в N_s разів порівняно з напругою окремого елемента, тоді як струм залишається незмінним. При паралельному з'єднанні N_p рядків струм модуля збільшується в N_p разів. Відповідно рівняння вихідного струму масиву записується з підстановкою N_s та N_p у відповідні поля виразу. Такий підхід дозволяє легко масштабувати модель для дослідження масивів довільної конфігурації шляхом зміни лише двох параметрів – кількості послідовних та паралельних елементів.

Рівняння вихідного струму масиву з урахуванням конфігурації описується формулою (2.5):

$$I = N_p \cdot \left\{ I_{ph} - I_0 \cdot \left[\exp \left(\frac{V/N_s + I \cdot R_s/N_p}{n \cdot V_t} \right) - 1 \right] - \frac{V/N_s + I \cdot R_s/N_p}{R_{sh}} \right\}, \quad (2.5)$$

де N_s – кількість послідовно з'єднаних елементів;

N_p – кількість паралельно з'єднаних рядків.

Практична реалізація однодіодної моделі вимагає визначення п'яти параметрів: I_{ph} , I_0 , n , R_s та R_{sh} . Оскільки виробники панелей не надають ці параметри безпосередньо, їх необхідно ідентифікувати з паспортних даних, що включають V_{oc} , I_{sc} , V_{mpp} та I_{mpp} за стандартних умов тестування та температурні коефіцієнти. Процедура ідентифікації базується на системі рівнянь, що записуються для трьох характерних точок I-V кривої: точки короткого замикання ($V=0$, $I=I_{sc}$), точки холостого ходу ($V=V_{oc}$, $I=0$) та точки максимальної потужності ($V=V_{mpp}$, $I=I_{mpp}$). Рівняння для трьох характерних точок мають вигляд (2.6), (2.7) та (2.8):

$$I_{sc} = I_{ph} - I_0 \cdot [\exp(I_{sc} \cdot R_s / n \cdot V_t) - 1] - I_{sc} \cdot R_s / R_{sh}, \quad (2.6)$$

$$I_{ph} - I_0 \cdot [\exp(V_{oc} / n \cdot V_t) - 1] - V_{oc} / R_{sh} = 0, \quad (2.7)$$

$$I_{mpp} = I_{ph} - I_0 \cdot [\exp(V_{mpp} + I_{mpp} \cdot R_s / n \cdot V_t) - 1] - (V_{mpp} + I_{mpp} \cdot R_s) / R_{sh}, \quad (2.8)$$

Рівняння розв'язуються методом Ньютона-Рафсона, який реалізований у середовищі MATLAB як вбудована функція `fsolve`. Для панелі процедура ідентифікації має такі значення стандартних умов: коефіцієнт ідеальності $n \approx 1,3$, послідовний опір $R_s \approx 0,221$ Ом, шунтуючий опір $R_{sh} \approx 415,4$ Ом.

2.3 Модель споживача (IoT-навантаження)

Коректне моделювання системи живлення IoT-вузла неможливе без точного опису енергетичного профілю навантаження. Енергетичний профіль – це часова залежність споживаної потужності, що відображає реальний

характер роботи пристрою протягом одного або кількох робочих циклів. На відміну від традиційних споживачів зі стаціонарним споживанням, IoT-вузол на базі ESP32 характеризується різко імпульсним профілем з чергуванням фаз активної роботи та глибокого сну. Активна фаза включає пробудження процесора, ініціалізацію периферії, підключення до мережі Wi-Fi, зчитування даних з сенсорів та їх передачу на сервер. Споживання струму в цій фазі становить 80-260 мА при напрузі 3,3 В залежно від інтенсивності радіообміну, а тривалість – 2-5 секунд. У режимі глибокого сну активним залишається лише таймер реального часу, споживання знижується до 10 мкА, а тривалість фази сну між вимірюваннями складає 5-15 хвилин (рис. 2.1).

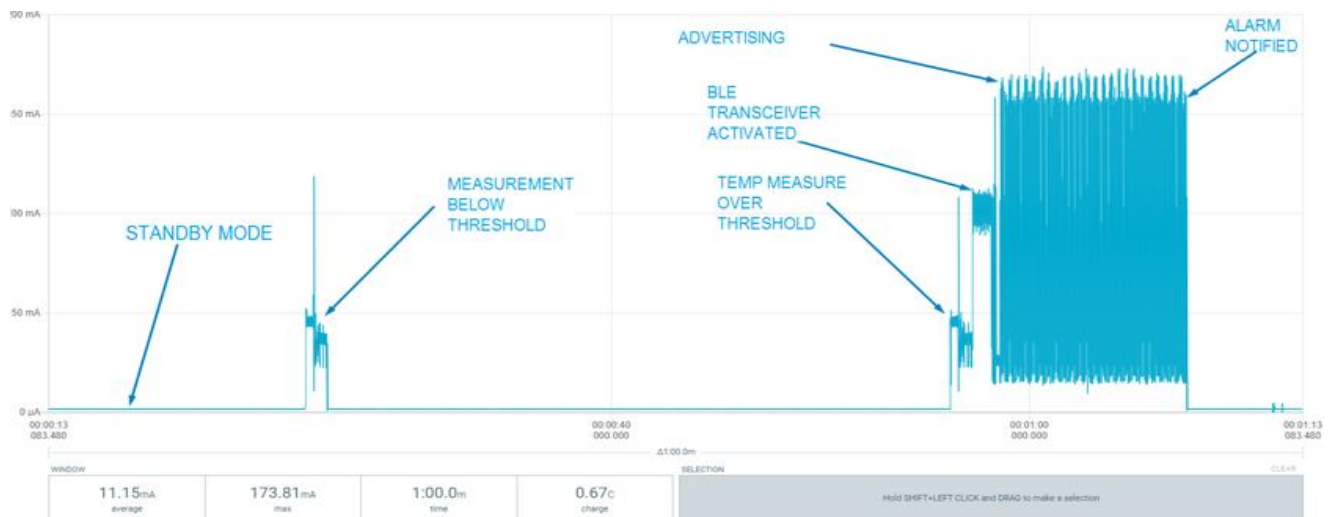


Рисунок 2.1 – Цикл роботи IoT-вузла ESP32 в режимах активності та глибокого сну [21]

Для потреб математичного моделювання енергетичний профіль IoT-вузла описується як дискретна часова функція споживаної потужності з двома рівнями. Параметри типового робочого циклу вузла прийняті такими: тривалість активної фази $t_{act} = 3$ с при середньому струмі споживання $I_{act} = 150$ мА, тривалість фази сну $t_{sleep} = 600$ с при струмі $I_{sleep} = 10$ мкА, напруга живлення $V_{cc} = 3,3$ В. Середній струм споживання за один повний цикл розраховується як середньозважена величина за формулою (2.9):

$$I_{avg} = (I_{act} \cdot t_{act} + I_{sleep} \cdot t_{sleep}) / (t_{act} + t_{sleep}), \quad (2.9)$$

де I_{act} – струм активної фази, А;

t_{act} – тривалість активної фази, с;

I_{sleep} – струм фази сну, А;

t_{sleep} – тривалість фази сну, с.

Підставляємо числові значення для ESP32:
 $I_{avg} = (0,150 \cdot 3 + 0,00001 \cdot 600) / (3 + 600) \approx 0,00075 \text{ А} = 0,75 \text{ мА}$, що відповідає середній потужності вузла 2,5 мВт. Добове енергоспоживання вузла визначається за формулою (2.10):

$$E_{daily} = I_{avg} \cdot V_{cc} \cdot 24 \cdot 3600, \quad (2.10)$$

де E_{daily} – добове енергоспоживання, Дж;

I_{avg} – середній струм за цикл, А;

V_{cc} – напруга живлення, В.

$E_{daily} = 0,00075 \cdot 3,3 \cdot 86400 \approx 213,8 \text{ Дж} \approx 0,06 \text{ Вт} \cdot \text{год}$. Добовий енергетичний профіль системи визначається не лише циклічним характером навантаження, але й нерівномірністю сонячної генерації протягом доби. Інтенсивність сонячного випромінювання змінюється від нуля вночі до максимальних значень у полудень і має характерний дзвоноподібний профіль. Відповідно система живлення повинна забезпечувати два режими: у денний час надлишок генерованої енергії спрямовується на заряджання акумулятора, у нічний час та в періоди низької інсоляції вузол живиться виключно від акумулятора. Зіставлення добового профілю генерації панелі Kyocera KC200GT з профілем споживання ESP32-вузла дозволяє визначити необхідну ємність акумулятора для забезпечення заданого рівня автономності системи, що є одним із ключових результатів моделювання.

2.4 Алгоритми керування та оптимізації

За результатами порівняльного аналізу MPPT-алгоритмів, проведеного в підрозділі 1.3, для реалізації в розроблюваній моделі обрано алгоритм збурення та спостереження (Perturb & Observe, P&O). Вибір обумовлений такими міркуваннями: алгоритм P&O не вимагає знання параметрів сонячної панелі та математичної моделі системи, що спрощує його практичну реалізацію; він добре документований у науковій літературі та має численні верифіковані реалізації в середовищі MATLAB/Simulink, що дозволяє порівнювати отримані результати з відомими; нарешті, обчислювальна простота алгоритму забезпечує його ефективну роботу в реальному часі в рамках інтерактивної моделі без суттєвого навантаження на обчислювальні ресурси [22].

Принцип роботи алгоритму P&O у контексті розроблюваної моделі полягає в такому. На кожному кроці симуляції контролер вимірює поточні значення вихідної напруги та струму панелі Kyocera KC200GT і обчислює поточну потужність формулою (2.11):

$$P(k) = V(k) \cdot I(k), \quad (2.11)$$

де $P(k)$ – потужність на k -му кроці, Вт;

$V(k)$ – напруга на k -му кроці, В;

$I(k)$ – струм на k -му кроці, А.

Далі до робочої напруги вноситься мале збурення фіксованого розміру ΔV , і після стабілізації системи знову вимірюється потужність. Якщо потужність після збурення зросла, наступне збурення виконується в тому ж напрямку; якщо зменшилась – у протилежному. Таким чином алгоритм ітераційно наближає робочу точку до MPP. Розмір кроку збурення ΔV є компромісом між швидкістю відстеження та точністю утримання в точці MPP: менший крок забезпечує менші

коливання в усталеному режимі, але уповільнює відстеження при різких змінах освітленості.

