

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет (ННІ) _____ Конструювання та дизайну _____

ПОГОДЖЕНО

Декан факультету
Конструювання та дизайну
(назва факультету)

_____ **Іван РОГОВСЬКИЙ**
(підпис) (ПІБ)

“ ___ ” _____ 2025 р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

Завідувач кафедри
Конструювання машин і обладнання
(назва кафедри)

_____ **Вячеслав ЛОВЕЙКІН**
(підпис) (ПІБ)

“ ___ ” _____ 2025 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

**на тему СИНТЕЗ ОПТИМАЛЬНОГО КЕРУВАННЯ РУХОМ
ОБЕРНЕНОГО МАЯТНИКА З РУХОМОЮ ТОЧКОЮ ПІДВІСУ**

Спеціальність _____ 133 – Галузеве машинобудування _____

Освітня програма Робототехнічні системи і комплекси сільськогосподарського виробництва
(код і назва)

Орієнтація освітньої програми _____ Професійна _____

Гарант освітньої програми

д.т.н., професор
(науковий ступінь та вчене звання)

_____ (підпис)

Юрій РОМАСЕВИЧ
(ПІ)

Керівник магістерської кваліфікаційної роботи

д.т.н., професор
(науковий ступінь та вчене звання)

_____ (підпис)

Юрій РОМАСЕВИЧ
(ПІ)

Виконав

_____ (підпис)

Максим НАЗАРОВ
(ПІ студента)

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ) Конструювання та дизайну

ЗАТВЕРДЖУЮ
Завідувач кафедри
конструювання машин і обладнання

Д.т.н. професор _____
(науковий ступінь, вчене звання) (підпис)

Вячеслав ЛОВЕЙКІН
(ПІБ)

“ _____ ” _____ 2025 р.

ЗАВДАННЯ
ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ

Назарову Максиму Вікторовичу

(прізвище, ім'я, по батькові)

Спеціальність 133 – Галузеве машинобудування

Освітня програма Робототехнічні системи і комплекси сільськогосподарського виробництва

Орієнтація освітньої програми Професійна

(код і назва)

Тема магістерської кваліфікаційної роботи - **Синтез оптимального керування рухом оберненого маятника з рухомою точкою підвісу** затверджена наказом ректора НУБіП України від “16” грудня 2024р. №2269 «С»

Термін подання завершеної роботи (проекту) на кафедру, _____
(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи

1.Лабораторна установка оберненого маятника з рухомою точкою підвісу.

Перелік питань, що підлягають дослідженню:

- 1.Провести кількісний та змістовий аналіз науково-технічних публікацій та розглянути прикладні аспекти використання обернених маятників.
- 2.Виконати постановку та розв'язок лінійної задачі, натренувати ШНМ
- 3.Виконати постановку та розв'язок нелінійної задачі.
- 4.Розглянути економічну доцільність розробки.
5. Розглянути питання охорони праці при роботі з оберненим маятником із рухомою точкою підвісу.

Перелік графічних документів (за потреби)

Дата видачі завдання “19” грудня 2025 р.

Керівник магістерської кваліфікаційної роботи _____ Юрій РОМАСЕВИЧ
(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____ Максим НАЗАРОВ
(підпис) (прізвище та ініціали студента)

ЗМІСТ

РЕФЕРАТ	4
ВСТУП	5
РОЗДІЛ 1. ОБГРУНТУВАННЯ АКТУАЛЬНОСТІ УДОСКОНАЛЕННЯ	6
1.1 Кількісний аналіз публікацій.....	6
1.2 Змістовний аналіз наукових публікацій.....	11
1.3 Прикладні аспекти використання маятникових систем.....	14
РОЗДІЛ 2. ПОСТАНОВКА І РОЗВ'ЯЗАННЯ ЗАДАЧІ СИНТЕЗУ ОПТИМАЛЬНОГО КЕРУВАННЯ ОБЕРНЕНИМ МАЯТНИКОМ У ЛІНІЙНІЙ ПОСТАНОВЦІ	19
2.1 Постановка задачі.....	19
2.2 Розробка моделі нейронної мережі.....	22
2.3 Тренування нейронної мережі.....	27
2.4 Аналіз результатів тренування.....	32
РОЗДІЛ 3. ПОСТАНОВКА І РОЗВ'ЯЗАННЯ ЗАДАЧІ СИНТЕЗУ ОПТИМАЛЬНОГО КЕРУВАННЯ ОБЕРНЕНИМ МАЯТНИКОМ У НЕЛІНІЙНІЙ ПОСТАНОВЦІ	38
3.1 Постановка задачі.....	38
3.2 Розробка оператора керування ШНМ.....	40
3.3 Ініціалізація тренування ШНМ.....	43
3.4 Результат тренування ШНМ.....	45
3.5 Підсумок тренувань.....	49
РОЗДІЛ 4. ОХОРОНА ПРАЦІ	50
4.1 Загальні положення.....	50
4.2 Вимоги безпеки перед початком роботи.....	51
4.3 Вимоги безпеки під час виконання роботи.....	52
4.4 Вимоги безпеки в аварійних ситуаціях.....	52
РОЗДІЛ 5. ЕКОНОМІЧНА ДОЦІЛЬНІСТЬ РОЗРОБКИ ТА СФЕРИ ПРАКТИЧНОГО ЗАСТОСУВАННЯ	53
5.1 Економічна ефективність розробки.....	53
5.2 Сфери використання розробленої системи.....	53
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56

РЕФЕРАТ

Магістерська кваліфікаційна робота містить вступ, п'ять розділів, висновки і список використаних джерел. Роботу викладено на друкованих аркушах та оформлено згідно з вимогами.

У першому розділі проведено аналіз актуальності дослідження, виконано кількісний та змістовний огляд наукових публікацій, а також розглянуто практичні приклади застосування маятникових систем.

Другий розділ присвячено синтезу оптимального керування для лінійної моделі оберненого маятника. Наведено математичну модель, проведено побудову LQR-регулятора та сформовано вибірку з 50000 розв'язків для подальшого навчання штучної нейронної мережі.

У третьому розділі досліджено нелінійну модель системи. Розроблено оператор керування на основі ШНМ та оптимізаційних алгоритмів рою частинок (VCTPSO). Показано здатність моделі забезпечувати переведення маятника у стійке положення при дотриманні обмежень на зусилля і переміщення візка.

Четвертий розділ містить комплексну характеристику вимог охорони праці при роботі з обладнанням маятникового типу, включаючи загальні правила, порядок дій перед, під час та після роботи, а також алгоритми дій у разі аварії.

П'ятий розділ присвячено аналізу економічної доцільності впровадження розробленої системи керування та визначенню можливих сфер практичного застосування - у транспорті, робототехніці, автоматизації, навчальному обладнанні та медичних технологіях.

Ключові слова: обернений маятник, оптимальне керування, LQR, нелінійні системи, нейронні мережі, PSO-оптимізація, стабілізація..

ВСТУП

Системи керування динамічно нестійкими об'єктами, зокрема оберненим маятником, є одними з базових моделей, на яких тестують сучасні алгоритми стабілізації та оптимального керування. У світовій практиці такі системи використовуються для досліджень у робототехніці, автономному транспорті, балансувальних платформах, дронах та інтелектуальних механізмах. В Україні ця тематика також набуває актуальності через активний розвиток інженерії, автоматизації, безпілотних систем та освітніх лабораторних комплексів.

Обернений маятник є зручною моделлю реальних балансувальних систем, адже демонструє складну нелінійну динаміку та вимагає високоточних алгоритмів керування. Розробка таких алгоритмів особливо важлива для створення енергоефективних та адаптивних пристроїв, здатних працювати в умовах змінних навантажень і параметрів.

У цій роботі досліджено синтез оптимального керування для оберненого маятника з використанням класичного LQR-регулятора та нейронних мереж, що дозволяють працювати з широким діапазоном параметрів і нелінійною динамікою. Отримані результати можуть бути застосовані в сучасних інженерних системах України та світу, включаючи персональний електротранспорт, робототехнічні платформи, навчальні стенди та автоматизовані механізми.

РОЗДІЛ 1. ОБГРУНТУВАННЯ АКТУАЛЬНОСТІ УДОСКОНАЛЕННЯ

1.1 Кількісний аналіз публікацій

На базі даних Scopus та Web of Science, буде проведений кількісний аналіз науково-технічних документів(статей) за тематикою дослідження.

Для аналізу у зазначених базах даних використовувалися запити за такими фразами:

- "Inverted pendulum optimal control",
- "Inverted pendulum stability",
- "Inverted pendulum control",
- "Inverted pendulum".

Ці запити допомагають встановити основні тенденції у використанні методів управління для оберненого маятника, його стабільності та контролю. Дані були зібрані на дату запиту – 8 жовтня 2025 року. При роботі з наукометричною базою даних Scopus запити виконувалися за назвою статті, коротким описом та ключовими словами.

У таблиці 1.1 – 1.4 занесли отримані кількісні дані

Таблиця 1.1 Кількість публікацій за темою дослідження у наукометричній базі даних Scopus по рокам

Рік											Всього в базі
2025	2024	2023	2022	2021	2020	2019	2018	2017	2016	2015	
Запит за фразою "Inverted pendulum optimal control"											
108	104	92	71	77	78	68	74	61	68	42	1643
Запит за фразою "Inverted pendulum stability"											
187	197	178	157	147	156	157	181	141	128	108	3621
Запит за фразою "Inverted pendulum control"											
426	437	417	435	412	416	414	462	398	353	369	8954
Запит за фразою "Inverted pendulum"											
495	524	511	537	523	524	526	586	487	451	453	11730

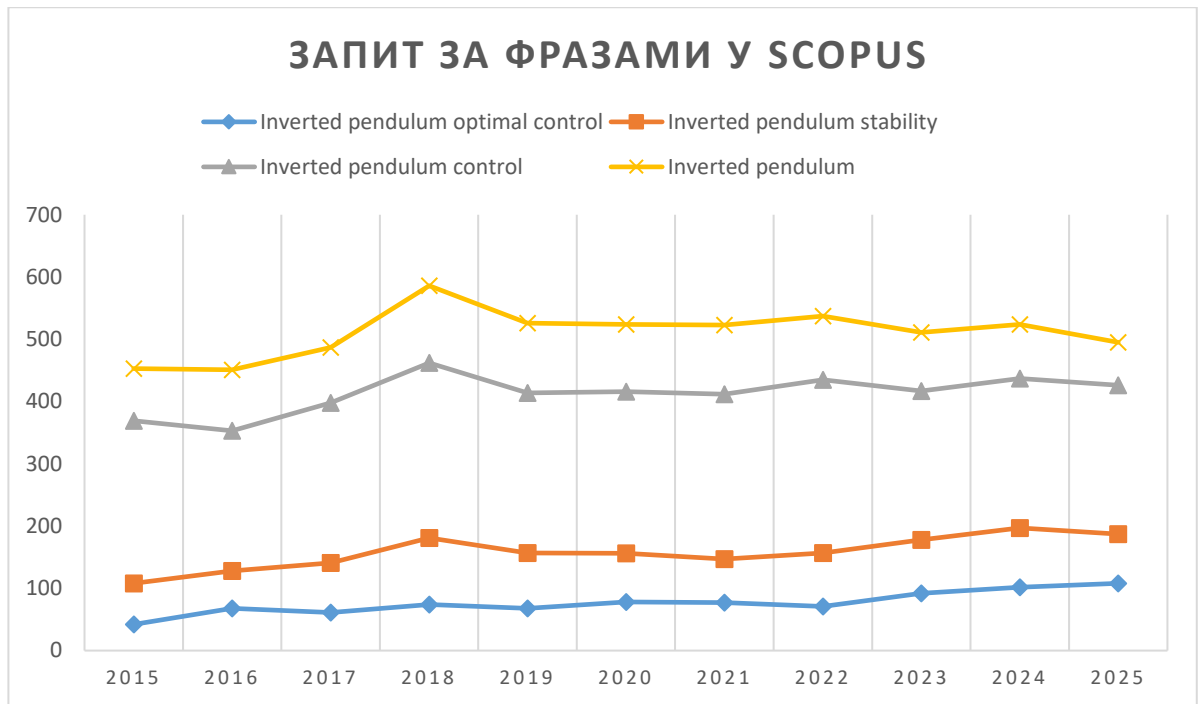


Рисунок 1.1 – Запит за фразами у Scopus

Таблиця 1.2 Кількість публікацій за країною походження

Рік											Всього в базі
2025	2024	2023	2022	2021	2020	2019	2018	2017	2016	2015	
Запит за фразою “Inverted pendulum optimal control”											
83	87	92	71	77	78	68	74	61	68	42	1254
Запит за фразою “Inverted pendulum stability”											
194	186	178	157	147	156	157	181	141	128	108	2910
Запит за фразою “Inverted pendulum control”											
421	432	417	435	412	416	414	462	398	353	369	8192
Запит за фразою “Inverted pendulum”											
531	516	511	537	523	524	526	586	487	451	453	10344