Інтерактивність розроблюваної моделі реалізується через механізм динамічного оновлення параметрів під час виконання симуляції. Зміна користувачем значення освітленості або температури через елементи керування інтерфейсу ініціює перерахунок параметрів однодіодної моделі – фотоструму I_{ph} та струму насичення I_0 – відповідно до температурних залежностей, описаних у підрозділі 2.2. Оновлені значення передаються до блоку PV Array у Simulink через змінні робочого простору MATLAB (Workspace variables), що забезпечує безперервність симуляції без її переривання та перезапуску. Алгоритм оновлення виконується з частотою, що відповідає кроку інтегрування симуляції, завдяки чому реакція моделі на зміну параметрів відображається на графіках практично миттєво.

Система керування в розроблюваній моделі реалізує два контури зворотного зв'язку. Перший контур – MPPT-регулятор, що вимірює вихідну потужність панелі і керує коефіцієнтом заповнення ШІМ-сигналу DC-DC boost-конвертера для утримання робочої точки поблизу MPP. Коефіцієнт заповнення ШІМ-сигналу визначається за формулою (2.12):

$$D = 1 - V_{out}/V_{in}, \quad (2.12)$$

де D – коефіцієнт заповнення ШІМ-сигналу;

V_{out} – вихідна напруга конвертера, В;

V_{in} – вхідна напруга конвертера, В.

Другий контур – регулятор заряду акумулятора, що контролює стан заряду (State of Charge, SoC) і обмежує зарядний струм при досягненні верхнього порогу напруги акумулятора. Обидва контури реалізуються засобами Simulink з використанням блоків порівняння, логіки перемикачів та блоку генератора ШІМ. Взаємодія між контурами забезпечує коректний розподіл енергії залежно від поточного балансу генерації та споживання.

РОЗДІЛ 3

РОЗРОБЛЕННЯ АПАРАТНОЇ ТА ПРОГРАМНОЇ ЧАСТИНИ ПРОЕКТУ

3.1 Програма моделювання та оптимізації ВДЕ

Програма `solar_1` код якої наведено на лістингу 3.1, реалізує моделювання системи відновлюваної генерації, що складається із сонячної панелі та вітрової турбіни, і виконує оптимізацію коефіцієнтів їх використання з метою мінімізації дефіциту покриття навантаження. Програма складається з трьох основних блоків: ініціалізація параметрів, моделювання покриття навантаження та оптимізація [23].

Лістинг 3.1 – Базова модель ВДЕ з оптимізацією

```
function solar_1
%% Моделювання та оптимізація відновлюваних джерел енергії
clc; clear; close all;
%% 1. Параметри системи
% Дані сонячної панелі
P_solar_max = 5; % кВт максимальна потужність сонячної панелі
irradiance = [0 200 400 600 800 1000]; % Вт/м^2
P_solar = P_solar_max * irradiance / 1000; % Потужність, що генерується
% Дані вітрової турбіни
P_wind_max = 3; % кВт
wind_speed = [0 3 6 9 12 15]; % м/с
P_wind = P_wind_max * (wind_speed/12).^3; % наближена залежність потужності
від швидкості вітру
P_wind(P_wind > P_wind_max) = P_wind_max; % обмеження максимальної
потужності
% Навантаження
load_demand = [1 2 3 4 4 5]; % кВт
%% 2. Моделювання покриття навантаження
% Сумарна потужність від ВДЕ
P_total = P_solar + P_wind;
% Дефіцит та надлишок
deficit = load_demand - P_total;
deficit(deficit < 0) = 0;
excess = P_total - load_demand;
excess(excess < 0) = 0;
%% 3. Візуалізація початкового стану
figure;
plot(irradiance, load_demand, 'k-o', 'LineWidth',1.5); hold on;
plot(irradiance, P_solar, 'r--o', 'LineWidth',1.5);
plot(irradiance, P_wind, 'b--o', 'LineWidth',1.5);
plot(irradiance, P_total, 'g-o', 'LineWidth',2);
xlabel('Інсоляція, Вт/м^2');
ylabel('Потужність, кВт');
```

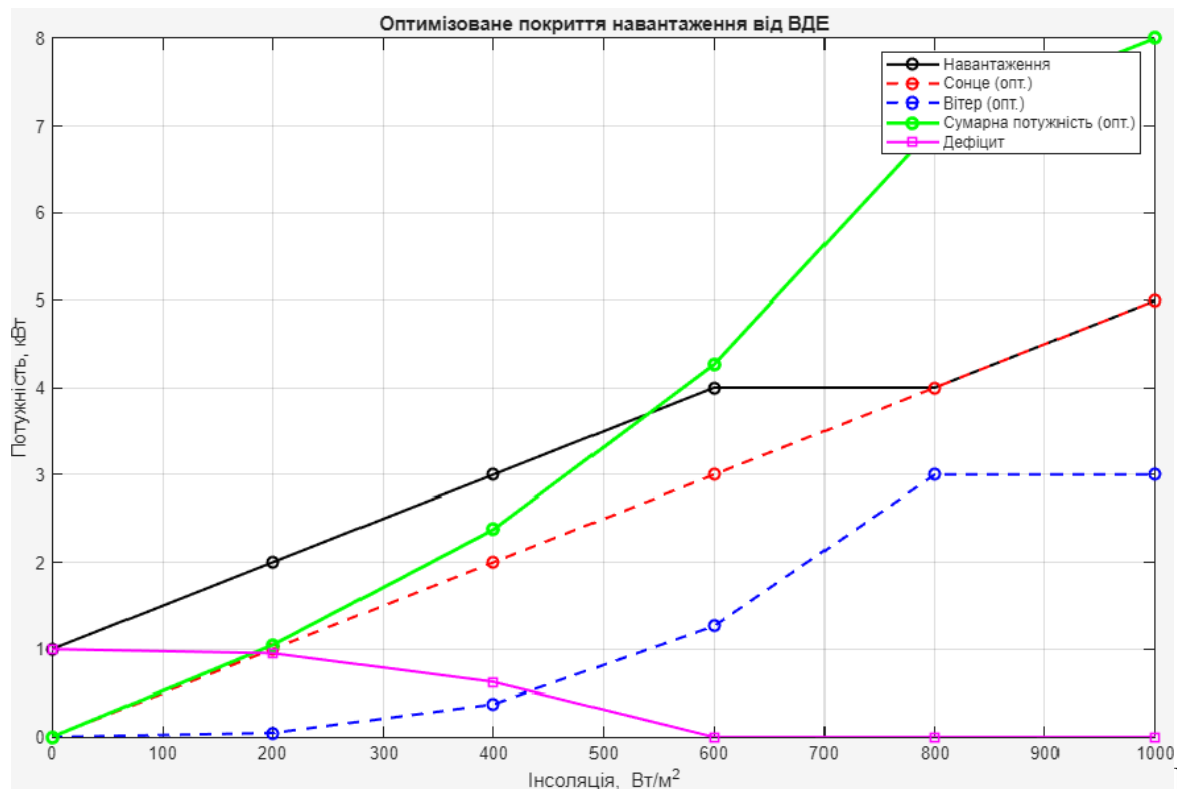
```

legend('Навантаження','Сонце','Вітер','Сумарна потужність');
grid on;
title('Моделювання відновлюваних джерел енергії');
%% 4. Оптимізація покриття навантаження
% Функція цілі: мінімізує сумарний дефіцит
fun = @(x) sum(max(load_demand - (x(1)*P_solar + x(2)*P_wind),0));
% Обмеження: коефіцієнти між 0 та 1
lb = [0 0];
ub = [1 1];
x0 = [0.5 0.5]; % початкове припущення
% Виконання оптимізації
options = optimoptions('fmincon','Display','iter');
[x_opt,fval] = fmincon(fun,x0,[],[],[],[],lb,ub,[],options);
disp(['Оптимальний коефіцієнт для сонця: ', num2str(x_opt(1))]);
disp(['Оптимальний коефіцієнт для вітру: ', num2str(x_opt(2))]);
%% 5. Візуалізація оптимальної генерації
P_total_opt = x_opt(1)*P_solar + x_opt(2)*P_wind;
deficit_opt = load_demand - P_total_opt;
deficit_opt(deficit_opt < 0) = 0;
figure;
plot(irradiance, load_demand,'k-o', 'LineWidth',1.5); hold on;
plot(irradiance, x_opt(1)*P_solar,'r--o', 'LineWidth',1.5);
plot(irradiance, x_opt(2)*P_wind,'b--o', 'LineWidth',1.5);
plot(irradiance, P_total_opt,'g-o', 'LineWidth',2);
plot(irradiance, deficit_opt,'m-s', 'LineWidth',1.5);
xlabel('Інсоляція, Вт/м^2');
ylabel('Потужність, кВт');
legend('Навантаження','Сонце (опт.)','Вітер (опт.)','Сумарна потужність (опт.)','Дефіцит');
grid on;
title('Оптимізоване покриття навантаження від ВДЕ');
end

```

кінець лістингу 3.1

Цільова функція обчислює суму додатних відхилень між навантаженням та зваженою сумою генерації. Змінними оптимізації є коефіцієнти $x(1)$ і $x(2)$ – масштабувальні множники для кожного джерела в діапазоні $[0, 1]$. Функція `fmincon` реалізує метод послідовного квадратичного програмування з обмеженнями на змінні. Початкове наближення $x0 = [0.5, 0.5]$ відповідає рівномірному розподілу між джерелами. Після оптимізації обчислюється залишковий дефіцит `deficit_opt` – потужність, яку не вдалося покрити навіть після оптимального розподілу. На рисунку 3.1 наведено результат виконання програми.



Рисунок

3.1 – Результат оптимізації покриття навантаження від ВДЕ (solar_1)

Оптимізований розподіл коефіцієнтів суттєво зменшує площу дефіциту потужності порівняно з початковим рівномірним розподілом $x = [0.5, 0.5]$. Крива сумарної потужності після оптимізації більш точно повторює профіль навантаження, а залишковий дефіцит зосереджений лише у точках з мінімальною інсоляцією.

3.2 Програма моделювання системи генерації з акумулятором

Програма solar_2 розширює модель solar_1 введенням акумуляторного накопичувача. Після оптимізації коефіцієнтів генерації запускається блок керування акумулятором, який дозволяє згладжувати дефіцит потужності за рахунок накопиченої енергії. Програма, код якої наведено на лістингу 3.2, включає блок ініціалізації акумулятора та покроковий цикл керування зарядом.

Лістинг 3.2 – Модель ВДЕ з акумулятором

```
function solar_2
```

```

%% Моделювання та оптимізація ВДЕ з акумулятором
clc; clear; close all;
%% 1. Параметри системи
% Сонячна панель
P_solar_max = 5; % кВт
irradiance = [0 200 400 600 800 1000]; % Вт/м^2
P_solar = P_solar_max * irradiance / 1000;
% Вітрова турбіна
P_wind_max = 3; % кВт
wind_speed = [0 3 6 9 12 15]; % м/с
P_wind = P_wind_max * (wind_speed/12).^3;
P_wind(P_wind > P_wind_max) = P_wind_max;
% Навантаження
load_demand = [1 2 3 4 4 5]; % кВт
% Акумулятор
battery_capacity = 5; % кВт·год
battery_charge = 0; % початковий заряд
eta_charge = 0.95; % ККД зарядки
eta_discharge = 0.95; % ККД розрядки
%% 2. Оптимізація покриття навантаження
fun = @(x) sum(max(load_demand - (x(1)*P_solar + x(2)*P_wind),0));
lb = [0 0];
ub = [1 1];
x0 = [0.5 0.5];
options = optimoptions('fmincon','Display','iter');
[x_opt,fval] = fmincon(fun,x0,[],[],[],[],lb,ub,[],options);
disp(['Оптимальний коефіцієнт для сонця: ', num2str(x_opt(1))]);
disp(['Оптимальний коефіцієнт для вітру: ', num2str(x_opt(2))]);
%% 3. Моделювання роботи з акумулятором
P_total_opt = x_opt(1)*P_solar + x_opt(2)*P_wind;
deficit_opt = zeros(size(load_demand));
battery_state = zeros(size(load_demand));
for t = 1:length(load_demand)
    surplus = P_total_opt(t) - load_demand(t);
    if surplus >= 0
        % Зарядка акумулятора
        battery_charge = battery_charge + surplus * eta_charge;
        if battery_charge > battery_capacity
            battery_charge = battery_capacity; % обмеження ємності
        end
    else
        % Використання акумулятора
        needed = -surplus / eta_discharge;
        if battery_charge >= needed
            battery_charge = battery_charge - needed;
            deficit_opt(t) = 0;
        else
            deficit_opt(t) = -surplus - battery_charge*eta_discharge;
            battery_charge = 0;
        end
    end
    battery_state(t) = battery_charge;
end

```

```

end
%% 4. Візуалізація результатів
figure;
plot(irradiance, load_demand,'k-o','LineWidth',1.5); hold on;
plot(irradiance, x_opt(1)*P_solar,'r--o','LineWidth',1.5);
plot(irradiance, x_opt(2)*P_wind,'b--o','LineWidth',1.5);
plot(irradiance, P_total_opt,'g-o','LineWidth',2);
plot(irradiance, deficit_opt,'m-s','LineWidth',1.5);
plot(irradiance, battery_state,'c-^','LineWidth',1.5);
xlabel('Інсоляція, Вт/м^2');
ylabel('Потужність, кВт');
legend('Навантаження','Сонце (опт.)','Вітер (опт.)','Сумарна
потужність','Дефіцит після акумулятора','Заряд батареї');
grid on;
title('Оптимізована система ВДЕ з акумулятором');
end

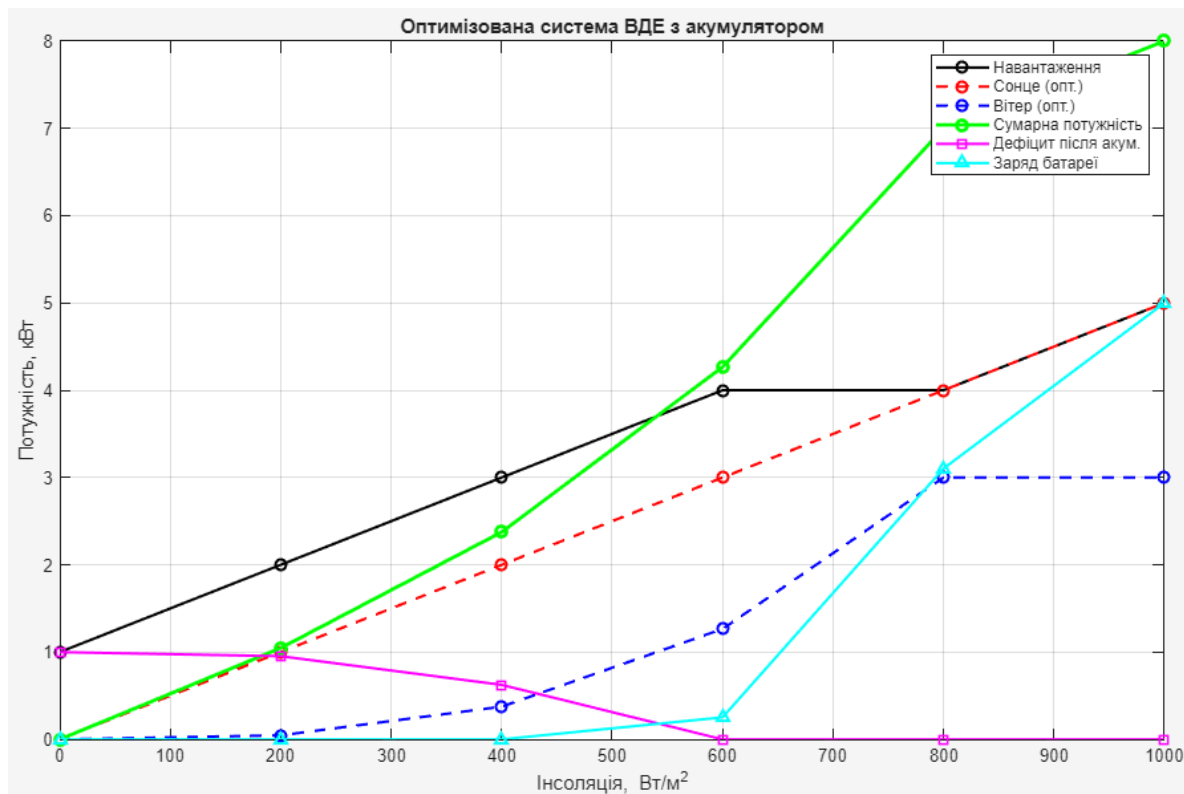
```

кінець лістингу 3.2

Ємність накопичувача $\text{battery_capacity} = 5 \text{ кВт} \cdot \text{год}$ та ККД зарядки й розрядки $\eta = 0,95$ є типовими характеристиками літій-іонних акумуляторів. Початковий заряд $\text{battery_charge} = 0$ відповідає початку розрахункового циклу з порожнім акумулятором. Вектори deficit_opt та battery_state ініціалізуються нулями для зберігання результатів покрокового моделювання.

На кожному кроці t обчислюється надлишок або дефіцит потужності як різниця між оптимізованою генерацією та навантаженням. При надлишку акумулятор заряджається з урахуванням ККД з обмеженням по ємності. При дефіциті спочатку використовується накопичений заряд; якщо заряду вистачає – дефіцит дорівнює нулю, інакше фіксується залишковий дефіцит після повного розряду. Поточний стан заряду зберігається у векторі battery_state для подальшої візуалізації.

На рисунку 3.2 представлено графік роботи системи з акумулятором. Крива заряду батареї демонструє накопичення енергії в години піку генерації та розряд у нічні й вечірні години. Порівняно з результатами solar_1 залишковий дефіцит суттєво зменшився – акумулятор компенсує нестачу потужності у більшості точок.



Рисунок

3.2 – Оптимізована система ВДЕ з акумулятором (solar_2)

3.3 Програма динамічного моделювання системи ВДЕ з акумулятором

Програма solar_3, код якої наведено на лістингу 3.3, переводить моделювання на 24-годинну часову вісь. Профілі генерації та навантаження задаються синусоїдальними функціями часу, що наближає модель до реального добового циклу роботи енергосистеми. Програма включає блок формування добових профілів та покроковий цикл динаміки акумулятора.

Лістинг 3.3 – Динамічна модель ВДЕ з акумулятором

```
function solar_3
% Динамічне моделювання та оптимізація ВДЕ з акумулятором
clc; clear; close all;
% 1. Параметри системи
hours = 0:1:23; % 24 години
% Сонячна панель (пік о 12 годині)
P_solar_max = 5; % кВт
irradiance = max(0, P_solar_max*sin(pi*(hours-6)/12)); % спрощена денна крива сонця
P_solar = irradiance;
```

```

% Вітрова турбіна (мінлива швидкість вітру)
P_wind_max = 3; % кВт
wind_speed = 5 + 3*sin(pi*hours/12); % м/с
P_wind = P_wind_max * (wind_speed/12).^3;
P_wind(P_wind > P_wind_max) = P_wind_max;
% Навантаження (мінлива протягом дня)
load_demand = 2 + sin(pi*(hours-6)/12)*2; % кВт
% Акумулятор
battery_capacity = 5; % кВт·год
battery_charge = 0; % початковий заряд
eta_charge = 0.95;
eta_discharge = 0.95;
%% 2. Оптимізація покриття навантаження
fun = @(x) sum(max(load_demand - (x(1)*P_solar + x(2)*P_wind),0));
lb = [0 0]; ub = [1 1]; x0 = [0.5 0.5];
options = optimoptions('fmincon','Display','iter');
[x_opt,fval] = fmincon(fun,x0,[],[],[],[],lb,ub,[],options);
disp(['Оптимальний коефіцієнт для сонця: ', num2str(x_opt(1))]);
disp(['Оптимальний коефіцієнт для вітру: ', num2str(x_opt(2))]);
%% 3. Динаміка системи з акумулятором
P_total_opt = x_opt(1)*P_solar + x_opt(2)*P_wind;
deficit_opt = zeros(size(load_demand));
battery_state = zeros(size(load_demand));
for t = 1:length(hours)
    surplus = P_total_opt(t) - load_demand(t);
    if surplus >= 0
        % Зарядка акумулятора
        battery_charge = battery_charge + surplus * eta_charge;
        if battery_charge > battery_capacity
            battery_charge = battery_capacity;
        end
        deficit_opt(t) = 0;
    else
        % Використання акумулятора
        needed = -surplus / eta_discharge;
        if battery_charge >= needed
            battery_charge = battery_charge - needed;
            deficit_opt(t) = 0;
        else
            deficit_opt(t) = -surplus - battery_charge*eta_discharge;
            battery_charge = 0;
        end
    end
    battery_state(t) = battery_charge;
end
%% 4. Візуалізація динаміки
figure;
plot(hours, load_demand,'k-o','LineWidth',1.5); hold on;
plot(hours, x_opt(1)*P_solar,'r--o','LineWidth',1.5);
plot(hours, x_opt(2)*P_wind,'b--o','LineWidth',1.5);
plot(hours, P_total_opt,'g-o','LineWidth',2);
plot(hours, deficit_opt,'m-s','LineWidth',1.5);

```

```
plot(hours, battery_state,'c-^','LineWidth',1.5);
xlabel('Години');
ylabel('Потужність, кВт');
legend('Навантаження','Сонце (опт.)','Вітер (опт.)','Сумарна
потужність','Дефіцит','Заряд батареї');
grid on;
title('Динамічна оптимізована система ВДЕ з акумулятором');
end
```

кінець лістингу 3.3

Функція $\text{irradiance} = \max(0, P_{\text{solar_max}} \cdot \sin(\pi \cdot (t-6)/12))$ моделює денний профіль сонячної інсоляції з нульовими значеннями у нічний період та піком о 12:00. Швидкість вітру задається функцією $\text{wind_speed} = 5 + 3 \cdot \sin(\pi \cdot t/12)$ з амплітудою 3 м/с навколо середнього значення 5 м/с. Навантаження описується синусоїдою з базовим рівнем 2 кВт та денним піком до 4 кВт.