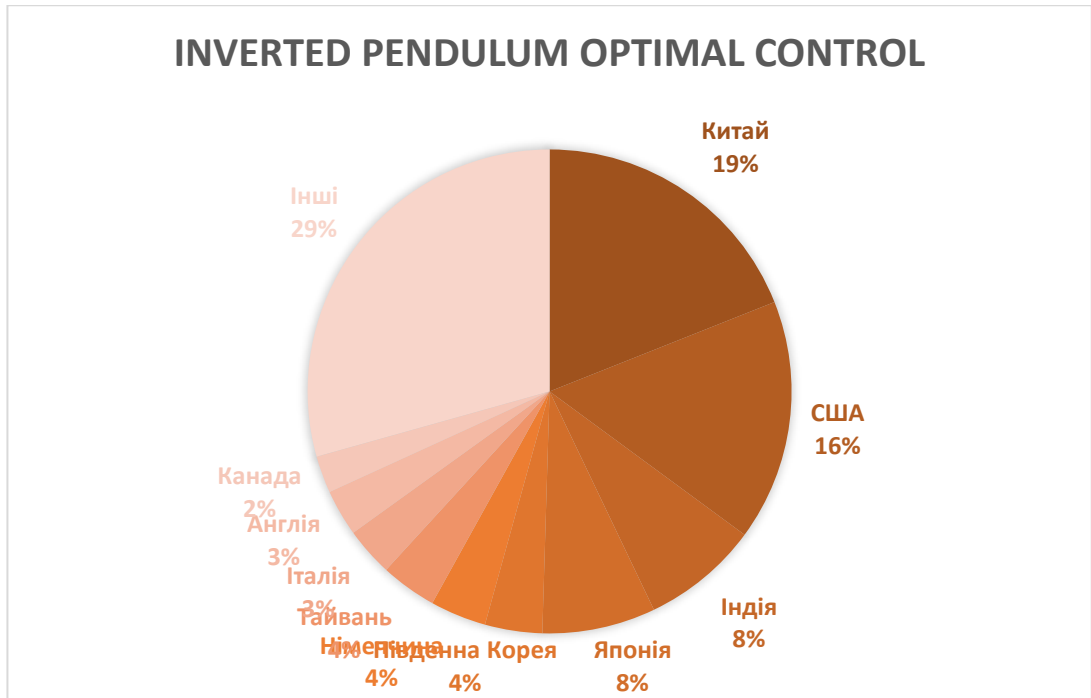


Рисунок 1.2 – Діаграма запиту по країнам

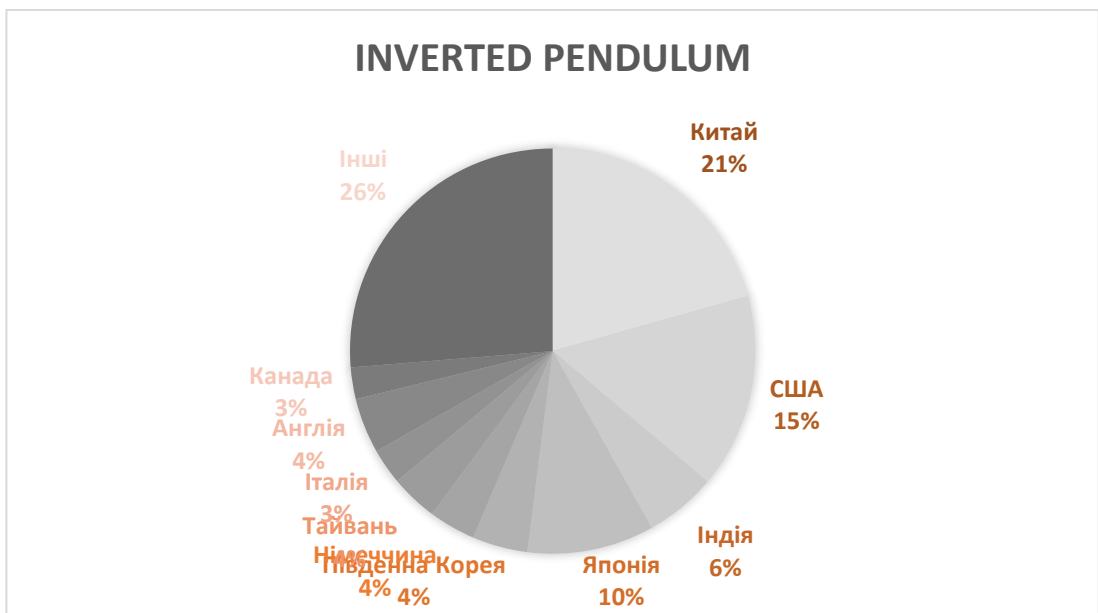


Рисунок 1.3 – Діаграма запиту по країнам

Таблиця 1.3 – Кількість публікацій за темою дослідження у наукометричній базі даних Web of Science по рокам.

Рік											Всього в базі
2025	2024	2023	2022	2021	2020	2019	2018	2017	2016	2015	
Запит за фразою “Inverted pendulum optimal control”											
48	53	56	47	65	68	60	52	50	65	45	923
Запит за фразою “Inverted pendulum stability”											
127	136	124	147	151	156	156	166	156	136	126	2413

Запит за фразу “Inverted pendulum control”											
280	297	272	314	359	354	369	382	392	356	356	6436
Запит за фразу “Inverted pendulum”											
354	370	325	383	427	428	429	466	456	432	414	7864

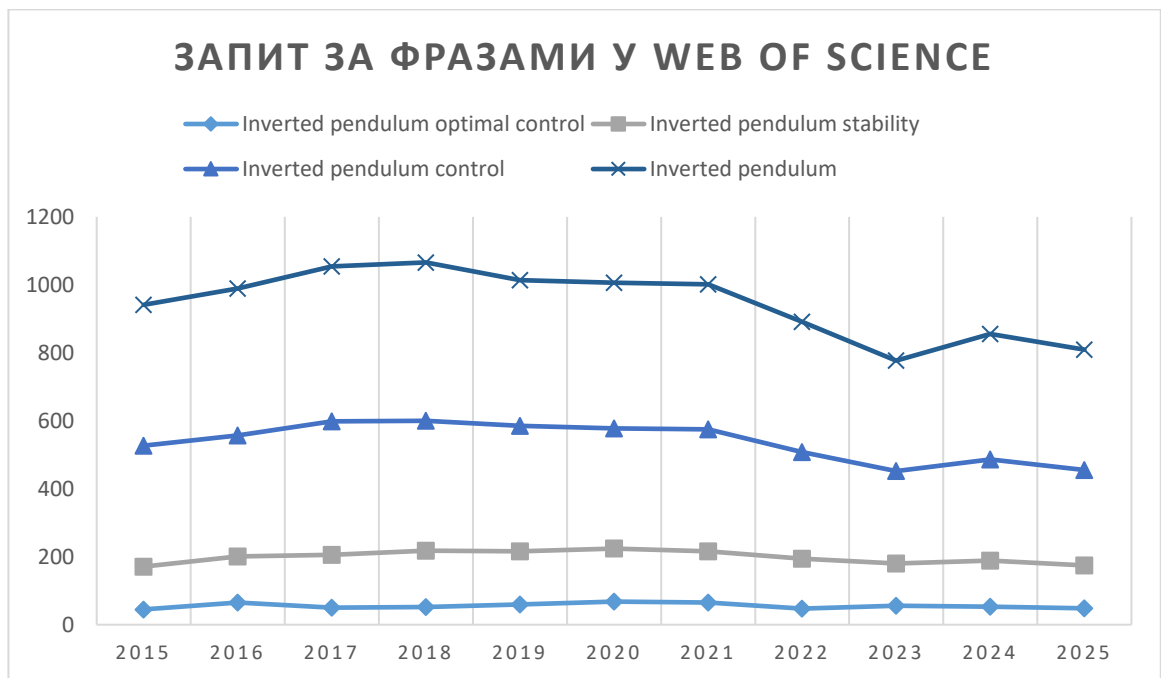


Рисунок 1.4 – Запит за фразами у Web of Science

Таблиці 1.4 – Кількість публікацій за країною походження на базі даних Web of science

Країна										Всього в базі
Китай	США	Індія	Японія	Південна Корея	Німеччина	Тайвань	Італія	Англія	Канада	
Запит за фразу “Inverted pendulum optimal control”										
182	158	62	74	33	45	33	37	31	26	923
Запит за фразу “Inverted pendulum stability”										
569	440	102	152	104	77	144	71	115	84	2413
Запит за фразу “Inverted pendulum control”										
1370	1029	326	625	314	273	321	207	261	191	6436
Запит за фразу “Inverted pendulum”										
1585	1347	378	761	388	330	347	277	346	243	7864

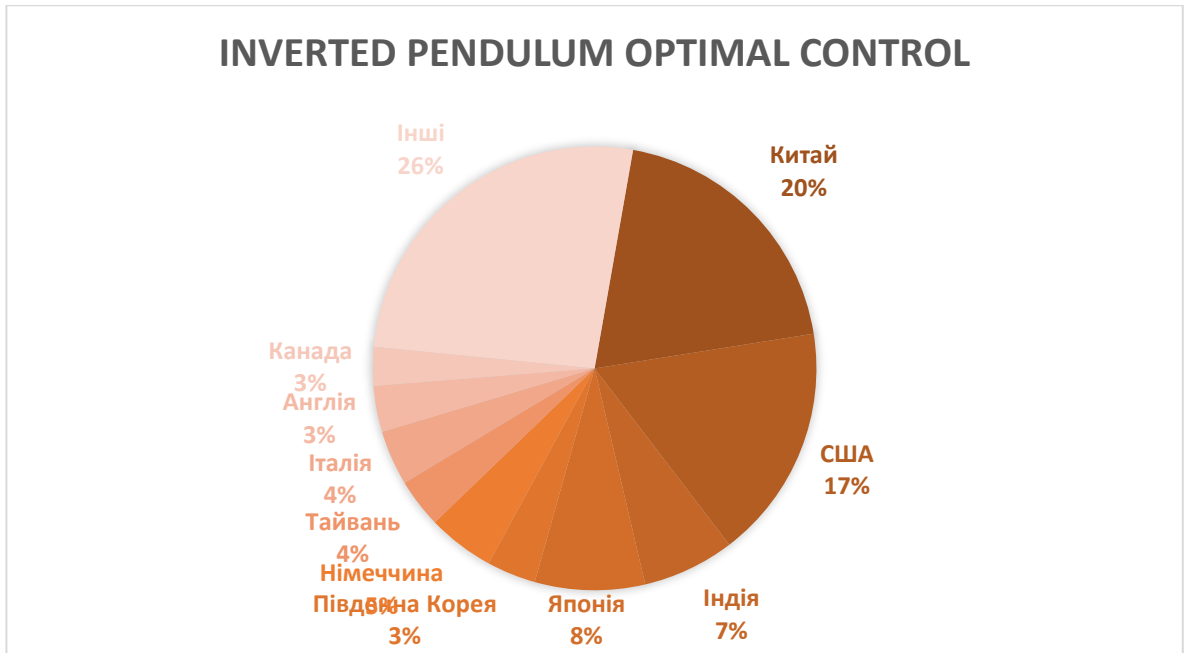


Рисунок 1.5 – Діаграма запиту по країнам

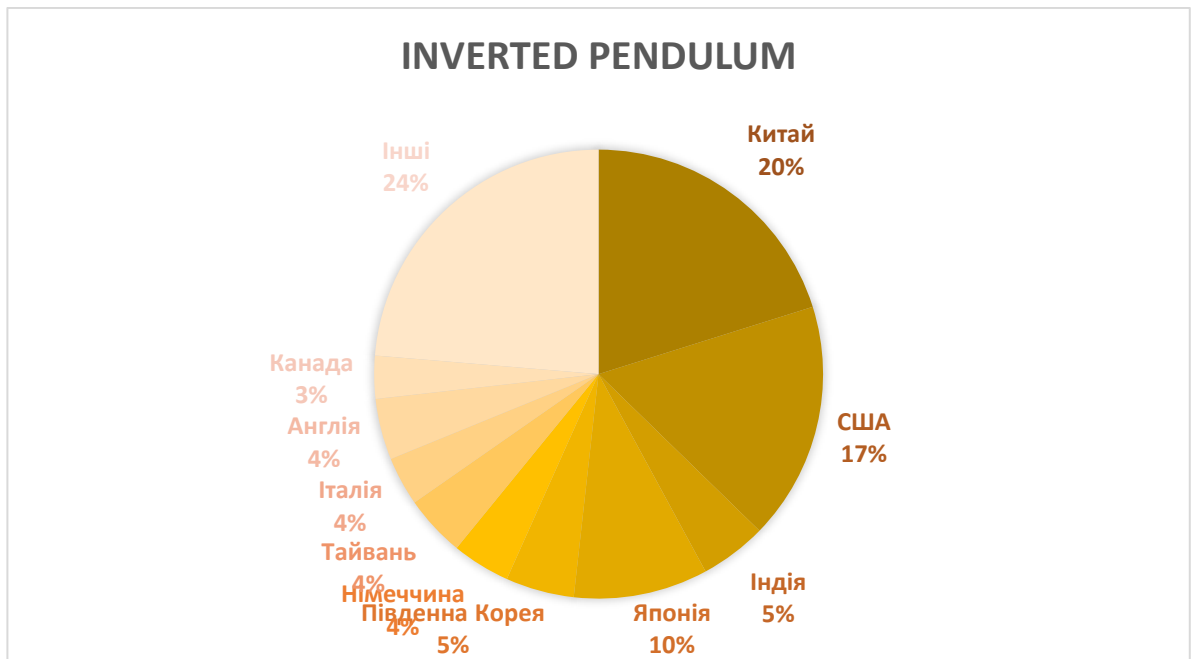


Рисунок 1.6 – Діаграма запиту по країнам

Якщо брати загальну інформацію по фразам які використовувались для запитів на ресурсах Scopus та Web of Science, можна прийти висновку, що дана тематика зберігає актуальність та тенденцію цікавості серед дослідників, які знаходять йому все більше застосувань.