Цикл перебирає всі 24 годинні точки та моделює добовий енергетичний баланс. Команда `min(..., battery_capacity)` замінює умовний оператор обмеження ємності, що робить код компактнішим. Вектор `battery_state` фіксує стан заряду в кожную годину доби, що дозволяє відстежити повний цикл заряду вдень та розряду вночі.

На рисунку 3.3 наведено добовий графік роботи системи. Чітко видно профіль сонячної генерації з піком о 12:00 та нульовими значеннями вночі. Крива заряду акумулятора зростає впродовж дня та знижується у вечірні й нічні години. Залишковий дефіцит фіксується переважно у ранкові години до початку активної сонячної генерації.

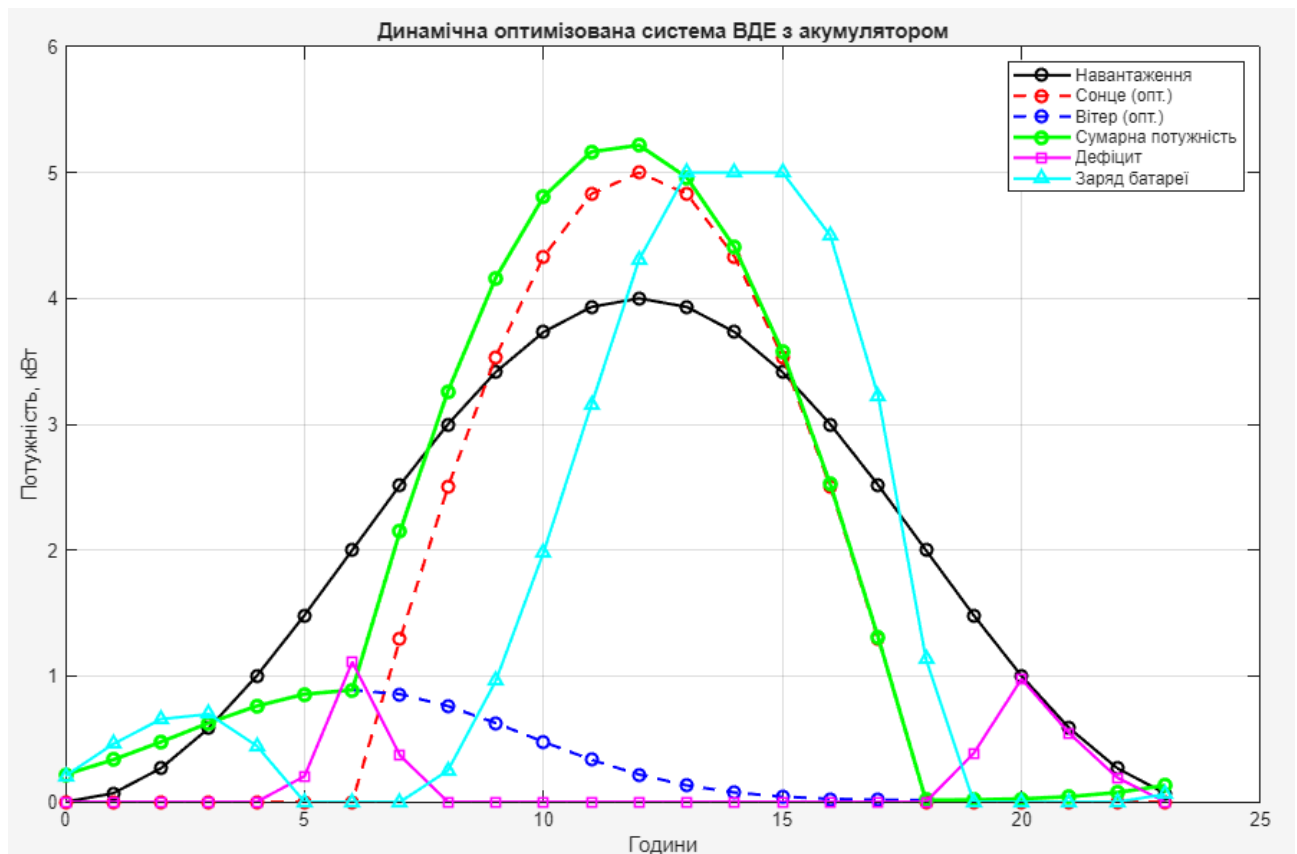


Рисунок 3.3 – Добова динаміка системи ВДЕ з акумулятором (solar_3)

3.4 Програма покадрової анімації роботи системи ВДЕ

Програма solar_4, код якої наведено на лістингу 3.4, відтворює роботу системи у вигляді покадрової анімації з часовим кроком одна година. На відміну від статичного графіка solar_3, результат відображається послідовно: кожен новий кадр додає поточну годину до вже побудованої частини графіка, що дозволяє спостерігати за розгортанням добового енергетичного балансу в динаміці.

Лістинг 3.4 – Покадрова анімація системи ВДЕ

```
function solar_4
%% Анімація роботи ВДЕ з акумулятором протягом дня
clc; clear; close all;
%% 1. Параметри системи
hours = 0:1:23; % 24 години
% Сонячна панель (пік о 12 годині)
P_solar_max = 5; % кВт
irradiance = max(0, P_solar_max*sin(pi*(hours-6)/12));
P_solar = irradiance;
% Вітрова турбіна (мінлива швидкість вітру)
```

```

P_wind_max = 3; % кВт
wind_speed = 5 + 3*sin(pi*hours/12); % м/с
P_wind = P_wind_max * (wind_speed/12).^3;
P_wind(P_wind > P_wind_max) = P_wind_max;
% Навантаження (мінлива протягом дня)
load_demand = 2 + sin(pi*(hours-6)/12)*2; % кВт
% Акумулятор
battery_capacity = 5; % кВт·год
battery_charge = 0;
eta_charge = 0.95;
eta_discharge = 0.95;
%% 2. Оптимізація
fun = @(x) sum(max(load_demand - (x(1)*P_solar + x(2)*P_wind),0));
lb = [0 0]; ub = [1 1]; x0 = [0.5 0.5];
options = optimoptions('fmincon','Display','off');
[x_opt,fval] = fmincon(fun,x0,[],[],[],[],lb,ub,[],options);
P_total_opt = x_opt(1)*P_solar + x_opt(2)*P_wind;
deficit_opt = zeros(size(load_demand));
battery_state = zeros(size(load_demand));
%% 3. Динаміка системи
for t = 1:length(hours)
surplus = P_total_opt(t) - load_demand(t);
if surplus >= 0
battery_charge = battery_charge + surplus * eta_charge;
if battery_charge > battery_capacity
battery_charge = battery_capacity;
end
deficit_opt(t) = 0;
else
needed = -surplus / eta_discharge;
if battery_charge >= needed
battery_charge = battery_charge - needed;
deficit_opt(t) = 0;
else
deficit_opt(t) = -surplus - battery_charge*eta_discharge;
battery_charge = 0;
end
end
battery_state(t) = battery_charge;
end
%% 4. Анімація
figure;
for t = 1:length(hours)
clf; % очищення фігури
hold on; grid on; box on;
bar(hours(1:t), load_demand(1:t),'FaceColor',[0.7 0.7 0.7],'BarWidth',0.4);
% навантаження
plot(hours(1:t), x_opt(1)*P_solar(1:t),'r-o','LineWidth',1.5); % сонце
plot(hours(1:t), x_opt(2)*P_wind(1:t),'b-o','LineWidth',1.5); % вітер
plot(hours(1:t), P_total_opt(1:t),'g-o','LineWidth',2); % сумарна
потужність
plot(hours(1:t), deficit_opt(1:t),'m-s','LineWidth',1.5); % дефіцит

```

```

plot(hours(1:t), battery_state(1:t),'c-^','LineWidth',1.5); % батарея
xlabel('Години'); ylabel('Потужність, кВт');
legend('Навантаження','Сонце (опт.)','Вітер (опт.)','Сумарна
потужність','Дефіцит','Заряд батареї','Location','northwest');
title(['Робота системи ВДЕ: година ', num2str(hours(t))]);
ylim([0 max([load_demand P_total_opt battery_capacity])+1]);
pause(0.5); % затримка між кадрами
end
end

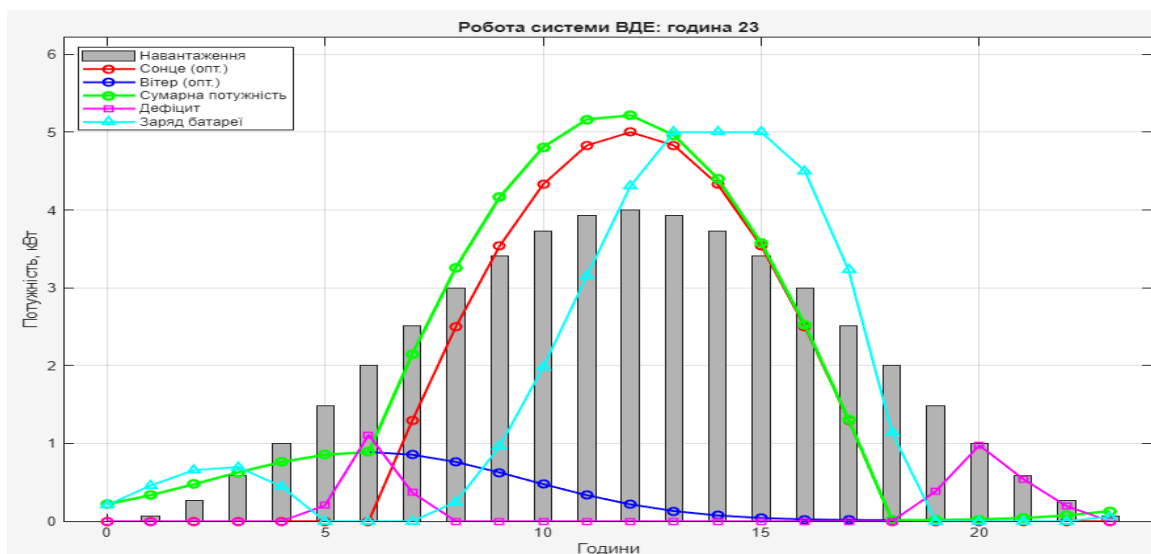
```

кінець лістингу 3.4

Параметр `Display = 'off'` прибирає виведення ітерацій `fmincon` у командне вікно, оскільки головний акцент програми – графічна анімація. Динаміка акумулятора обчислюється повністю до запуску анімаційного циклу, що забезпечує коректне відображення `battery_state` на кожному кадрі без додаткових обчислень всередині анімаційного циклу.

Команда `clf` на початку кожної ітерації повністю очищає графічне вікно. Зрізи масивів `hours(1:t)` та `P_solar(1:t)` забезпечують поступове нарощування графіка – відображаються лише дані від початку доби до поточного кроку `t`. Фіксований діапазон осі `ylim` запобігає стрибкам масштабу між кадрами.

На рисунку 3.4 наведено один із кадрів анімації – стан системи о 23:00.



Рисунок

3.4 – Кадр покадрової анімації роботи системи ВДЕ (solar_4)

3.5 Програма плавної анімації системи ВДЕ з інтерполяцією

Програма solar_5, код якої наведено на лістингу 3.4, є вдосконаленням solar_4 і забезпечує плавну анімацію з кроком 10 хвилин замість однієї години. Ключовим нововведенням є сплайнова інтерполяція всіх профілів з 24 годинних точок до 138 точок дрібної часової сітки. Програма містить блок інтерполяції та блок плавної анімації [24].

Лістинг 3.5 – Плавна анімація ВДЕ з інтерполяцією

```
function solar_1
%% Плавна анімація роботи ВДЕ з акумулятором (інтервал 10 хв)
clc; clear; close all;
%% 1. Параметри системи
hours = 0:1:23; % 24 години
time_fine = 0:1/6:23; % інтерполяція кожні 10 хвилин (1/6 години)
% Сонячна панель (спрощена денна крива)
P_solar_max = 5;
irradiance = max(0, P_solar_max*sin(pi*(hours-6)/12));
P_solar = interp1(hours, irradiance, time_fine, 'spline');
% Вітрова турбіна
P_wind_max = 3;
wind_speed = 5 + 3*sin(pi*hours/12);
P_wind = P_wind_max * (wind_speed/12).^3;
P_wind(P_wind > P_wind_max) = P_wind_max;
P_wind = interp1(hours, P_wind, time_fine, 'spline');
% Навантаження
load_demand = 2 + sin(pi*(hours-6)/12)*2;
load_demand = interp1(hours, load_demand, time_fine, 'spline');
% Акумулятор
battery_capacity = 5;
battery_charge = 0;
eta_charge = 0.95;
eta_discharge = 0.95;
%% 2. Оптимізація
fun = @(x) sum(max(load_demand(1:6:end) - (x(1)*irradiance +
x(2)*(P_wind(1:6:end))),0));
lb = [0 0]; ub = [1 1]; x0 = [0.5 0.5];
options = optimoptions('fmincon','Display','off');
[x_opt,fval] = fmincon(fun,x0,[],[],[],[],lb,ub,[],options);
P_total_opt = x_opt(1)*P_solar + x_opt(2)*P_wind;
deficit_opt = zeros(size(time_fine));
battery_state = zeros(size(time_fine));
%% 3. Динаміка з акумулятором
for t = 1:length(time_fine)
surplus = P_total_opt(t) - load_demand(t);
if surplus >= 0
battery_charge = battery_charge + surplus * eta_charge;
if battery_charge > battery_capacity
battery_charge = battery_capacity;
```

```

end
deficit_opt(t) = 0;
else
needed = -surplus / eta_discharge;
if battery_charge >= needed
battery_charge = battery_charge - needed;
deficit_opt(t) = 0;
else
deficit_opt(t) = -surplus - battery_charge*eta_discharge;
battery_charge = 0;
end
end
battery_state(t) = battery_charge;
end
%% 4. Плавна анімація
figure;
for t = 1:length(time_fine)
clf; hold on; grid on; box on;
% Навантаження (бар)
bar(time_fine(1:t), load_demand(1:t),'FaceColor',[0.7 0.7
0.7],'BarWidth',0.4);
% Сонце, вітер, сумарна потужність, дефіцит, батарея
plot(time_fine(1:t), x_opt(1)*P_solar(1:t),'r-o','LineWidth',1.5);
plot(time_fine(1:t), x_opt(2)*P_wind(1:t),'b-o','LineWidth',1.5);
plot(time_fine(1:t), P_total_opt(1:t),'g-o','LineWidth',2);
plot(time_fine(1:t), deficit_opt(1:t),'m-s','LineWidth',1.5);
plot(time_fine(1:t), battery_state(1:t),'c-^','LineWidth',1.5);
xlabel('Години'); ylabel('Потужність, кВт');
legend('Навантаження','Сонце (опт.)','Вітер (опт.)','Сумарна
потужність','Дефіцит','Заряд батареї','Location','northwest');
title(['Робота системи ВДЕ: година ', num2str(time_fine(t),'%.2f')]);
ylim([0 max([load_demand P_total_opt battery_capacity])+1]);
pause(0.1); % швидкість анімації
end
end

```

кінець лістингу 3.5

Функція `interp1` із методом `'spline'` будує кубічний сплайн через вузлові точки та обчислює значення у всіх точках дрібної сітки `time_fine`. На відміну від лінійної інтерполяції, сплайн забезпечує гладкість першого та другого порядку, що усуває кутові переломи між годинними значеннями. Результатом є три масиви по 138 елементів замість 24.

Оптимізація виконується на зрізі `load_demand(1:6:end)`, що відповідає кожній цілій годині, – це уникає надлишкових 138 обчислень. Маркери точок на кривих

відсутні (лінії без 'o'), що пришвидшує відрисовку 138 кадрів. Затримку зменшено до `pause(0.1)` – 10 кадрів на секунду.

На рисунку 3.5 наведено результат виконання програми. Порівняно з рисунком 3.4 криві виглядають значно плавніше завдяки сплайновій інтерполяції – відсутні кутові переломи між годинними значеннями. Густа часова сітка точніше відображає перехідні процеси, зокрема момент переходу від зарядки до розрядки акумулятора.

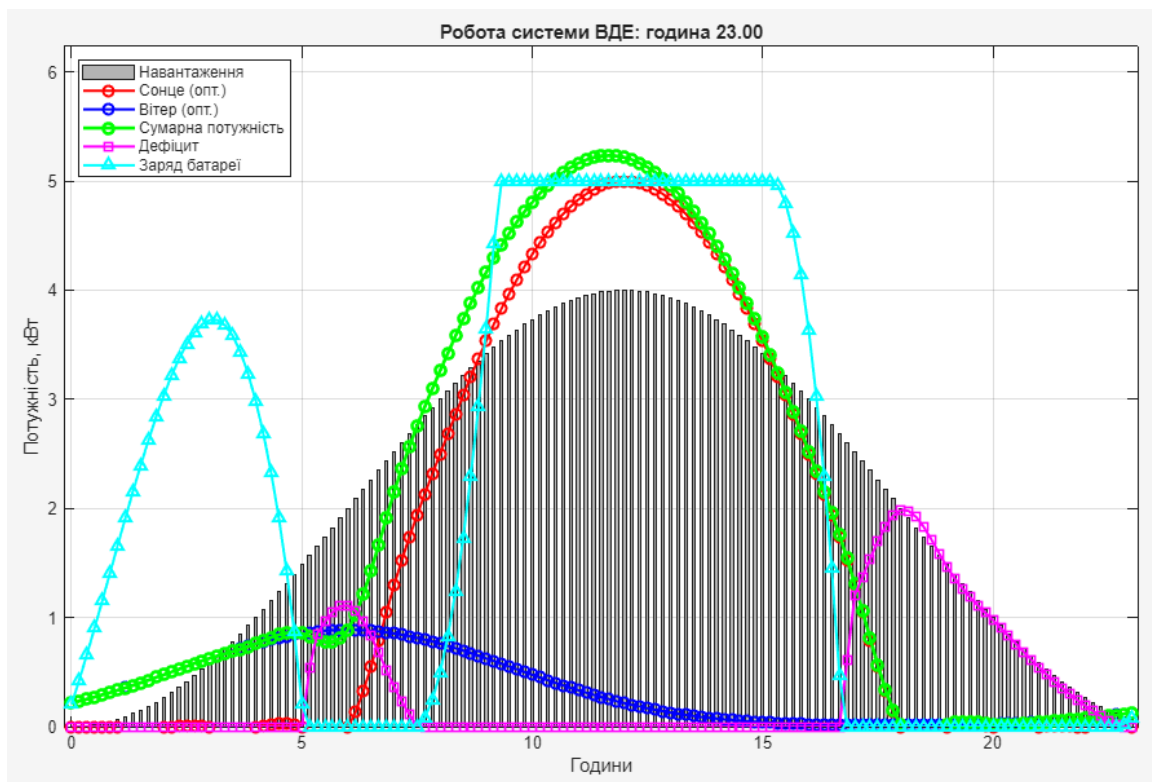


Рисунок 3.5 – Кадр плавної анімації системи ВДЕ з кроком 10 хвилин (solar_5)

3.6 Програма інтерактивної моделі системи ВДЕ з керуванням параметрами

Програма `solar_6`, код якої наведено на лістингу 3.6, реалізує повністю інтерактивний режим роботи: користувач у реальному часі керує трьома ключовими параметрами системи через графічні повзунки – потужністю сонячної панелі, потужністю вітрової турбіни та ємністю акумулятора. Графік автоматично перебудовується після кожної зміни положення повзунка.