За даними, які були отримані в ході аналітики публікацій, маємо висновок що тематика є цікава і актуальна для досліджування її і по сьогоднішній день.

1.2 Змістовний аналіз наукових публікацій.

Метод МПК

Модель прогнозованого керування (МПК) визначається як алгоритм контролера, що використовує оптимальні обчислення. Спосіб МПК значно вплинув на якість деяких програм такі як нафтохімічний завод, електронні пристрої, силова електроніка, робототехніка та безпілотний літальний апарат. Весь дизайн МПК в літературі був успішно реалізований і дав відмінну продуктивність систем. Проте ця модель є розробленою лише для систем, які мають чотири стану змінних. Для балансування оберненого маятника часто використовують класичні методи керування, зокрема ПД-регулятори, як показано в [8].

Пропонований МПК був розроблений для стабілізації двоступеневого перевернутого маятника що має вищий порядок, нелінійний, дуже нестабільний, багатозмінний і має 6 змінних стану. Метод МПК розроблений через динамічну модель цього маятника з використанням рівняння Ейлера-Лагранжа. Потім метод оцінюється за кількома умовами щоб визначити продуктивність за допомогою програмного забезпечення MATLAB Simulink. Результат показує, що три параметри системи, а саме параметр прогнозування горизонту N_p , налаштування параметра g_w і параметр контролю горизонту N_c може впливати на систему виводу. Параметр N_p вплинула на швидкість системи виводу. Тоді як параметри N_c і g_w впливають на максимальне відхилення кута маятника

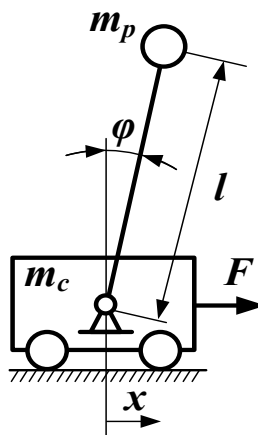


Рисунок 1.7 – Схематичне зображення одно-ступеневого оберненого маятника на рухомому візку

Контролер МПК розроблявся на основі динамічної моделі маятника за допомогою рівняння Ейлера-Лагранжа.

Метод хаотичних змінних

Щоб зменшити труднощі у виборі коефіцієнта квантування та коефіцієнта пропорції в загальному нечіткому контролері пропонується контролер із новим алгоритмом, заснованим на логістичній хаотичній змінній. Вносячи хаотичні змінні фактора в глобальному пошуку, можна отримати неоптимальне рішення та скоротити інтервал оптимізації. Глобальне оптимальне рішення отримується поблизу субоптимального рішення за інтегральним індексом ефективності.. Результати моделювання цього методу показують його ефективність.

Теорія оптимального керування була використана для розробки функції злиття для зменшення розмірності вхідних змінних нечіткого контролера для системи з подвійним перевернутим маятником. Потім змінні хаотичної послідовності були введені в систему нечіткого керування.

Нарешті запропонований метод перевірено на прикладі. Ефективність і спроможність цього підходу доведено результатами моделювання. Для подальшого вдосконалення практичного застосування запропонованого аналізу параметрів та оптимізації техніки нечіткого керування наші майбутні роботи будуть зосереджені на інших реальних системах, таких як стохастичні системи перемикання, і це буде цікавою та складною областю.

Питання мінімізації часу керування для маятникових систем із рухомою точкою підвісу детально розглянуто у фундаментальній роботі [6].

Розробка контролерів для перемикання перевернутого маятника, що самозводиться, відіграє вирішальну роль через його надзвичайно динамічну ситуацію. Рухи вгору потребують досить великої енергії та швидкої реакції, щоб миттєво затягнути маятник у вертикальне положення навколо своєї лінії рівноваги. Були реалізовані нечіткі методи керування для повороту стрижня маятника з нижнього прямого положення на 180 градусів у верхнє положення приблизно на 0 градусів.

Згідно з визначенням, оскільки маятник перебував у вертикальному перевернутому положенні, було показано, що маятник стабілізувався. Оскільки лінеаризація була дозволена в цій ситуації, посилення контролера зворотного зв'язку за станом було встановлено для стабілізації штока маятника у вертикальному положенні навколо лінії рівноваги шляхом переміщення візка вздовж горизонтальної осі. Лабораторним експериментом було показано розгойдування вгору та задачу стабілізації для перевернутого маятника можна розв'язати за допомогою поєднання нечітких контролерів і контролерів зі зворотним зв'язком. Невеликі збурення, тобто імпульс і імпульсні сили на стрижні маятника, все ж долалися методами контролю.

Системи стабілізації останнім часом набули простої класичної задачі керування. Крім того, їх методи не просто стабілізують системи, але й поліпшують їх відповідь з великим впливом. У дослідженні розглядали використання перевернутого маятника як моделі нелінійної системи, аналогічної реальним застосуванням, таким як рука робота чи системи балансування. Оптимальне управління здійснювалось за допомогою лінійного квадратичного регулятора (LQR), який спрямований на збалансування маятника. Проведено аналіз стабільності за допомогою класичного методу розміщення полюсів і порівняно результати з лінійно-квадратичним регулятором.

Методи були застосовані до перевернутого маятника для стабілізації положення візка і маятника. Техніка розміщення полюсів полегшує стабілізацію системи, розміщуючи полюси в потрібних місцях. Це досягається за допомогою

відповідної матриці підсилення для зворотного зв'язку за станом. Хоча характеристика розміщення полюсів працює добре, система покращується з використанням методів LQR. Ці методи надають оптимальне регулювання зворотного зв'язку, що дозволяє стабільній системі досягти оптимального рівня відповіді. Такий контролер працює з цільовою функцією інтегрального часу, яка мінімізується за допомогою вибору матриць зважування Q і R . В результаті застосування методу LQR показує, що значення R повинно бути значно меншим, щоб уникнути високих посилення.

1.3 Прикладні аспекти використання маятникових систем

Історія обертових маятників із рухомою точкою підвісу бере свій початок у ранніх працях з механіки та фізики. Ще у XV столітті Леонардо да Вінчі досліджував будову та поведінку маятників. Хоча він не створив конкретної конструкції обертового маятника зі змінною точкою підвісу, його ідеї та спостереження суттєво вплинули на подальший розвиток цієї галузі.

Маятникові системи з подібною будовою відомі вже багато століть. Відомості про них зустрічаються у працях давньогрецьких мислителів, зокрема Архімеда і Герона.

Однак одним із найвідоміших практичних застосувань маятника став годинниковий механізм, пов'язаний із дослідженнями Галілео Галілея у XVII столітті. Його годинник мав простий маятник, підвішений на нитці, а рівномірні коливання маятника стали основою для вимірювання часу.

У наступні століття інтерес учених до маятників із рухомою точкою підвісу не зникав. Проте справжній прорив у вивченні таких систем настав у XIX столітті,

коли французький дослідник Леон Фуко та низка інших учених почали детально аналізувати рухи маятників.

Фуко з'ясував, що зміна положення точки підвісу відкриває нові фізичні властивості та закономірності, що привело до формулювання важливих принципів механіки.

Згодом маятникові системи з рухомою точкою підвісу набули широкого використання в різних сферах:

- Навігація: у компасах, секстантах та інших приладах для визначення напрямку та координат.
- Геодезія: маятники застосовували для вимірювання сили тяжіння, що дозволяло оцінювати форму Землі.
- Фізика: для глибшого вивчення законів механічного руху.
- Інженерія: у сейсмографах, метрономах та інших приладах, де важливі стабільні коливальні процеси.

Актуальність теми підтверджується використанням подібних систем у сучасній робототехніці, зокрема в мобільних платформах, як показано в стат [3]

У ХХ столітті інтерес до маятникових механізмів лише зростав, особливо у контексті технічних застосувань.

Американський інженер Чарльз Кульман використовував маятникові принципи для створення високоточних гіроскопів, які згодом стали невід'ємними елементами авіаційних та навігаційних систем. У другій половині ХХ століття із розвитком новітніх технологій обертові маятники з рухомою точкою підвісу почали широко використовувати як моделі для створення складних систем стабілізації, навігації та автоматичного керування — від автопілотів до високоточних технічних приладів.

Сучасні застосування таких систем охоплюють:

- Гіроскопи, що використовуються в літаках, кораблях, дронах та смартфонах.
- Датчики вібрацій, встановлені у промисловому обладнанні для контролю стану механізмів.
- Медичні пристрої, зокрема в обладнанні для зчитування електричних сигналів органів, наприклад у системах ЕКГ.

Загалом маятники продовжують широко застосовуватися у різних сферах діяльності — від наукових досліджень та технічних систем до побутових приладів і навіть розважальних пристроїв.

Одним із класичних прикладів побутового застосування є маятникові годинники(Рисунок 1.1). У таких механізмах коливання маятника забезпечують рівномірну роботу внутрішніх шестерень, які рухають стрілки.



Рисунок 1.8 – Маятниковий годинник

У сучасності маятниковий принцип активно застосовується і в персональному електротранспорті.

Гіроборд частково працює на основі маятникових закономірностей(Рисунок 1.2) завдяки:

- гіроскопічному ефекту обертання коліс у протилежних напрямках, який забезпечує стійкість по горизонталі;
- датчикам нахилу, що визначають положення тіла користувача та регулюють швидкість і напрям руху.



Рисунок 1.9 - Гіроборд

Сігвей значно ближчий до моделі маятника з рухомою точкою підвісу, оскільки:

- його точка опори знаходиться між колесами, що дозволяє платформі рухатися вперед і назад;
- центр мас розташований над точкою опори, що створює нестійку рівновагу та характерні маятникові коливання;
- керування здійснюється нахилом тіла вперед або назад, що змушує конструкцію реагувати подібно до обертового маятника.



Рисунок 1.10 - Сігвей

Таким чином, маятникові системи, починаючи з найдавніших досліджень, пройшли довгий шлях розвитку: від простих експериментів і годинникових механізмів до складних сучасних технологій керування, стабілізації та навігації. Їхнє вивчення і надалі залишається актуальним у багатьох наукових і технічних галузях.

РОЗДІЛ 2. ПОСТАНОВКА І РОЗВ'ЯЗАННЯ ЗАДАЧІ СИНТЕЗУ ОПТИМАЛЬНОГО КЕРУВАННЯ ОБЕРНЕНИМ МАЯТНИКОМ У ЛІНІЙНІЙ ПОСТАНОВЦІ

2.1 Постановка задачі

У даному пункті виконується розв'язування та апроксимація масиву розв'язків LQR-задачі (задачі оптимального керування із лінійноквадратичним критерієм якості).

Основи застосування LQR-контролера для системи оберненого маятника детально описані в статті [5].

У дослідженні будемо використовувати динамічну модель руху системи „одно ступеневого оберненого маятника на рухомому візку”, яка зображена (Рисунок 1.2).

Отже розрахунки розпочинаються з математичної моделі, що наведена на Рисунку 2.1, що описує рух динамічної системи, в якій x_1 – це переміщення візка, x_2 – це переміщення вантажу, який знаходиться зверху на рухомій платформі. Далі m_1 - маса візка, m_2 – маса вантажу, що знаходиться вгорі. Літера l позначає довжину маятникової ланки, а літера F позначає рушійне зусилля, яке діє на рух візка.

Для опису об'єкта керування використано модель оберненого маятника на візку, деталі якої наведені в статті [1].

Задача полягає в тому щоб стабілізувати рух цієї системи коли вона рухається. Система є лінійною. Кут маятникової ланки є невеликим. А якщо кут є невеликим, то ми можемо таку систему лінеаризувати.

$$\begin{cases} \ddot{x}_1 m_1 - m_2 \frac{l}{g} (x_1 - x_2) = F \\ \ddot{x}_2 m_2 = m_2 \frac{l}{g} (x_1 - x_2) \end{cases}$$

Рисунок 2.1 – Математична модель

Це рівняння руху, система рівнянь руху. Ми її переводимо в форму вхід, стан, вихід. Це робиться наступним чином, ми записуємо наше рівняння кодом(Рисунок 2.2).

```
In[14]: SSM = StateSpaceModel[{x1''[t] * m1 - m2 * 1/g * (x1[t] - x2[t]) == F[t], x2''[t] == 1/g * (x1[t] - x2[t])},
    {{x1[t], 0}, {x2[t], 0}, {x1'[t], 0}, {x2'[t], 0}}, {{F[t], 0}}, {x1'[t], t}]
```

Рисунок 2.2 – Математична модель у вигляді коду

Змінні стану, записані наступним чином $x_1 - \{x_1[t], 0\}$, та $x_2 - \{x_2[t], 0\}$

$\{x_1'[t], 0\}$ та $\{x_2'[t], 0\}$ це швидкості. Тобто змінні стану, це положення двох елементів і їхні швидкості. Рушійне зусилля - $\{F[t], 0\}$, це той фактор, який є рушійним, який є керуванням. Виходом цієї системи є $\{x_1'[t]\}$ – швидкість візка.

Після переведення ми отримуємо матричну структуру(Рисунок 2.3).

$$\text{Out[1]} = \left(\begin{array}{cccc|c} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{1}{m_2} & -\frac{1}{m_2} & 0 & 0 & \frac{1}{m_1} \\ g & g & 0 & 0 & 0 \\ \frac{1}{g} & -\frac{1}{g} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{array} \right)^s$$

Рисунок 2.3 – Матрична структура математичної моделі

Рівняння типу вхід\стан\вихід має вигляд,

$$X' = AX + BU$$

$$Y = CX + DU$$

Де, X' – це вектор змінних стану(Рисунок 2.4), U – це рушійне зусилля F , це те що діє на систему(Рисунок 2.5), Y – це вихід із системи, тобто це те що ми вимірюємо, ми вимірюємо швидкість руху візка(Рисунок 2.6). Тобто можна поставити датчик і він буде вимірювати цей сигнал. Літери A, B, C, D – це матриці, вони зображені на Рисунок 2.7.

$$\{\{x_1[t], 0\}, \{x_2[t], 0\}, \{x_1'[t], 0\}, \{x_2'[t], 0\}\}$$

Рисунок 2.4 – Вектор змінних, X'

$$\{\{F[t], 0\}\}$$

Рисунок 2.5 – Рушійне зусилля, U

$$\{x_1'[t]\}$$

Рисунок 2.6 – Вихід із системи, Y

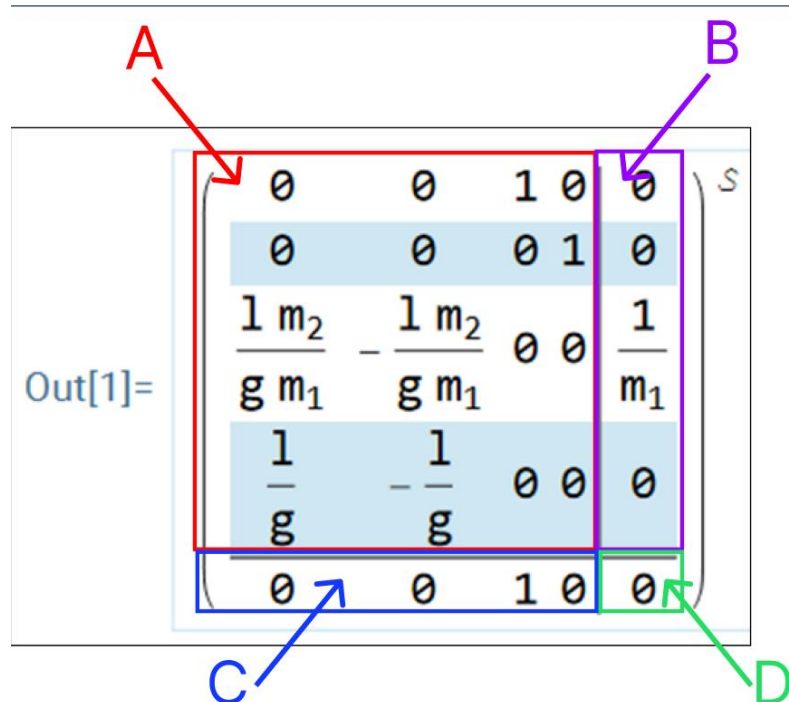


Рисунок 2.7 – Пояснення розміщення матриць і їх позначення

Отримуємо ось таку структуру. Рівняння руху це одна із форм математичних моделей. Нам ця форма не потрібна. Ми перейшли у форму вхід\стан\вихід.

2.2 Розробка моделі нейронної мережі

Маючи цю форму ми можемо синтезувати лінійно-квадратичний регулятор.

Задаємо певні чисельні параметри – Рисунок 2.8, де $lMax, lMin$ це максимальне та мінімальне значення маятникової системи, а $m2Max$ та $m2Min$ це максимальна та мінімальна вага маятникової системи. Також задаємо початкові умови, що зображені на Рисунку 2.8.1, де $InitialStateVector = List[0$ – це початкове положення, 0 – також початкове положення, тобто $x1$ та $x2$, а швидкість, тобто $x1'$ та $x2' = v$].

Це означає, що наша система рухається і вона має певну швидкість і нам потрібно зупинити саме в моменті вирівнювання системи в вертикальному положенні вгору. Тобто під кінець ми гасимо нашу швидкість, але положення в якому ми почали гальмувати, ми в нього маємо повернутись і тут зупинитись. Саме це означають початкові умови, $0, 0$.

```
lMax = 0.4;
```

```
lMin = 1.2;
```

```
m2Max = 80;
```

```
m2Min = 10;
```

Рисунок 2.8 – Задання числових параметрів

Після задання чисельних параметрів, отримуємо 4 коефіцієнти регулятора, так як ми маємо змінних стану 4. І 4 коефіцієнтів множаться на кожен із компонентів змінних стану. І отримуємо ось таке регулювання.

Далі для того щоб задати регулювання, треба задавати вагові матриці Q та R, Рисунок 2.9 та 2.10.

```
Q = DiagonalMatrix[{10, 10, 50, 50}];
```

Рисунок 2.9 – Матриця вагових коефіцієнтів інтегрального критерію оптимізації – біля фазових змінних

```
R = {{0.005}};
```

Рисунок 2.10 - Матриця вагових коефіцієнтів інтегрального критерію оптимізації – біля керування

Матриця R говорить наскільки важливо отримати керування, тобто наскільки потрібно мінімізувати керування. А матриця Q говорить наскільки нам важливо мінімізувати кожен із компонентів змінних стану.

Після того як задали всі потрібні параметри, можна будувати графічні залежності. Графічна залежність, того як змінюється рушійна сила, зображено на Рисунку 2.11.

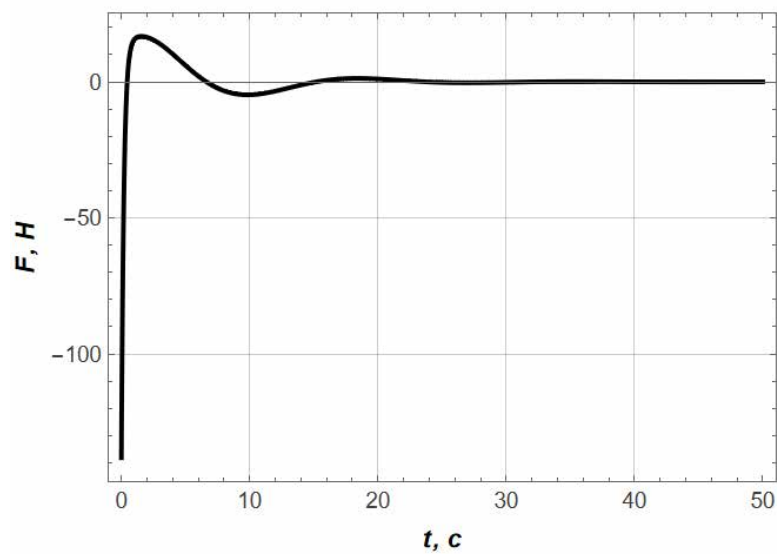


Рисунок 2.11 – Графічна залежність зміни рушійної сили в часі

Далі фазовий портрет, він будується в координатах x_1-x_2 по X , та $x_1'-x_2'$ по Y , зображено на Рисунку 2.12. Він починається з 0 і повинен прийти назад в 0. Тобто це означає що коливання маятникової системи ми усунули.

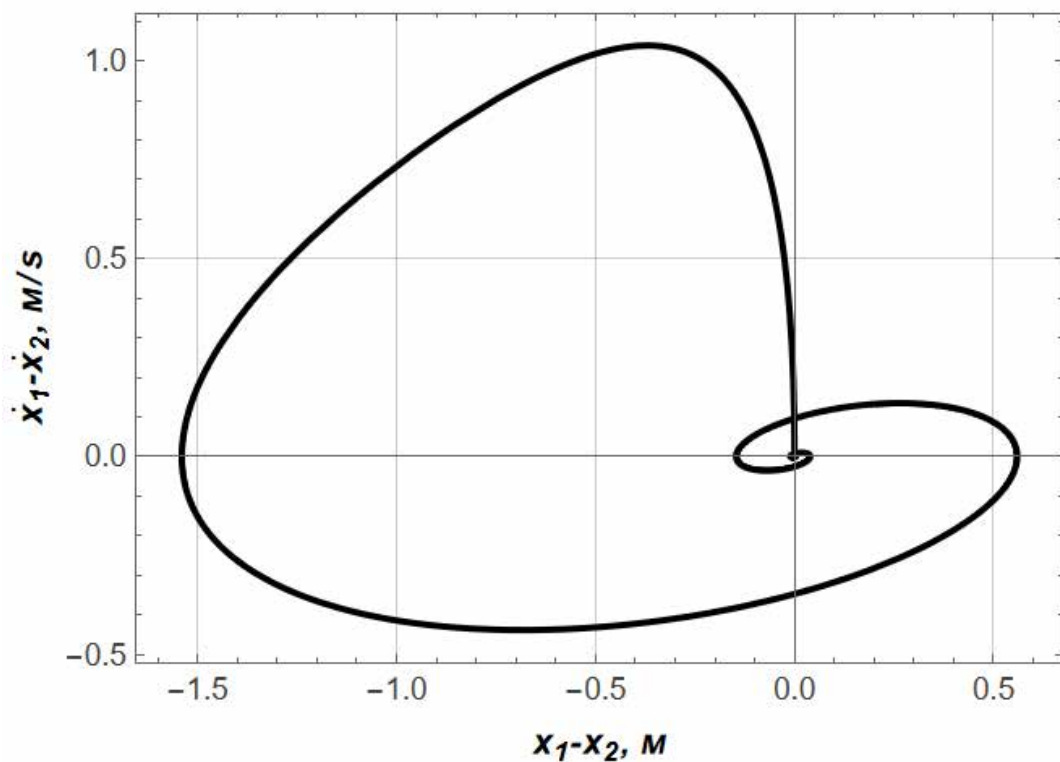


Рисунок 2.12 – Фазовий портрет

І останнє це \dot{x}_1 , це те що ми вимірюємо (Рисунок 2.13). Наприклад ми задали що візок рухається швидкістю пів метра на секунду, а візку потрібно зупинитись і повернутися в місце, звідки починався процес гальмування. Це і є суть нашої задачі.

На графіку видно як він починає зупинятися, після чого змінює напрямок руху декілька разів, і врешті решт зупиняється. Візку знадобилось 40 секунд, щоб загальмувати і знайти свій фінальний стан, стан спокою.

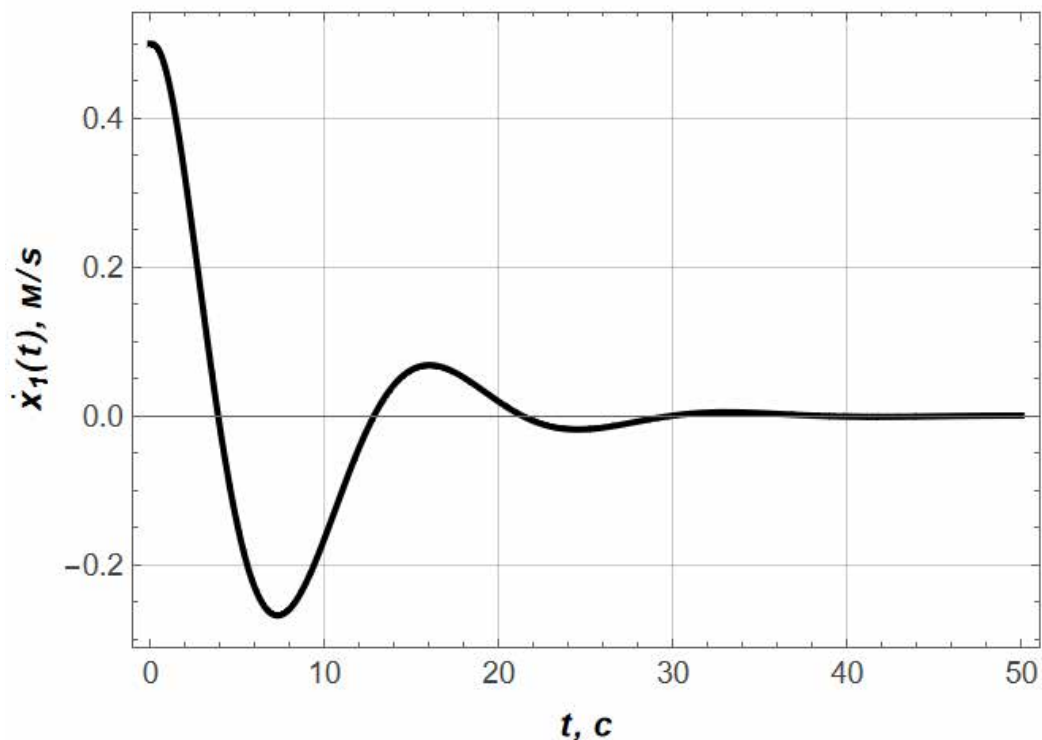


Рисунок 2.13 – Графічна залежність руху візка під час гальмування

Це приклад одного розв'язку LQR.

Наша задача полягає в іншому, нам потрібно знайти загальний розв'язок задачі LQR. Ми це робимо наступним чином.

Кодом задаємо числове значення вибірки у розмірі 50000(Рисунок 2.14).

```
In[27]:=
LQRSolutionsList = List[];
SolutionsNumber = 50000; (*Об'єм вибірки 100000*)
```

Рисунок 2.14 – Задання значення вибірки

Таким чином ми створюємо масив даних на яких ми будемо навчати нейронну мережу. Одиничка, тобто кожен розв'язок із 50 000. Це те що було продемонстровано раніше як один розв'язок задачі LQR.

Але це масив розв'язків ми набиваємо на випадкових значення параметрів. Тут також маємо два параметра(Рисунок 2.15), де l – це довжина маятникової системи, а m_2 – це маса маятникової системи..

```
(*Задання меж змін параметрів системи*)
lMax = 0.4;
lMin = 2;

m2Max = 80;
m2Min = 10;

(*Випадкове задання параметрів системи*)
l = RandomReal[{lMin, lMax}]
m2 = RandomReal[{m2Min, m2Max}]
```

Рисунок 2.15 – Задання випадкових значень параметрів

Задання випадкових параметрів робиться для того, щоб нашу систему можна було впровадити в системи керування гіробордів або інших пристроїв. Тобто будь яка людина може стати на гіроборд і її вага вже буде прорахована. Показники максимальних та мінімальних параметрів можна змінити за необхідності.