Лістинг 3.6 – Інтерактивна модель ВДЕ з керуванням

```
function solar_6
%% Інтерактивна модель ВДЕ з акумулятором
clc; clear; close all;
%% 1. Параметри часу
hours = 0:1:23;
time_fine = 0:1/6:23; % кожні 10 хв
%% 2. Базові дані генерації та навантаження
P_solar_max = 5;
irradiance = max(0, P_solar_max*sin(pi*(hours-6)/12));
P_solar = interp1(hours, irradiance, time_fine, 'spline');
P_wind_max = 3;
wind_speed = 5 + 3*sin(pi*hours/12);
P_wind = P_wind_max * (wind_speed/12).^3;
P_wind(P_wind > P_wind_max) = P_wind_max;
P_wind = interp1(hours, P_wind, time_fine, 'spline');
load_demand = 2 + sin(pi*(hours-6)/12)*2;
load_demand = interp1(hours, load_demand, time_fine, 'spline');
%% 3. Створення фігури та графіка
fig = figure('Name','Інтерактивна ВДЕ система','Position',[100 100 1000
600]);
h_solar = plot(time_fine,P_solar,'r-o','LineWidth',1.5);
h_wind = plot(time_fine,P_wind,'b-o','LineWidth',1.5);
h_total = plot(time_fine,P_solar+P_wind,'g-o','LineWidth',2);
h_deficit = plot(time_fine,zeros(size(time_fine)),'m-s','LineWidth',1.5);
h_batt = plot(time_fine,zeros(size(time_fine)),'c-^','LineWidth',1.5);
xlabel('Години'); ylabel('Потужність, кВт');
legend('Навантаження','Сонце','Вітер','Сумарна потужність','Дефіцит','Заряд
батареї','Location','northwest');
%% 4. Повзунки для інтерактивності
% Потужність сонця
uicontrol('Style','text','Position',[820 500 120 20],'String','Потужність
сонця (кВт)');
slider_solar =
uicontrol('Style','slider','Min',0,'Max',10,'Value',5,'Position',[820 480
120 20]);
% Потужність вітру
uicontrol('Style','text','Position',[820 440 120 20],'String','Потужність
вітру (кВт)');
slider_wind =
uicontrol('Style','slider','Min',0,'Max',10,'Value',3,'Position',[820 420
120 20]);
% Ємність батареї
uicontrol('Style','text','Position',[820 380 120 20],'String','Ємність
батареї (кВт·год)');
slider_batt =
uicontrol('Style','slider','Min',0,'Max',10,'Value',5,'Position',[820 360
120 20]);
%% 5. Основний цикл для оновлення графіка
while ishandle(fig)
```

```

P_solar_max_val = slider_solar.Value;
P_wind_max_val = slider_wind.Value;
battery_capacity = slider_batt.Value;
P_solar = interp1(hours, max(0,P_solar_max_val*sin(pi*(hours-6)/12)),
time_fine, 'spline');
P_wind_raw = P_wind_max_val * (wind_speed/12).^3;
P_wind_raw(P_wind_raw > P_wind_max_val) = P_wind_max_val;
P_wind = interp1(hours, P_wind_raw, time_fine, 'spline')
P_total = P_solar + P_wind;
deficit = zeros(size(time_fine));
battery_charge = 0;
battery_state = zeros(size(time_fine));
eta_charge = 0.95; eta_discharge = 0.95;
for t = 1:length(time_fine)
surplus = P_total(t) - load_demand(t);
if surplus >=0
battery_charge = battery_charge + surplus*eta_charge;
if battery_charge > battery_capacity
battery_charge = battery_capacity;
end
deficit(t) = 0;
else
needed = -surplus/eta_discharge;
if battery_charge >= needed
battery_charge = battery_charge - needed;
deficit(t) = 0;
else
deficit(t) = -surplus - battery_charge*eta_discharge;
battery_charge = 0;
end
end
battery_state(t) = battery_charge;
end
% Оновлення графіка
h_solar.YData = P_solar;
h_wind.YData = P_wind;
h_total.YData = P_total;
h_deficit.YData = deficit;
h_batt.YData = battery_state;
pause(0.1);
end
end

```

кінець лістингу 3.6

Фігура створюється з фіксованим розміром 1100×600 пікселів, де права частина відведена під панель керування. Кожен повзунок формується парою `uicontrol`: текстова мітка 'text' та елемент 'slider' із заданими межами і кроком `SliderStep = [дрібний, великий]`. Параметр 'Value' задає початкове положення повзунка.

Цикл while зчитує значення всіх трьох повзунків та перераховує профілі генерації з оновленими параметрами. Оптимізація запускається повторно при кожному оновленні. Оновлення графіка відбувається через властивість YData без виклику plot заново – це зберігає налаштування осей і легенду та значно прискорює відгук інтерфейсу.

На рисунку 3.6 наведено вигляд інтерактивного інтерфейсу. При збільшенні потужності сонячної панелі крива сонячної генерації зростає пропорційно, а оптимізатор автоматично коригує коефіцієнт $x(1)$. Збільшення ємності акумулятора помітно зменшує залишковий дефіцит у нічні години.

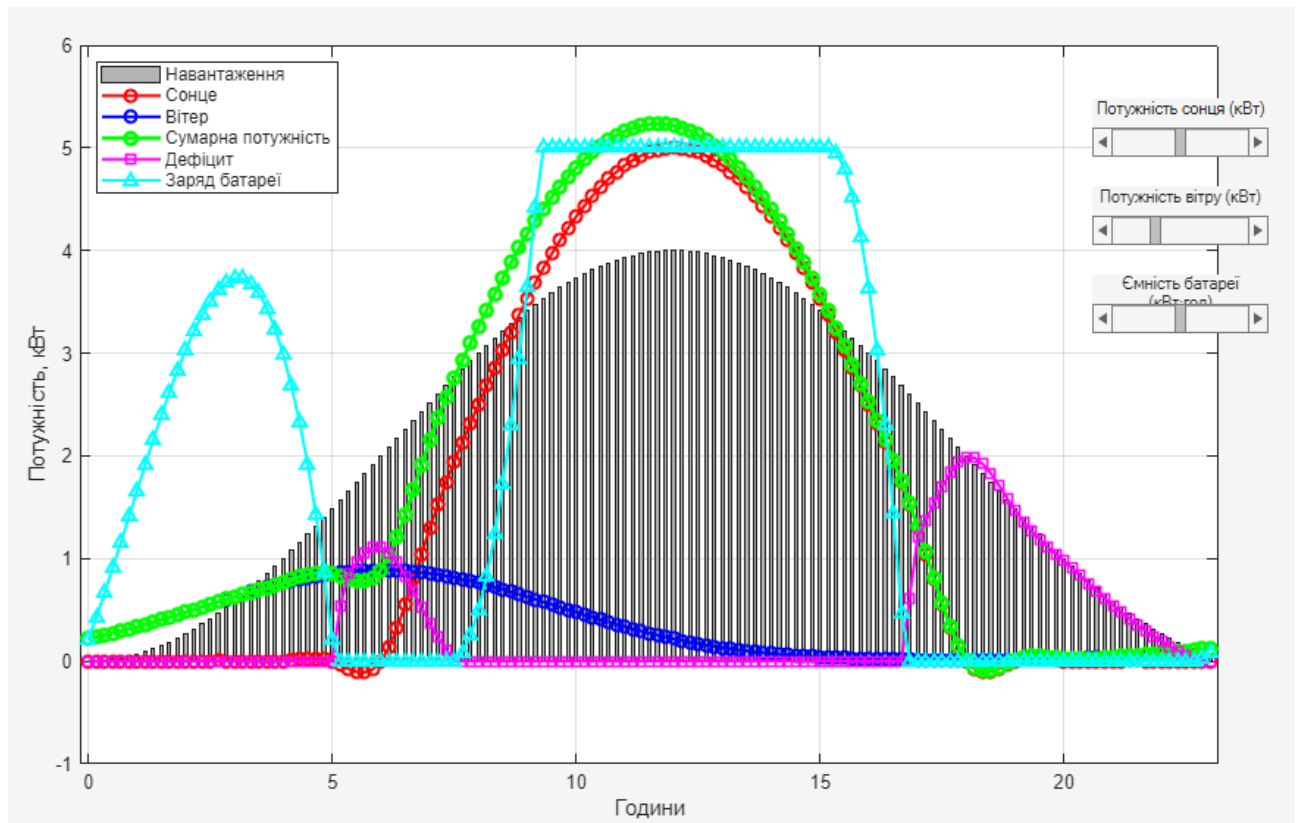


Рисунок 3.6 – Інтерактивний інтерфейс керування параметрами системи ВДЕ (solar_6)

3.7 Програма інтерактивного моделювання системи ВДЕ з вибором години

Програма solar_7 дозволяє користувачу обирати момент часу доби за допомогою повзунка та спостерігати за станом системи на обраний момент. На

відміну від solar_6, де керуються параметри системи, тут змінюється спостережуваний момент часу. Система автоматично симулює роботу акумулятора від початку доби до вибраного моменту та оновлює графік і вертикальний маркер поточного часу.