Після закінчення розрахунків, отримуємо масив даних(рисунок 2.16).

In[39]:- LQRSolutionsList

```
{[0.594158, 50.2387, 72.8222, -9.5766, 113.635, 206.081], [1.05163, 33.7993, 81.9018, -18.6562, 115.222, 163.313], [1.06185, 67.6923, 86.4093, -23.1637, 116.002, 163.267], [0.753219, 78.4052, 79.4496, -16.204, 114.795, 187.535], [1.04294, 27.0997, 80.9233, -17.6778, 115.052, 163.801], [0.503023, 67.8689, 71.2868, -8.04126, 113.364, 220.986], [1.76746, 76.0931, 104.446, -41.2007, 119.071, 133.974], [1.22284, 65.8607, 89.9939, -26.7484, 116.618, 154.364], [1.9145, 60.8174, 103.874, -40.628, 118.975, 129.352], [0.815632, 75.7307, 80.9446, -17.699, 115.056, 181.571], [1.800479, 25.9992, 93.8628, -30.6173, 117.28, 131.787], [1.07899, 56.6181, 85.3678, -22.1223, 115.822, 161.985], [1.41489, 56.6687, 92.7433, -29.4977, 117.089, 145.524], [0.744353, 11.718, 73.4928, -10.2472, 113.753, 187.656], [1.44597, 12.4075, 85.7896, -22.5441, 115.895, 143.806], [1.20403, 25.8279, 83.7779, -20.5324, 115.547, 154.664], [1.68208, 76.7797, 102.649, -39.4031, 118.768, 136.601], [0.66784, 28.3481, 73.1202, -9.87463, 113.687, 196.229], [0.84414, 64.9585, 80.6287, -17.3832, 115.001, 178.856], [1.46509, 67.879, 95.878, -32.6325, 117.623, 143.846], [0.890264, 78.0488, 83.2258, -19.9802, 115.451, 175.38], [1.47844, 51.5217, 93.1289, -29.8834, 117.154, 142.914], [1.09121, 10.0729, 79.7564, -16.5109, 114.849, 160.816], [0.798411, 31.6974, 76.355, -13.1094, 114.255, 182.465], [0.89707, 42.0751, 79.5554, -16.3099, 114.814, 174.169], [1.02666, 13.5998, 79.0455, -15.7999, 114.725, 164.787], [1.38716, 71.1595, 94.6935, -31.448, 117.421, 147.057], [1.35979, 31.011, 87.3605, -24.1149, 116.165, 147.408], [1.21761, 63.0397, 89.4445, -26.199, 116.524, 154.553], [1.44823, 16.315, 86.465, -23.2195, 116.011, 143.712], [1.92408, 77.0909, 108.235, -44.9896, 119.705, 129.668], [1.5123, 54.5377, 94.3769, -31.1313, 117.367, 141.713], [1.82812, 37.29, 96.7108, -33.4652, 117.764, 131.219], [1.02285, 38.1514, 81.836, -18.5905, 115.21, 165.184], [1.80433, 74.1466, 104.813, -41.5671, 119.132, 132.83], [1.62827, 39.133, 93.6155, -30.3699, 117.237, 137.367], [1.04289, 24.7221, 80.6396, -17.3941, 115.003, 163.791], [0.725888, 23.5633, 74.0626, -10.8171, 113.853, 189.624], [1.87166, 44.2712, 99.0635, -35.8179, 118.163, 130.115], [1.30517, 20.4593, 84.7796, -21.534, 115.72, 149.761], [1.50154, 23.9946, 88.6254, -25.3799, 116.383, 141.691], [1.62884, 78.3172, 101.747, -38.5019, 118.617, 138.377], [0.581768, 77.2753, 74.3004, -11.0549, 113.895, 208.261], [1.18555, 69.3925, 89.6437, -26.3981, 116.558, 156.354], [1.71903, 18.4224, 90.9934, -27.7478, 116.789, 134.32], [1.75661, 29.6355, 93.8667, -30.6212, 117.28, 133.221], [1.44228, 33.8406, 89.3092, -26.0636, 116.501, 144.038], [0.863934, 54.5731, 80.0705, -16.8249, 114.904, 177.009], [1.59992, 41.7751, 93.6211, -30.3755, 117.238, 138.36], [0.639556, 42.6913, 73.485, -10.2394, 113.751, 199.859], [0.922472, 54.4656, 81.4645, -18.219, 115.146, 172.405], [1.18962, 79.5599, 91.2702, -28.0247, 116.837, 156.446], [1.97113, 66.9205, 106.593, -43.3472, 119.431, 128.078], [0.483664, 65.105, 70.5315, -7.28598, 113.231, 224.583], [1.68941, 61.2849, 99.3919, -36.1463, 118.219, 135.857], [1.60106, 14.9632, 86.5733, -25.3278, 116.374, 138.137], [1.78716, 52.1886, 99.3239, -36.0783, 118.207, 132.662], [1.19402, 46.3019, 86.3918, -23.1462, 115.999, 155.596], [1.94041, 15.5402, 93.5685, -30.3229, 117.229, 128.083], [0.992515, 56.3028, 83.3265, -20.081, 115.469, 167.462], [0.717373, 10.7224, 72.8605, -9.61493, 113.642, 190.504], [1.28393, 45.5804, 88.2036, -24.958, 116.311, 150.987], [1.28017, 53.5427, 89.366, -26.1204, 116.51, 151.311], [1.00081, 74.2001, 85.7033, -22.4578, 115.88, 167.312], [0.688706, 69.2454, 76.8377, -13.5921, 114.339, 194.265], [0.94369, 32.4481, 79.5397, -16.2942, 114.811, 170.553], [1.77658, 19.0657, 91.9789, -28.7334, 116.958, 132.591], [1.45654, 73.6067, 96.7601, -33.5145, 117.773, 144.357],
```

Рисунок 2.16 – Масив даних після розрахунку

Масив містить 50 000 різних значень параметрів. Візьмемо один для прикладу(Рисунок 2.17). Перші два числа(1.07899, 56.6181) це l та m2. Це випадково задані параметри. Вони змінюються в межах, які ми задавали. А наступні 4 числа(85.3678, -22.1223, 115.822, 161.985) це наші 4 коефіцієнти регулятора.

Тобто ми задаємо випадкові параметри і під них шукаємо відповідні коефіцієнти регулятора.

{1.07899, 56.6181, 85.3678, -22.1223, 115.822, 161.985}

Рисунок 2.17 – Випадкове значення параметрів з масиву

Масив отримано, можна починати тренування ШНМ.

2.3 Тренування нейронної мережі

Для початку тренувань, зводимо отриману бурю різних значень до масштабу від -1 до 1. Тобто найбільше значення буде відповідати 1, а найменше 0. Це потрібно для того, щоб наша нейронна мережа могла потренуватись на цих даних, щоб вона не зайшла в насичення(параліч).

Нормалізація виконується для кожного параметру, для l(Рисунок 2.18), для m2(Рисунок 2.19). І також для коефіцієнтів (Рисунок 2.20)

```
lUpper = Max [Transpose [LQRSolutionsList] [[1]]];  
lLower = Min [Transpose [LQRSolutionsList] [[1]]];
```

Рисунок 2.18 – Нормалізація параметру L

```
m2Upper = Max [Transpose [LQRSolutionsList] [[2]]];  
m2Lower = Min [Transpose [LQRSolutionsList] [[2]]];
```

Рисунок 2.19 – Нормалізація параметру m2

```
G1Upper = Max [Transpose [LQRSolutionsList] [[3]]];  
G1Lower = Min [Transpose [LQRSolutionsList] [[3]]];  
  
G2Upper = Max [Transpose [LQRSolutionsList] [[4]]];  
G2Lower = Min [Transpose [LQRSolutionsList] [[4]]];  
  
G3Upper = Max [Transpose [LQRSolutionsList] [[5]]];  
G3Lower = Min [Transpose [LQRSolutionsList] [[5]]];  
  
G4Upper = Max [Transpose [LQRSolutionsList] [[6]]];  
G4Lower = Min [Transpose [LQRSolutionsList] [[6]]];
```

Рисунок 2.20 – Нормалізація коефіцієнтів

Після нормалізації, записуємо всі отриманні данні в відповідні масиви(Рисунок 2.21).

$$lList = N \left[\frac{\text{Transpose}[LQRSolutionsList][[1]] - lLower}{lUpper - lLower} * 2 - 1 \right];$$

$$m2List = N \left[\frac{\text{Transpose}[LQRSolutionsList][[2]] - m2Lower}{m2Upper - m2Lower} * 2 - 1 \right];$$

$$G1List = N \left[\frac{\text{Transpose}[LQRSolutionsList][[3]] - G1Lower}{G1Upper - G1Lower} * 2 - 1 \right];$$

$$G2List = N \left[\frac{\text{Transpose}[LQRSolutionsList][[4]] - G2Lower}{G2Upper - G2Lower} * 2 - 1 \right];$$

$$G3List = N \left[\frac{\text{Transpose}[LQRSolutionsList][[5]] - G3Lower}{G3Upper - G3Lower} * 2 - 1 \right];$$

$$G4List = N \left[\frac{\text{Transpose}[LQRSolutionsList][[6]] - G4Lower}{G4Upper - G4Lower} * 2 - 1 \right];$$

Рисунок 2.21 – Групування даних по масивах

Після нормалізації і отримання готової вибірки, вона розділяється на дві підвибірки. Перша вибірка це навчальна, на ній ми будемо навчати, а друга вибірка, це тестувальна вибірка.

Тобто вибірка розділяється в пропорціях 90\10. 90% це навчальна вибірка, а 10% це тестувальна. Код з розподілом вибірки показано на Рисунку 2.22.

```
In[57]:= (*Формування навчальної та тестової вибірок*)
TrainingData = List[]; (*Навчальна вибірка*)
TestData = List[]; (*Тестова вибірка*)
Do[If[RandomReal[] < 0.9,
  q = RandomInteger[{1, Length[LQRSolutionsList]}];
  AppendTo[TrainingData, {lList[[q]], m2List[[q]]} -> {G1List[[q]], G2List[[q]], G3List[[q]], G4List[[q]]}],
  AppendTo[TestData, {lList[[q]], m2List[[q]], G1List[[q]], G2List[[q]], G3List[[q]], G4List[[q]]}], {q, 1, Length[LQRSolutionsList]}];
Length[LQRSolutionsList]
Length[TrainingData]
Length[TestData]

Out[60]= 50 000
Out[61]= 44 934
Out[62]= 5066
```

Рисунок 2.22 – Розподіл вибірки на навчальну та тестову

Як бачимо з 50 000, ми отримали 44 934 навчальних та 5066 тестових. Числа можуть змінювати від перерахунків.

Далі обираємо випадкову навчальну пару(Рисунок 2.23).

```
(*Випадкова навчальна пара*)
```

```
s = RandomInteger[{1, Length[TrainingData]}];  
TrainingData[[s]]
```

Рисунок 2.23 – Випадкова навчальна пара

Ми отримуємо масив, в якому перші два числа це параметри, а інші чотири числа це коефіцієнти керування(Рисунок 2.24).

```
{-0.614269, -0.689565} → {-0.648417, 0.648417, -0.638525, 0.113036}
```

Рисунок 2.24 – Масив значень

Далі розробляємо модель ШНМ (Рисунок 2.25). В якій вказуємо Input – 2, тобто два параметра, два перших числа з масиву, і на вихід чотири коефіцієнти, інші чотири числа з масиву.

```
(*Розробка моделі ШНМ*)
```

```
NetModeln = NetInitialize @NetChain[{5, Tanh, 5, Tanh, 4, Tanh}, "Input" → 2]  
(*LogisticSigmoid*)
```

Рисунок 2.25 – Створення моделі ШНМ

Нейронну мережу ми створюємо з двома прихованими шарами, і в кожному прихованому шарі по 5 нейронів. Активаційна функція тангенс гіперболічний. Тобто ми маємо вхідний шар (Input), він не виконує нічого, далі два прихованих шари(5, Tanh, 5, Tanh), та вихідний шар(4, Tanh). У вихідному шарі чотири виходи, тому що ми маємо отримати чотири числа на виході з нейронної мережі.