Лістинг 3.7 – Інтерактивне моделювання з вибором години

```
function solar_7
%% Інтерактивна ВДЕ система з вибором години
clc; clear; close all;
%% 1. Параметри часу
hours = 0:1:23;
time_fine = 0:1/6:23; % інтерполяція кожні 10 хв
%% 2. Базові дані генерації та навантаження
P_solar_max = 5;
irradiance = max(0, P_solar_max*sin(pi*(hours-6)/12));
P_solar = interp1(hours, irradiance, time_fine, 'spline');
P_wind_max = 3;
wind_speed = 5 + 3*sin(pi*hours/12);
P_wind = P_wind_max * (wind_speed/12).^3;
P_wind(P_wind > P_wind_max) = P_wind_max;
P_wind = interp1(hours, P_wind, time_fine, 'spline');
load_demand = 2 + sin(pi*(hours-6)/12)*2;
load_demand = interp1(hours, load_demand, time_fine, 'spline');
%% 3. Параметри батареї
battery_capacity = 5;
eta_charge = 0.95; eta_discharge = 0.95;
%% 4. Створення фігури та графіка
fig = figure('Name','Інтерактивна ВДЕ система з вибором години','Position',[100 100 1000 600]);
ax = axes('Parent',fig);
hold(ax,'on'); grid(ax,'on'); box(ax,'on');
h_load = bar(time_fine, load_demand,'FaceColor',[0.7 0.7 0.7],'BarWidth',0.4);
h_solar = plot(time_fine,P_solar,'r-o','LineWidth',1.5);
h_wind = plot(time_fine,P_wind,'b-o','LineWidth',1.5);
h_total = plot(time_fine,P_solar+P_wind,'g-o','LineWidth',2);
h_deficit = plot(time_fine,zeros(size(time_fine)),'m-s','LineWidth',1.5);
h_batt = plot(time_fine,zeros(size(time_fine)),'c-^','LineWidth',1.5);
xlabel('Години'); ylabel('Потужність, кВт');
legend('Навантаження','Сонце','Вітер','Сумарна потужність','Дефіцит','Заряд батареї','Location','northwest');
%% 5. Повзунок для вибору години
uicontrol('Style','text','Position',[820 500 120 20],'String','Година дня');
slider_time =
uicontrol('Style','slider','Min',0,'Max',23,'Value',0,'SliderStep',[1/23 1/23],'Position',[820 480 120 20]);
%% 6. Основний цикл для оновлення графіка
```

```

battery_state = 0; % початковий заряд
while ishandle(fig)
t_idx = round(slider_time.Value*6 + 1); % індекс у time_fine
if t_idx > length(time_fine)
t_idx = length(time_fine);
end
% Генерація
P_total = P_solar + P_wind;
deficit = zeros(size(time_fine));
battery_charge = battery_state;
% Симуляція роботи батареї від 0 до обраного часу
for t = 1:t_idx
surplus = P_total(t) - load_demand(t);
if surplus >= 0
battery_charge = battery_charge + surplus*eta_charge;
if battery_charge > battery_capacity
battery_charge = battery_capacity;
end
deficit(t) = 0;
else
needed = -surplus/eta_discharge;
if battery_charge >= needed
battery_charge = battery_charge - needed;
deficit(t) = 0;
else
deficit(t) = -surplus - battery_charge*eta_discharge;
battery_charge = 0;
end
end
battery_state = battery_charge;
end
% Оновлення графіка
h_total.YData = P_total;
h_deficit.YData = deficit;
h_batt.YData = battery_state*ones(size(time_fine));
% Маркер для поточної години
if exist('h_marker','var') && isvalid(h_marker)
delete(h_marker)
end
h_marker = line([time_fine(t_idx) time_fine(t_idx)], [0 max([load_demand
P_total battery_capacity])+1], 'Color','k','LineWidth',2,'LineStyle','--');
pause(0.1);
end
end

```

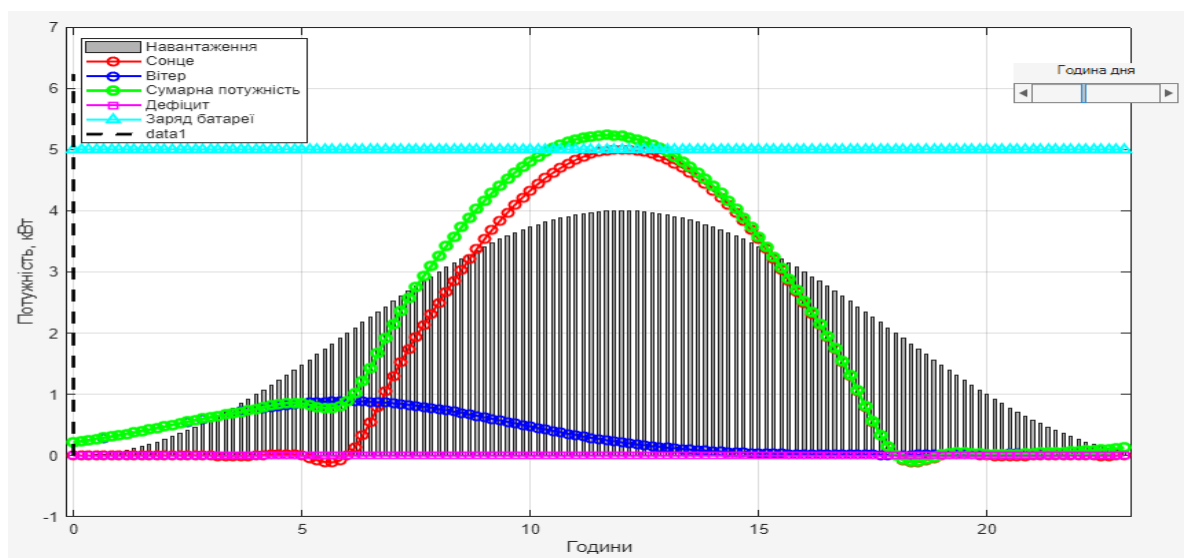
кінець лістингу 3.7

На відміну від solar_4, де графік будувався заново на кожному кадрі, тут усі шість кривих створюються одразу для повного добового діапазону. Криві дефіциту

та заряду батареї ініціалізуються нульовими векторами і оновлюватимуться через YData. Такий підхід виключає перебудову осей при взаємодії з повзунком.

Перетворення значення повзунка в індекс сітки здійснюється множенням на 6. Симуляція акумулятора запускається заново від нуля при кожній зміні повзунка, що гарантує коректний стан заряду незалежно від напрямку переміщення. Функція xline будує вертикальну штрихову лінію на поточній годині; попередній маркер видаляється перед побудовою нового.

На рисунку 3.7 наведено вигляд інтерфейсу. Ліворуч від маркера відображаються ненульові значення дефіциту та заряду батареї, що відповідають симульованій роботі системи від початку доби. Праворуч – нульові значення, оскільки система ще не досягла цих моментів.



Рисунок

3.7 – Інтерактивний інтерфейс вибору години доби (solar_7)

3.8 Програма плавної анімації ізольованої системи з акумулятором

Програма solar_only моделює спрощену систему, що складається виключно із сонячної панелі та акумулятора без вітрової турбіни. Мета програми – продемонструвати внесок вітрової складової шляхом порівняння результатів із попередніми програмами. Оптимізація зводиться до підбору одного скалярного коефіцієнта, що суттєво спрощує задачу.

Лістинг 3.8 – Анімація ізольованої сонячної системи

```
function solar_only
%% Плавна анімація роботи сонячної системи з акумулятором (інтервал 10 хв)
clc; clear; close all;
%% 1. Параметри системи
hours = 0:1:23; % 24 години
time_fine = 0:1/6:23; % інтерполяція кожні 10 хвилин (1/6 години)
% Сонячна панель (спрощена денна крива)
P_solar_max = 5;
irradiance = max(0, P_solar_max * sin(pi*(hours-6)/12));
P_solar = interp1(hours, irradiance, time_fine, 'spline');
% Навантаження
load_demand = 2 + sin(pi*(hours-6)/12)*2;
load_demand = interp1(hours, load_demand, time_fine, 'spline');
% Акумулятор
battery_capacity = 5;
battery_charge = 0;
eta_charge = 0.95;
eta_discharge = 0.95;
%% 2. Оптимізація лише сонячної частки
fun = @(x) sum(max(load_demand(1:6:end) - (x(1)*irradiance), 0));
lb = 0; ub = 1; x0 = 0.5;
options = optimoptions('fmincon','Display','off');
[x_opt, fval] = fmincon(fun, x0, [], [], [], [], lb, ub, [], options);
P_total_opt = x_opt * P_solar;
deficit_opt = zeros(size(time_fine));
battery_state = zeros(size(time_fine));
%% 3. Динаміка з акумулятором
for t = 1:length(time_fine)
    surplus = P_total_opt(t) - load_demand(t);
    if surplus >= 0
        battery_charge = battery_charge + surplus * eta_charge;
        if battery_charge > battery_capacity
            battery_charge = battery_capacity;
        end
        deficit_opt(t) = 0;
    else
        needed = -surplus / eta_discharge;
        if battery_charge >= needed
            battery_charge = battery_charge - needed;
            deficit_opt(t) = 0;
        else
            deficit_opt(t) = -surplus - battery_charge * eta_discharge;
            battery_charge = 0;
        end
    end
    battery_state(t) = battery_charge;
end
%% 4. Плавна анімація
figure;
for t = 1:length(time_fine)
    clf; hold on; grid on; box on;
```

```

% Навантаження (бар)
bar(time_fine(1:t), load_demand(1:t), 'FaceColor', [0.7 0.7 0.7],
'BarWidth', 0.4);
% Сонце, сумарна потужність, дефіцит, батарея
plot(time_fine(1:t), x_opt * P_solar(1:t), 'r-o', 'LineWidth', 1.5);
plot(time_fine(1:t), P_total_opt(1:t), 'g-o', 'LineWidth', 2);
plot(time_fine(1:t), deficit_opt(1:t), 'm-s', 'LineWidth', 1.5);
plot(time_fine(1:t), battery_state(1:t), 'c-^', 'LineWidth', 1.5);
xlabel('Години'); ylabel('Потужність, кВт');
legend('Навантаження', 'Сонце (опт.)', 'Сумарна потужність', 'Дефіцит',
'Заряд батареї', 'Location', 'northwest');
title(['Робота сонячної системи: година ', num2str(time_fine(t), '%.2f')]);
ylim([0 max([load_demand P_total_opt battery_capacity]) + 1]);
pause(0.1); % швидкість анімації
end
end

end

```

кінець лістингу 3.8

Цільова функція містить лише одну змінну x – скалярний коефіцієнт для сонячної панелі, на відміну від вектора $[x(1), x(2)]$ у попередніх програмах. Межі lb та ub задаються скалярами, а не векторами. Відсутність вітрової турбіни призводить до вищого залишкового дефіциту $fval$ порівняно з комбінованими системами.

Графік містить п'ять кривих замість шести – відсутня крива вітрової турбіни, що відображається і у скороченій легенді. Завдяки цьому зменшується візуальне навантаження та чіткіше виділяється поведінка сонячної складової. Анімація виконується по 138 кадрах аналогічно `solar_5`, що дозволяє коректно порівняти результати двох програм.

На рисунку 3.8 наведено кадр анімації о 14:33. Порівняно з рисунком 3.5 чітко видно збільшення дефіциту потужності – особливо у вечірні та нічні години, де відсутня вітрова генерація. Це наочно підтверджує необхідність комбінування сонячної та вітрової генерації у гібридній системі.