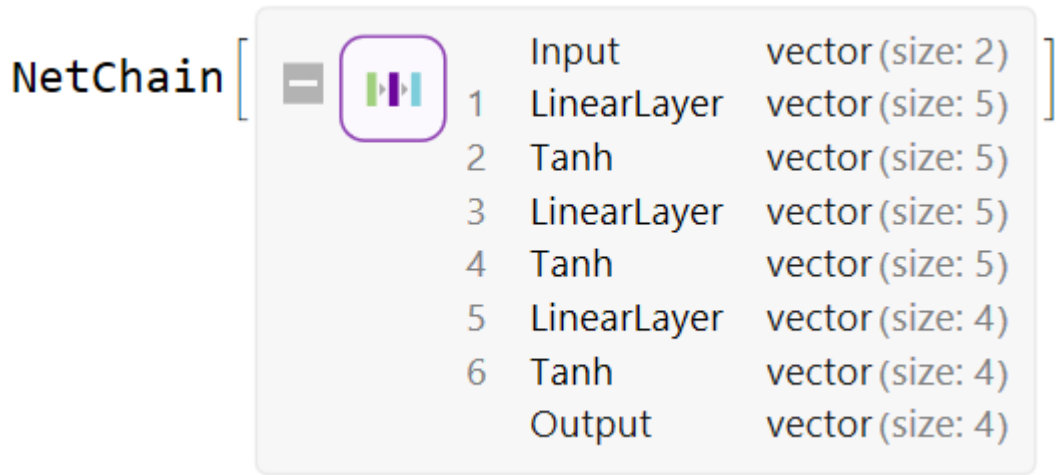


Рисунок 2.26 – Вигляд нейронної мережі за слоями

NetGraph [NetModelIn]

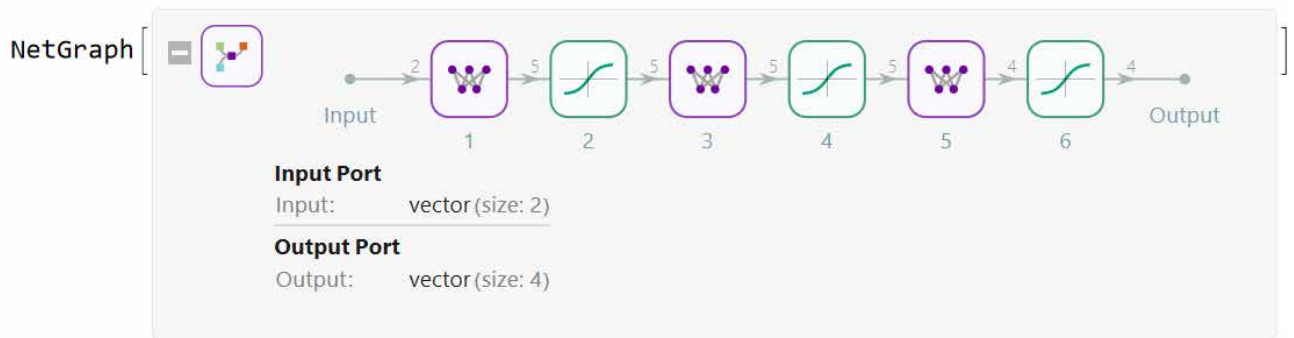


Рисунок 2.27 – Графічне представлення моделі

Далі ми починаємо тренувати ШНМ(Рисунок 2.28). Тренування проводиться методом ADAM. Розмірність пакетів становить 200. Кількість раундів становить 10 000.

(*Тренування ШНМ*)

```
TrainedNet = NetTrain[NetModelIn, TrainingData, All, BatchSize -> 200, Method -> "ADAM", MaxTrainingRounds -> 10 000, ValidationSet -> Scaled[0.2]]
```

Рисунок 2.28 – Тренування ШНМ

Тренування нейронної мережі відбувалося протягом 444 хвилин, на графіку видно тренувальний графік(жовтого кольору) та валідаційний графік(синього кольору). З їхнього руху впливає те що вони зменшуються, а якщо зменшуються то прогнозування нейронної мережі працює.

Починаємо тренувати ШНМ(Рисунок 2.28). Тренування проводиться методом ADAM. Розмірність пакетів становить 200. Кількість раундів становить 10 000.

(*Тренування ШНМ*)

```
TrainedNet = NetTrain[NetModelIn, TrainingData, All, BatchSize -> 200, Method -> "ADAM", MaxTrainingRounds -> 10000,  
ValidationSet -> Scaled[0.2]]
```

Рисунок 2.28 – Тренування ШНМ

Тренування нейронної мережі відбувалося протягом 38 хвилин, на графіку видно тренувальний графік(жовтого кольору) та валідаційний графік(синього кольору). З їхнього руху випливає те що вони зменшуються, а якщо зменшуються то прогнозування нейронної мережі працює.

Загалом було використано 1 800 000 пакетів та 10 000 раундів для тренування.

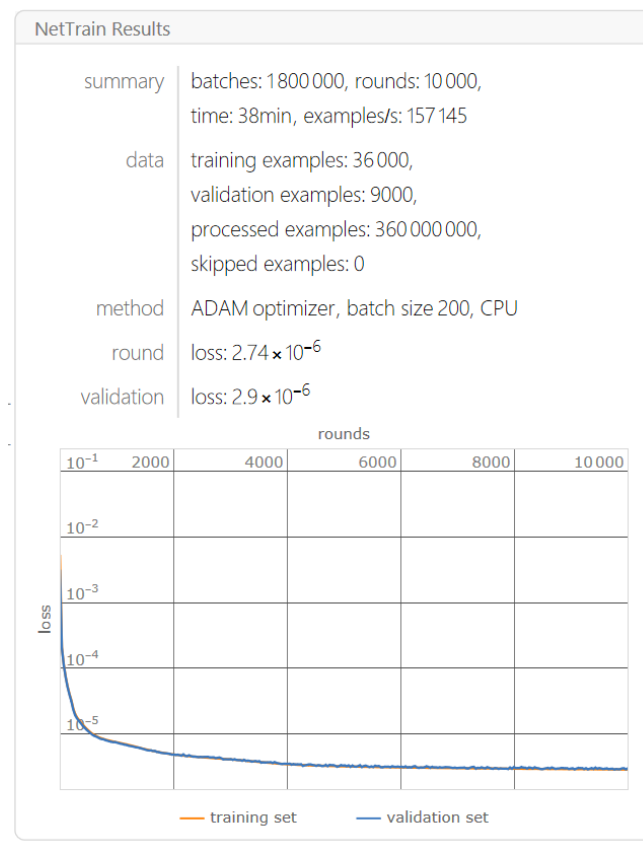


Рисунок 2.29 – Результат тренування ШНМ

2.4 Аналіз результатів тренування

Побудуємо окремо графік для аналізу результатів тренування(Рисунок 2.30).

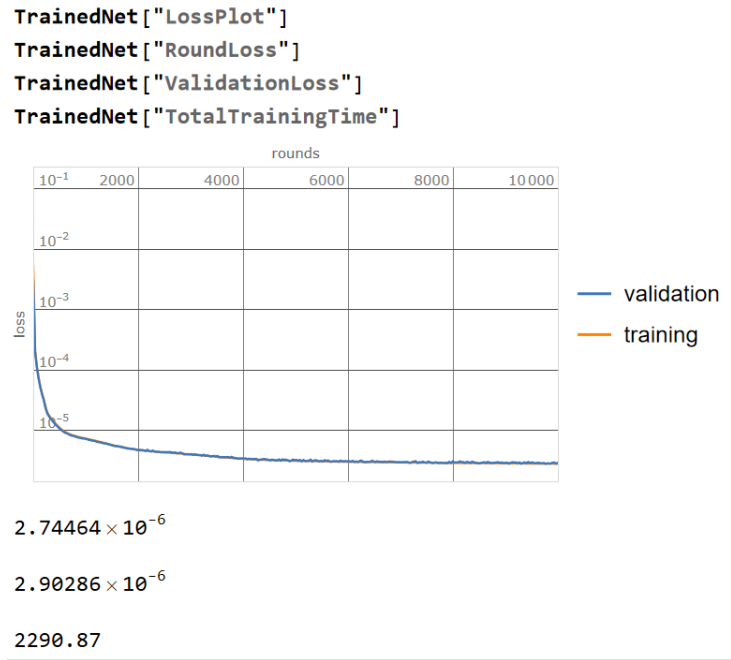


Рисунок 2.30 – Графік оцінки результатів тренування ШНМ

Як бачимо похибка в кінці тренування складає $-2.74464 \cdot 10^{-6}$. Похибка на валідаційні вибірці складає $-2.90286 \cdot 10^{-6}$. Та кількість секунд які були затрачені на тренування нейронної мережі складає 2290 с, тобто 38 хвилин.

Далі застосовуємо натреновану нейронну мережу до тестових даних і отримуємо результат(Рисунок 2.31).

i = 1;

```

TrainedNet["TrainedNet"][{TestData[[i]][1], TestData[[i]][2]}]
{-0.713614, 0.713614, -0.705207, 0.18946}

```

Рисунок 2.31 – Застосування тестових даних до натренованої нейронної мережі

Але це тільки для однієї пари, а нам потрібно пройтись по всій вибірці, тому виконуємо аналіз(Рисунок 2.32).

```

(*Оцінка результатів тренування ШНМ на тестовій вибірці*)
(*абсолютна*)
AbsolutErrorsANNList =
  Table[TrainedNet["TrainedNet"][{TestData[[i]][1], TestData[[i]][2]}] -
    {TestData[[i]][3], TestData[[i]][4], TestData[[i]][5], TestData[[i]][6]},
    {i, 1, Length[TestData]}];
(*Відносна*)
ErrorsANNList = AbsolutErrorsANNList * 100;

```

Рисунок 2.32 – Оцінка результатів тренування ШНМ на тестовій вибірці

І після цього будемо графіки та дивимось, що вони нам покажуть.