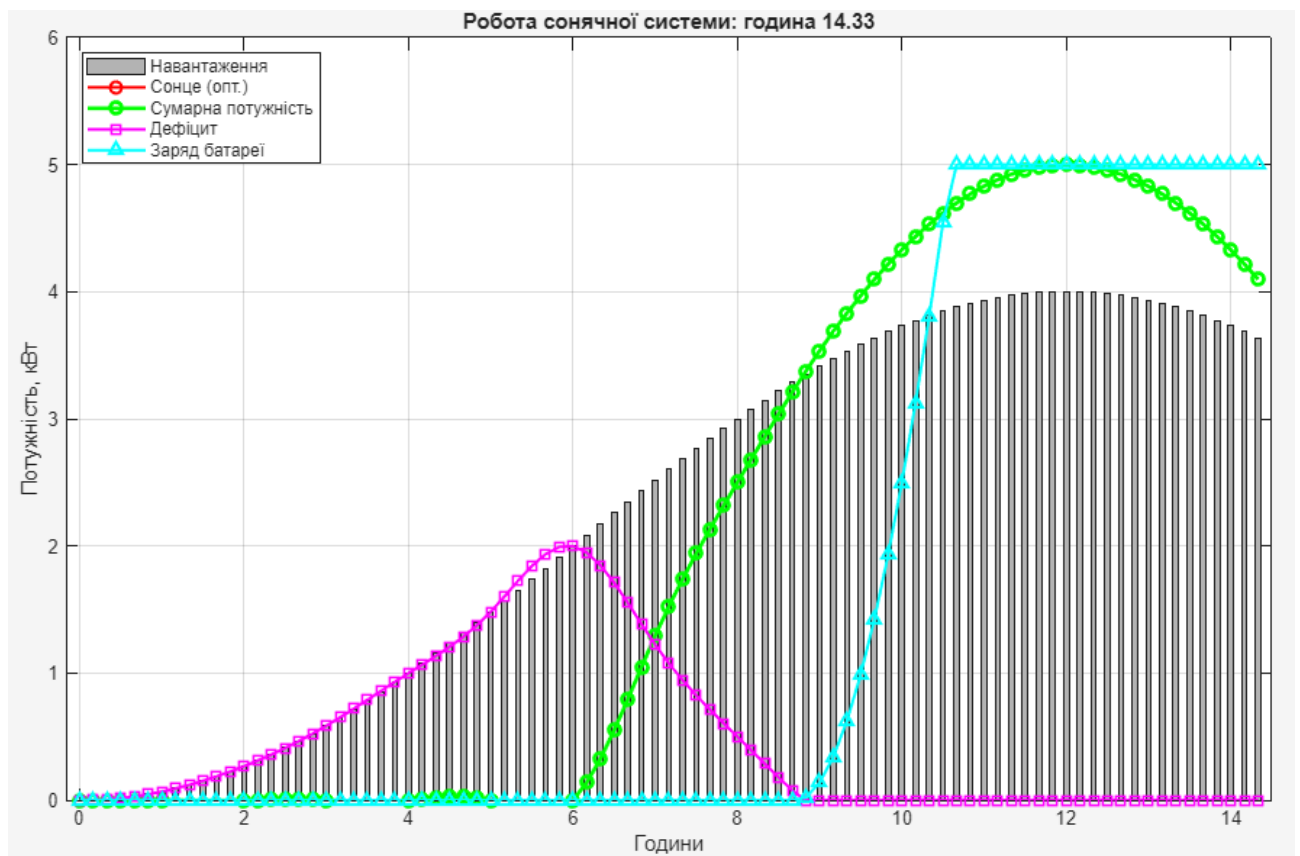


Рисунок 3.8 – Плавна анімація ізольованої сонячної системи з акумулятором (solar_only)

ВИСНОВКИ

Досліджено принципи роботи фотоелектричних перетворювачів та проаналізовано однодіодну і дводіодну моделі сонячних батарей. Розглянуто фізичні основи фотоелектричного ефекту, температурні залежності параметрів фотомодуля та алгоритми відстеження точки максимальної потужності.

Реалізовано інтерактивну імітаційну модель сонячної батареї у середовищі MATLAB з можливістю динамічної зміни рівня освітленості та параметрів навантаження безпосередньо під час симуляції за допомогою графічних повзунків. Модель забезпечує відображення робочих характеристик системи в режимі реального часу та дозволяє досліджувати вплив змінних умов освітленості на вихідну потужність фотомодуля.

Візуалізовано результати моделювання у вигляді графіків вихідної напруги, струму та потужності за різних умов освітленості. Як бачимо з отриманих графіків, зміна рівня інсоляції суттєво впливає на робочу точку системи. Реалізовано покадрову та плавну анімацію з кроком 10 хвилин на основі сплайнової інтерполяції, що наочно відображає динаміку добового енергетичного балансу системи.

Запропоновано серію з восьми MATLAB-програм для аналізу роботи системи відновлюваної генерації з акумулятором. Програми реалізують оптимізацію коефіцієнтів генерації функцією `fmincon`, покроковий цикл керування зарядом акумулятора з урахуванням ККД, а також інтерактивні інтерфейси з повзунками для керування параметрами системи в режимі реального часу. Комбінування сонячної та вітрової генерації суттєво підвищує ефективність покриття добового навантаження. Розроблену систему можна використовувати для навчальних цілей, інженерного аналізу фотоелектричних систем та попереднього проектування систем автономного живлення на базі відновлюваних джерел енергії.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Olajuyin E. A., Akintayo P. I., Ayodele A. T. Modelling of an IoT-Enabled Solar Photovoltaic Plant for Optimal Output Power Applications and Monitoring. *Journal of Innovation Science and Technology*. 2026. Vol. 5, No. 1. P. 2-7. URL: <https://jist.fuoye.edu.ng/index.php/jist/article/view/148> (date of access: 19.04.2026).
2. Working principle of solar cell with p-n junction structure. ResearchGate: URL: https://www.researchgate.net/figure/a-working-principle-of-solar-cell-with-p-n-junction-structure-and-b-loss-mechanism-in_fig1_355906446 (date of access: 24.03.2026).
3. Types of Solar Panels: Monocrystalline, Polycrystalline, and Thin-film. SolarReviews. URL: <https://www.solarreviews.com/blog/pros-and-cons-of-monocrystalline-vs-polycrystalline-solar-panels> (date of access: 17.03.2026).
4. Solar Cell I-V Curve Schematic. Wikimedia Commons: URL: https://commons.wikimedia.org/wiki/File:Solar_Cell_I-V_Curve_Schematic.svg (date of access: 18.03.2026).
5. I-V characteristic curves of solar cells under different conditions. ResearchGate: URL: https://www.researchgate.net/figure/The-I-V-characteristic-curve-of-solar-cells-under-different-sunlight-intensity_fig10_258388008 (date of access: 26.04.2026).
6. Solar Cell I-V Characteristic and the Solar Cell I-V Curve. Alternative Energy Tutorials: URL: <https://www.alternative-energy-tutorials.com/photovoltaics/solar-cell-i-v-characteristic.html> (date of access: 11.03.2026).
7. Tamrakar V., Gupta S. C., Sawle Y. Study of Characteristics of Single and Double Diode Electrical Equivalent Circuit Models of Solar PV Module. 2015 International Conference on Energy Systems and Applications (ICESA) (Pune, India, 30 Oct.-1 Nov. 2015). 2015. P. 312-317. DOI: 10.1109/ICESA.2015.7503362.
8. Rawa M., Calasan M., Abusorrah A., Alhussainy A. A., Al-Turki Y., Ali Z. M., Sindi H., Mekhilef S., Abdel Aleem S. H. E., Bassi H. Single Diode Solar Cells – Improved Model and Exact Current–Voltage Analytical Solution Based on Lambert’s W Function. *Sensors*. 2022. Vol. 22, No. 11. Article 4173. DOI: 10.3390/s22114173.

9. Perturb and Observe MPPT algorithm. ResearchGate: URL: https://www.researchgate.net/figure/Perturb-and-Observe-P-O-MPPT-algorithm_fig7_325419198 (date of access: 16.03.2026).

10. Flowchart of the fixed-step-size Incremental Conductance MPPT method. ResearchGate: URL: https://www.researchgate.net/figure/A-flowchart-of-the-fixed-step-size-incremental-conductance-MPPT-method_fig4_349150983 (date of access: 22.04.2026).

11. Attaway S. MATLAB: A Practical Introduction to Programming and Problem Solving. 5th ed. Oxford: Butterworth-Heinemann, 2022. 674 p.

12. Hahn B., Valentine D. Essential MATLAB for Engineers and Scientists. 7th ed. Oxford: Academic Press, 2022. 440 p.

13. Moore H. MATLAB for Engineers. 5th ed. Hoboken: Pearson, 2022. 624 p.

14. How to Build an ESP32 Low Power Sensor Node for Long-Life Battery IoT Projects. SunFounder: URL: <https://www.sunfounder.com/blogs/news/how-to-build-an-esp32-low-power-sensor-node-for-long-life-battery-iot-projects> (date of access: 27.03.2026).

15. Espressif Systems. ESP32 Series Datasheet. Version 4.4. Shanghai: Espressif Systems, 2023. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (date of access: 16.04.2026).

16. Block diagram of Solar Energy Harvesting System using MPPT Control. ResearchGate: URL: https://www.researchgate.net/figure/Block-diagram-of-Solar-Energy-Harvesting-System-using-MPPT-Control_fig1_337487050 (date of access: 26.04.2026).

17. Linden D., Reddy T. B. Handbook of Batteries. 4th ed. New York: McGraw-Hill, 2022. 1200 p.

18. Simscape Electrical: Documentation. MathWorks: URL: <https://www.mathworks.com/help/sps/index.html> (date of access: 25.03.2026).

19. Kyocera Solar. KC200GT High Efficiency Multicrystal Photovoltaic Module: Datasheet. Kyocera Corporation, 2022. URL: <https://www.kyocerasolar.com/assets/001/5191.pdf> (date of access: 15.03.2026).

20. Solar Cell block – Simscape Electrical. MathWorks: URL: <https://www.mathworks.com/help/sps/ref/solarcell.html> (date of access: 11.03.2026).
21. ESP32 node: Low-power mode (Deep Sleep) and sensor reading. ResearchGate: URL: https://www.researchgate.net/figure/ESP32-node-Low-power-mode-Deep-Sleep-and-sensor-reading_fig9_364433823 (date of access: 13.04.2026).
22. Hussein Selman N. Comparison Between Perturb & Observe, Incremental Conductance and Fuzzy Logic MPPT Techniques at Different Weather Conditions. International Journal of Innovative Research in Science, Engineering and Technology. 2016. Vol. 5, Issue 7. P. 12556-12569. URL: <https://www.researchgate.net/publication/306010127> (date of access: 26.04.2026).
23. MATLAB fmincon Documentation. MathWorks: URL: <https://www.mathworks.com/help/optim/ug/fmincon.html> (date of access: 06.04.2026).
24. Interp1 – 1-D data interpolation. MathWorks: URL: <https://www.mathworks.com/help/matlab/ref/interp1.html> (date of access: 12.03.2026).