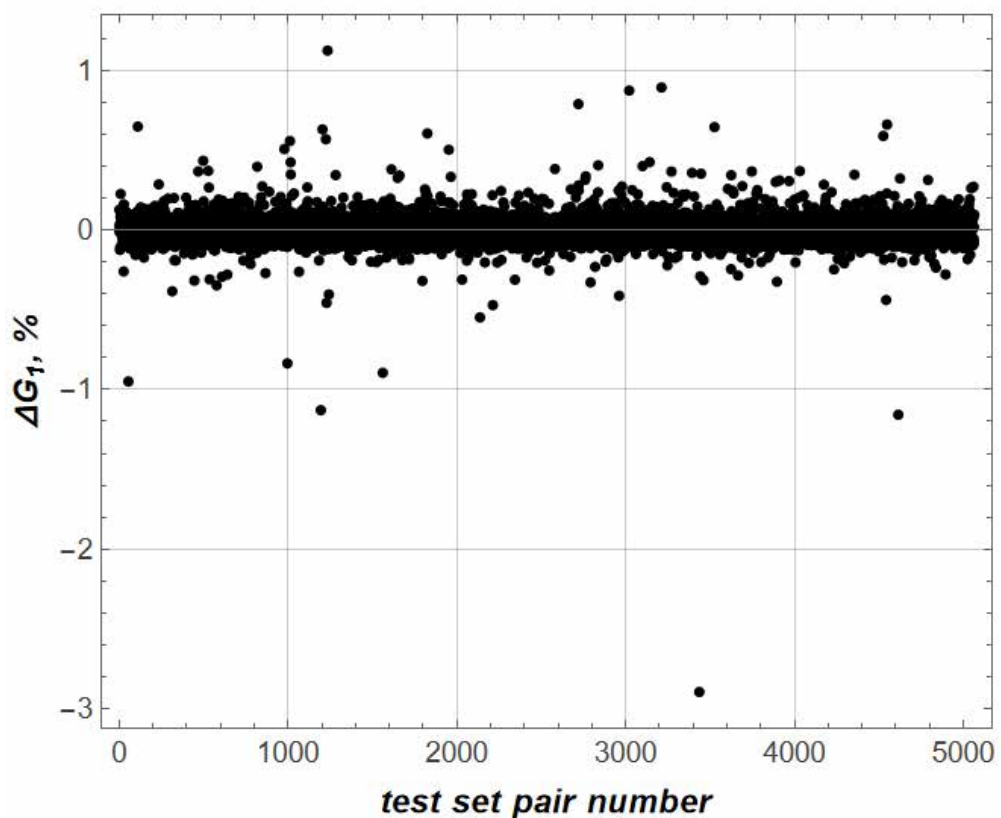


Рисунок 2.33 – Графік похибки G1

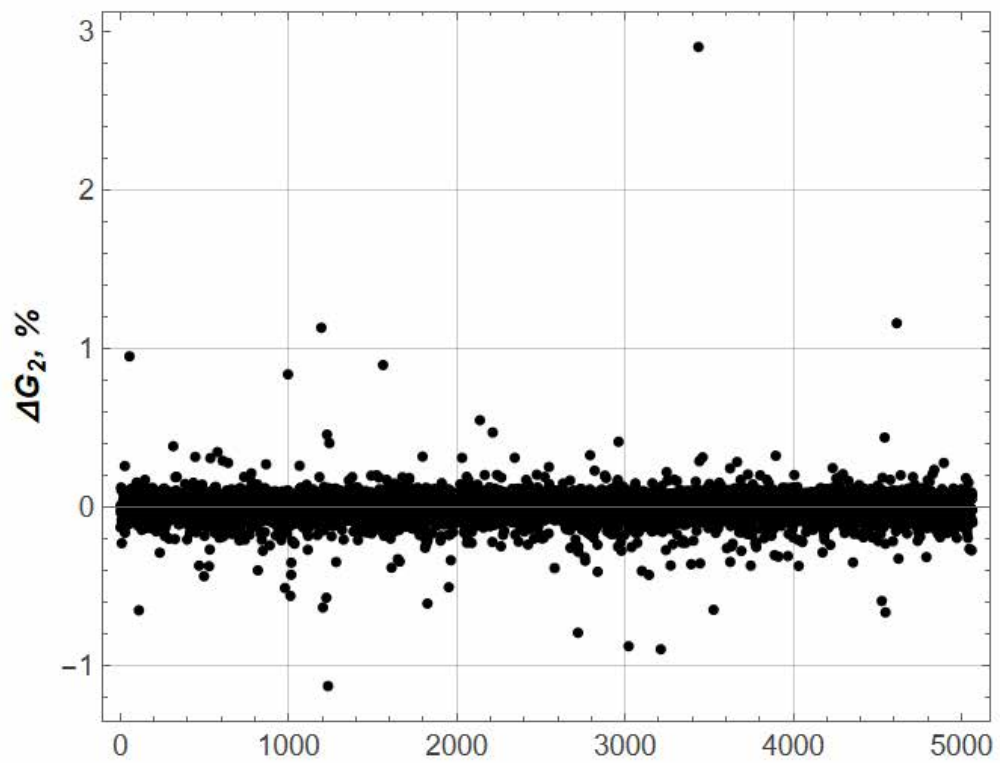


Рисунок 2.34 – Графік похибки G2

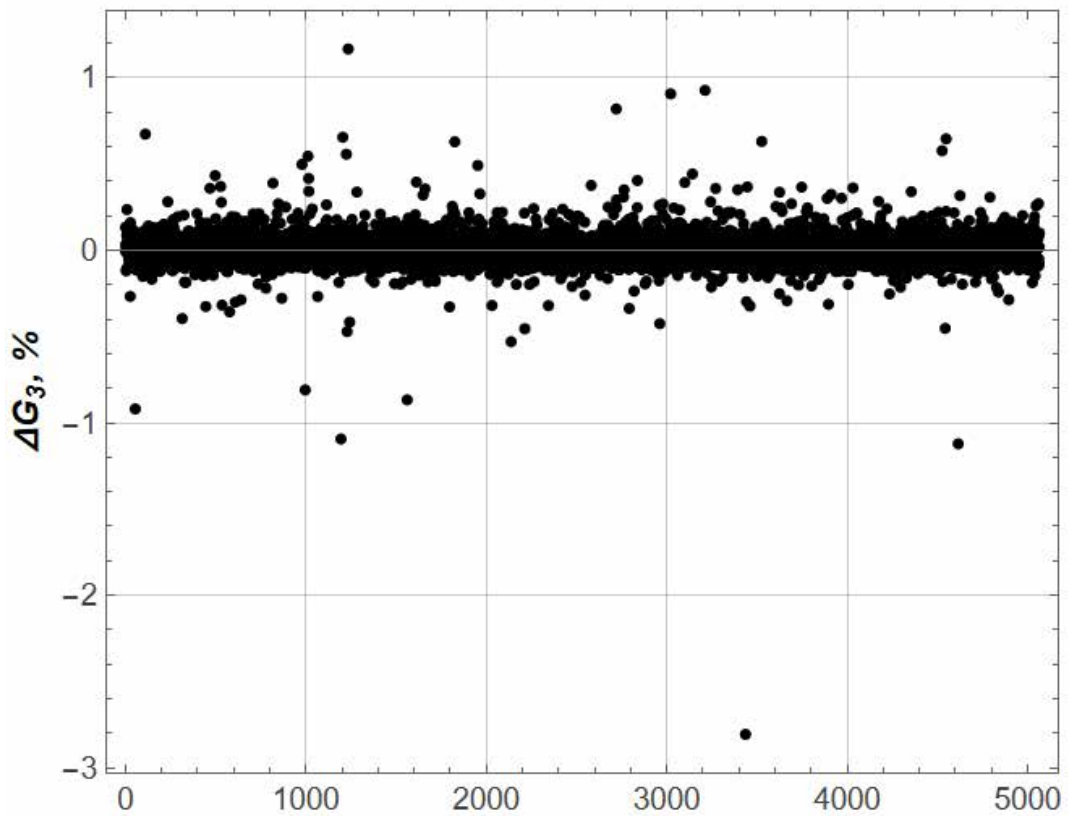


Рисунок 2.35 – Графік похибки G3

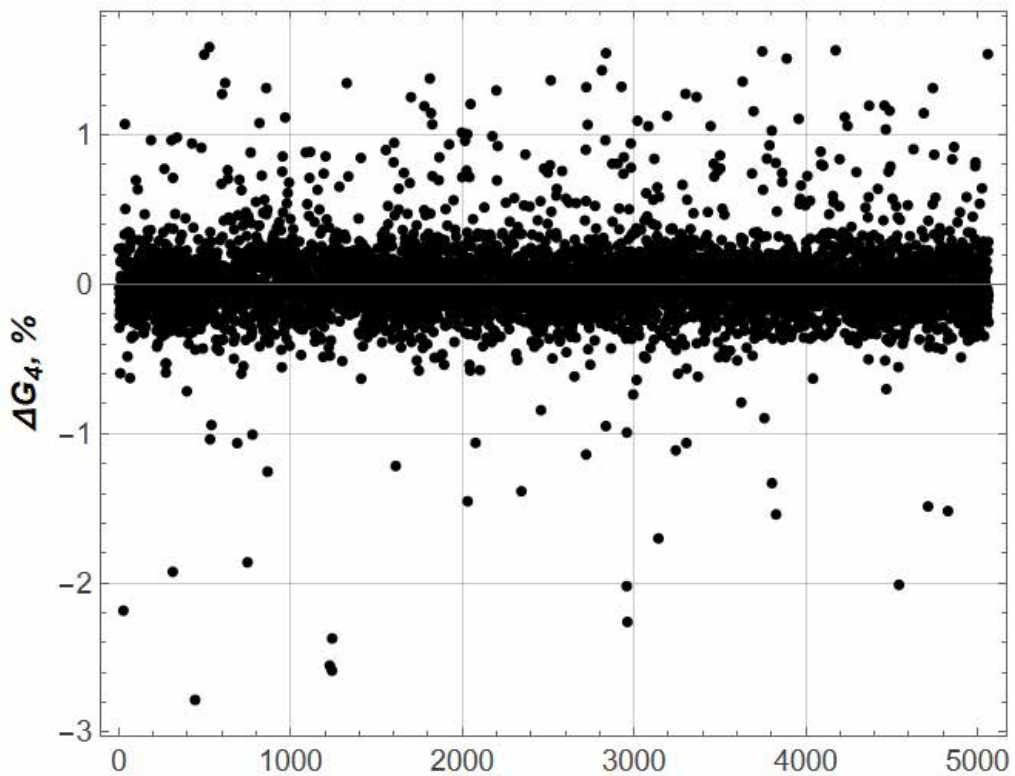


Рисунок 2.36 – Графік похибки G4

Як видно з графіків, що представлені на Рисунку 2.33 – 2.36 похибка не виходить за діапазон від -1 до 1, та від -2 до 2, окремо є поодинокі випадки похибки за діапазоном -2 до 2.

Чисельна оцінка представлена на Рисунку 2.37 та Рисунку 2.38.

Оцінка результатів тренування ШНМ на тестовій вибірці – RMS оцінки

$e_{G_1}, \%$	0.103746
$e_{G_2}, \%$	0.103747
$e_{G_3}, \%$	0.103251
$e_{G_4}, \%$	0.271207

Рисунок 2.37 – Оцінка результатів тренування ШНМ, RMS оцінки

Оцінка результатів тренування ШНМ на тестовій вибірці – максимальні оцінки

$e_{G_1}, \%$	2.89933
$e_{G_2}, \%$	2.89934
$e_{G_3}, \%$	2.80663
$e_{G_4}, \%$	1.58747

Рисунок 2.38 – Оцінка результатів тренування ШНМ, максимальні оцінки
Середньо квадратична похибка на чисельному аналізі складає $G1 - 0.103746\%$,
 $G2 - 0.103747\%$, $G3 - 0.103251\%$ та $G4 - 0.271207\%$

Та максимальна оцінка похибки складає $G1 - 2.89933\%$, $G2 - 2.89934$, $G3 - 2.80663$ та $G4 - 1.58747\%$.

Ми натренували нейронну мережу. Ми її можемо використовувати для того, щоб вона нам робила прогноз коефіцієнтів регулятора, а цей регулятор буде чудово керувати нашою установкою. Наприклад, гіробордом. Стає доросла людина, він вимірює вагу, визначає l , система ідентифікує людину. Потім подає на нейронну мережу ці параметри, а нейронна мережа видає коефіцієнти, ці коефіцієнти прописуються в регулятор. І на цьому регулятор керує рухом нашої динамічної системи. Тобто людина нормально рухається, їде та не падає та інше. Стає дитина, це вже інша маса, інша l (висота), і все це повторюється наново. Тому ми можемо спиратись на нашу натреновану нейронну мережу.

РОЗДІЛ 3. ПОСТАНОВКА І РОЗВ'ЯЗАННЯ ЗАДАЧІ СИНТЕЗУ ОПТИМАЛЬНОГО КЕРУВАННЯ ОБЕРНЕНИМ МАЯТНИКОМ У НЕЛІНІЙНІЙ ПОСТАНОВЦІ

3.1 Постановка задачі

Наступний етап у ілюстрації ефективності роботи методу навчання нейронних мереж для задач регулювання полягає у тому, що буде досліджено динаміку регулювання (стабілізації) малоприводних систем. Такі системи є нелінійними і тому задачі синтезу оптимальних регуляторів для них представляються досить складними.

У статті [2] проведено дослідження та реалізацію автоматичного керування розгойдуванням маятника.

Перевернутий маятник, модель якого зображена на Рисунку 3.1, описується системою наступних нелінійних диференціальних рівнянь:

$$\begin{cases} (m_c + m_p)\ddot{x} + m_p l(\ddot{\varphi} \cos \varphi - \dot{\varphi}^2 \sin \varphi) = F; \\ \ddot{x} \cos \varphi + l\ddot{\varphi} + g \sin \varphi = 0, \end{cases}$$

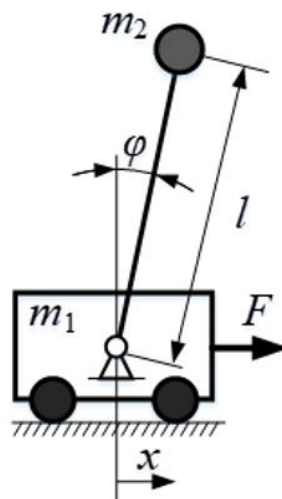


Рисунок 3.1 – Модель рухомого маятника

Тож створюємо першу функцію математичної моделі Рисунок 3.2. Задаємо масу візка, масу вантажу на стержні та довжину стержня, і подаємо у функцію поточні змінні стану і рушійне зусилля. А сама функція CartPole, повертає нам

змінні стану при дії зусилля які будуть через 0.025 секунд. Та переводимо наше диференціальне рівняння у код Рисунок 3.3

```

CartPole[ $\phi$ _,  $\omega$ _,  $x$ _,  $v$ _,  $F$ _] := (
    (*постійні значення*)
    m1 = 0.3; (*маса візка*)
    m2 = 0.2; (*маса вантажу на стержні*)
    L = 0.5; (*довжина стержня*)
    g = 9.81;
     $\Delta t$  = 0.025; (*тривалість кроку моделювання руху системи*)

```

Рисунок 3.2 – Створення функції математичної моделі

```

 $\phi$ new =  $\phi$  +  $\omega$  *  $\Delta t$ ;
 $\omega$ new =  $\omega$  + ((- (m1 + m2) * g * Sin[ $\phi$ ] - m2 * L *  $\omega$ ^2 * Sin[ $\phi$ ] * Cos[ $\phi$ ] - F * Cos[ $\phi$ ]) / (m1 * L + m2 * L * (Sin[ $\phi$ ])^2)) *  $\Delta t$ ;
 $x$ new =  $x$  +  $v$  *  $\Delta t$ ;
 $v$ new =  $v$  +  $\frac{m2 * \text{Sin}[\phi] * (L * \omega^2 + g * \text{Cos}[\phi]) + F}{m1 + m2 * (\text{Sin}[\phi])^2} * \Delta t$ ;

Return[{ $\phi$ new,  $\omega$ new,  $x$ new,  $v$ new}];

```

Рисунок 3.3 – Переведення диференціального рівняння у код

Перевіряємо чи працює функція на Рисунок 3.4.

```
In[90]:= CartPole[0.2, 1, -0.6, 2, 2]
```

```
Out[90]= {0.225, 0.520277, -0.55, 2.19503}
```

Рисунок 3.4 – Перевірка працездатності функції

Як бачимо що через 0.025с наші параметри змінились наступним чином. Кутова координата змінилась з 0.2 на 0.225, кутова швидкість з 1 на 0.520277, положення візка було -0.6 стало -0.55 та швидкість візка змінилась з 2 на 2.19503. Рушійне зусилля у нас складало всього лише 2. Якщо число буде більше чи менше відповідно числа будуть змінюватись.

Тож все працює.

3.2 Розробка оператора керування ШНМ

Сучасні підходи до синтезу оптимального керування оберненим маятником із застосуванням штучних нейронних мереж представлені в [7].

Наступним кроком задаємо модель нейронної мережі на Рисунку 3.5.

```
ANN[w111_, w121_, w131_, w141_, w112_, w122_, w132_, w142_, w113_, w123_, w133_, w143_, w114_, w124_, w134_, w144_, w115_, w125_, w135_, w145_, w511_, w521_, w531_, w541_, w551_, b11_,  
b12_, b13_, b14_, b15_, b51_, ϕ_, ω_, x_, v_] :=  
  
(*Structure of the ANN*)  
HiddenLayers = 1;  
NodesInEachLayer = 5;  
NumberOfOutputs = 1;  
  
FirstLayerWeights =  $\begin{pmatrix} w111 & w121 & w131 & w141 \\ w112 & w122 & w132 & w142 \\ w113 & w123 & w133 & w143 \\ w114 & w124 & w134 & w144 \\ w115 & w125 & w135 & w145 \end{pmatrix}$ ;  
  
LastLayerWeights = {w511 w521 w531 w541 w551};  
  
FirstLayerBiases =  $\begin{pmatrix} b11 \\ b12 \\ b13 \\ b14 \\ b15 \end{pmatrix}$ ;  
  
LastLayerBiases = b51;  
  
Inputs = List[ϕ, ω, x, v];  
  
LayerOut_1 = Table[ArcTan[(FirstLayerWeights.Inputs + FirstLayerBiases)[[r]], {r, 1, NodesInEachLayer}];  
  
NetworkOut = Table[ArcTan[(LastLayerWeights.LayerOut_1 + LastLayerBiases)[[r]], {r, 1, NumberOfOutputs}];
```

Рисунок 3.5 – Задання моделі нейронної мережі

```
ANN[w111_, w121_, w131_, w141_, w112_, w122_, w132_, w142_, w113_, w123_, w133_, w143_, w114_, w124_, w134_, w144_, w115_, w125_, w135_,  
w145_, w511_, w521_, w531_, w541_, w551_, b11_, b12_, b13_, b14_, b15_, b51_, ϕ_, ω_, x_, v_] :=
```

Рисунок 3.6 – Демонстрація вигляду вагів та біосів у кодї

Ми їй даємо на навчання параметри виділені чорним(Рисунок 3.6). Це ваги та біоси, а наступні параметри це змінні стану, вони працюють як регулятор. Тобто ми подаємо змінні стану, а модель нам видає одну певну величину, Рисунок 3.7

```
Return [NetworkOut]
```

Рисунок 3.7 – Функція отримання даних

В моделі ми маємо активаційні функції ArcTan.

Далі нам потрібно мінімізувати функцію ObjectiveFunction на Рисунок 3.8

```
(... Задати цільову функцію ObjectiveFunction)
ObjectiveFunction[w111_, w121_, w131_, w141_, w112_, w122_, w132_, w142_, w113_, w123_, w133_, w143_, w114_, w124_, w134_, w144_, w115_, w125_, w135_,
w145_, w511_, w521_, w531_, w541_, w551_, b11_, b12_, b13_, b14_, b15_, b51_] := {
```

Рисунок 3.8 – Мінімізація функції ObjectiveFunction

Ідея полягає в наступному, оптимізаційний алгоритм підбирає величину під кожен параметр вагових коефіцієнтів і біосів, при яких досягається мета керування, тобто системи переходить з вертикального положення вниз у вертикальне положення вгору і при цьому досягається певний мінімум критерію оптимізації

Тобто перша вимога що маятник з початкового положення(Рисунок 3.9), перейшов і протилежне положення, щоб число π змінилось на 0.

```
Inputs = List[ $\pi$ , 0, 0, 0];
```

Рисунок 3.9 – Задання початкового положення

Друга вимога полягає в тому, щоб візок переміщувався не більше ніж 1.5 метри вліво та вправо відносно початкового положення(Рисунок 3.10).

```
xMax = 1.5;
If[Max[Abs[xList]] < xMax, PIneq = 0, PIneq = 10^4 * (Max[Abs[xList]] - xMax)];
```

Рисунок 3.10 – Штрафна функція руху візка

А третя вимога полягає в тому щоб критерій оптимізації був мінімальний(Рисунок 3.10). Критерій складається наступним чином, середньо квадратичне значення рушійного зусилля піднесене до максимального, і середнє квадратичне потужності віднесене до максимальної потужності.

$$vnom = 0.5;$$

$$Cr = 0.1 * \frac{\text{RootMeanSquare}[FList]}{Fmax} + 0.9 * \frac{\text{RootMeanSquare}[FList * vList]}{Fmax * vnom};$$

Рисунок 3.11 – Задання критерію оптимізації

Але ми не можемо задати точні кінцеві умови у вигляді [0, 0, 0, 0], бо буде мінімальна похибка, тому ми задаємо так (Рисунок 3.11)

```
(*Задання вектора V*)
V = List[0.01, 0.01, 0.005, 0.01];
(*Задання вектора X_T - бажаних кінцевих умов системи*)
XT = List[0, 0, 0, 0];
(*Задання вектора X(T) - фактичних кінцевих умов системи*)
XFinal = List[phiList[[-1]], omegaList[[-1]], xList[[-1]], vList[[-1]]];
```

Рисунок 3.12 – Задання кінцевих умов

Кутова координата і кутова швидкість похибка складає 0.01, похибка положення візка складає 0.005 та похибка швидкості 0.01

І в кінці, функція на Рисунку 3.13, нам повертає перший компонент, це штраф, якщо ми не досягаємо кінцевих умов, навіть наближено, наступний компонент це також штраф, якщо візок виходить за обмеження 1.5м вліво або вправо і останній компонент це величина критерію оптимізації.

```
Return [PGoal + PIneq + Cr]
```

Рисунок 3.13 – Функція повернення значень штрафів та критерію оптимізації

3.3 Ініціалізація тренування ШНМ

Наступним кроком ініціалізуємо тренування нашої нейронної мережі маючи різні оптимайзери такі як, LDWPSO(Рисунок 3.14), PSORingRot(Рисунок 3.15) та VCTPSO(Рисунок 3.16). Ми будемо використовувати оптимайзер VCTPSO.

```
(* Ініціалізація параметрів для функції ціни *)
LDWPSO[function_, Arguments_, DomainArgumentList_, SwarmPopulation_, Iterations_] :=
(
(*
function - is cost function;
Arguments - are all the independent variables of the cost function. It is array of arguments;
DomainArgumentList - is the search domain. It represents the borders of multivariable domain. It is the array of the borders;
InitialArgumentValues - is an initial values for each argument;
SwarmPopulation - is a number of the particles in the swarm;
Iterations - is a number of the iterations of the main cycle;
CostFunctionValues - is a array of the cost function values during the optimization procedure.
*)

(*ЗАДАННЯ ФУНКЦІЇ ЦІНИ*)
func = function;

CostFunctionValues = List[];
(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)

(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)
c2 = 1.19; c1 = c2;

(*ІНІЦІАЛІЗАЦІЯ РОЮ*)
Swarm = List[];
Do[
(*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)
ParticlePosition = Table[RandomReal[DomainArgumentList[[i]], {i, 1, Length[Arguments], 1}];
(*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)
ParticleVelocity = Table[0, {i, 1, Length[Arguments], 1}];
ParticleCost = Apply[func, ParticlePosition];
(*ВИЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ ЧАСТИНКИ
Найкращі положення спочатку відповідають початковим положенням*)
ParticleBest = ParticlePosition;
ParticleBestCost = ParticleCost;
(*k-та частинка*)
Particle = List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest, ParticleBestCost];
AppendTo[Swarm, Particle], SwarmPopulation];
```

Рисунок 3.14 – Оптимайзер LDWPSO, частина коду

```

PSORingRot[function_, Arguments_, DomainArgumentList_, SwarmPopulation_, Iterations_] :=
(
(*
function - is cost function;
Arguments - are all the independent variables of the cost function. It is array of arguments;
DomainArgumentList - is the search domain. It represents the borders of multivariable domain. It is the array of the borders;
InitialArgumentValues - is an initial values for each argument;
SwarmPopulation - is a number of the particles in the swarm;
Iterations - is a number of the iterations of the main cycle;
CostFunctionValues - is a array of the cost function values during the optimization procedure.
*)

(*ЗАДАДАННЯ ФУНКЦІЇ ЦІНИ*)
func = function;

CostFunctionValues = List[];
(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)

(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)
w = 0.72; c2 = 1.19; c1 = c2;

(*ІНІЦІАЛІЗАЦІЯ РОЮ*)
Swarm = List[];
Do[
(*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)
ParticlePosition = Table[RandomReal[DomainArgumentList[[i]], {i, 1, Length[Arguments]}, 1]];

(*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)
ParticleVelocity = Table[0, {i, 1, Length[Arguments]}, 1]];
ParticleCost = Apply[func, ParticlePosition];
(*ВІЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ ЧАСТИНКИ
Найкращі положення спочатку відповідають початковим положенням*)
ParticleBest = ParticlePosition;
ParticleBestCost = ParticleCost;
(*к-та частинка*)
Particle = List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest, ParticleBestCost];
AppendTo[Swarm, Particle], SwarmPopulation];

```

Рисунок 3.15 – Оптимайзер PSORingRot, частина коду

```

VCTPSO[function_, Arguments_, DomainArgumentList_, SwarmPopulation_, Iterations_, RC_] :=
(
(*
function - is cost function;
Arguments - are all the independent variables of the cost function. It is array of arguments;
DomainArgumentList - is the search domain. It represents the borders of multivariable domain. It is the array of the borders;
InitialArgumentValues - is an initial values for each argument;
SwarmPopulation - is a number of the particles in the swarm;
Iterations - is a number of the iterations of the main cycle;
CostFunctionValues - is a array of the cost function values during the optimization procedure.
*)

(*ЗАДАДАННЯ ФУНКЦІЇ ЦІНИ*)
func = function;

CostFunctionValues = List[];
(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)

(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)
w = 0.72; c2 = 1.19; c1 = c2;

(*ІНІЦІАЛІЗАЦІЯ РОЮ*)
Swarm = List[];
Do[
(*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)
ParticlePosition = Table[RandomReal[DomainArgumentList[[i]], {i, 1, Length[Arguments]}, 1]];

(*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)
ParticleVelocity = Table[0, {i, 1, Length[Arguments]}, 1]];
ParticleCost = Apply[func, ParticlePosition];
(*ВІЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ ЧАСТИНКИ
Найкращі положення спочатку відповідають початковим положенням*)
ParticleBest = ParticlePosition;
ParticleBestCost = ParticleCost;
(*к-та частинка*)
Particle = List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest, ParticleBestCost];
AppendTo[Swarm, Particle], SwarmPopulation];

```

Рисунок 3.16 – Оптимайзер VCTPSO, частина коду

Далі щоб почати нам потрібно задати область зміни параметрів ШНМ, задаємо в діапазоні від -2 до 2(Рисунок 3.17). Кількість параметрів у нас складає відповідно 31.

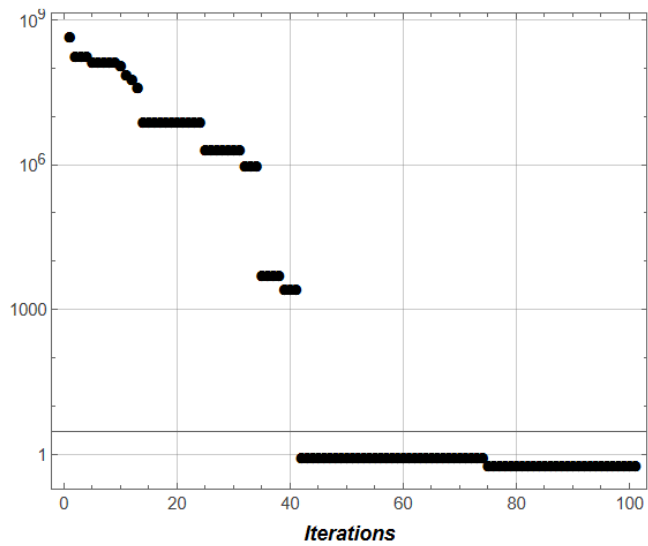


Рисунок 3.20 – Графік зміни функції ObjectiveFunction

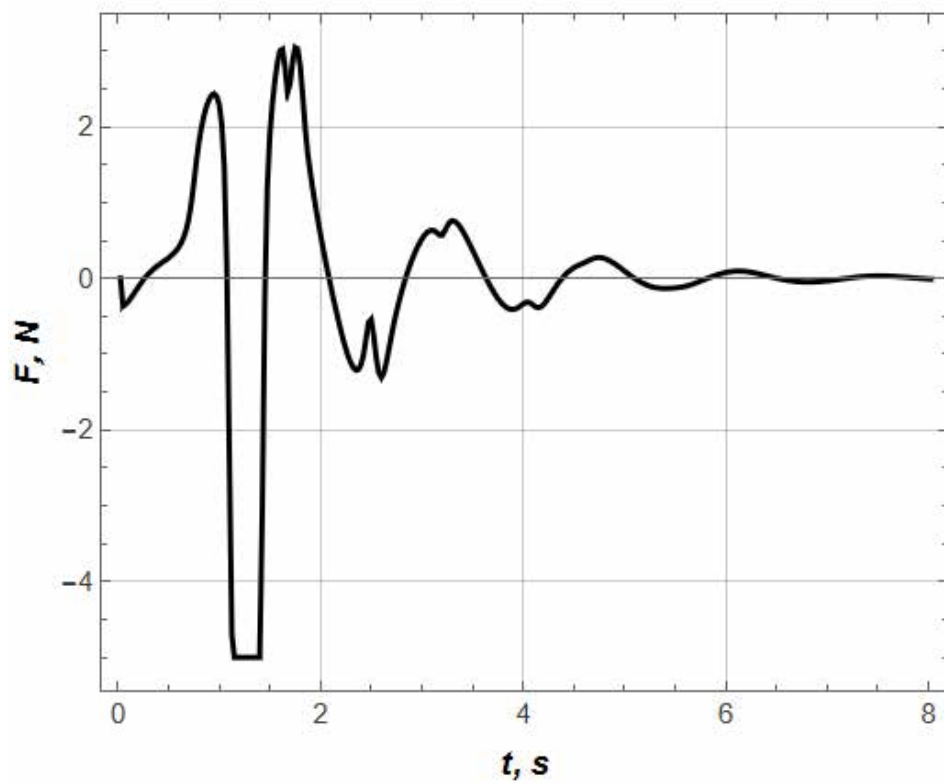


Рисунок 3.21 – Графік зміни рушійного зусилля

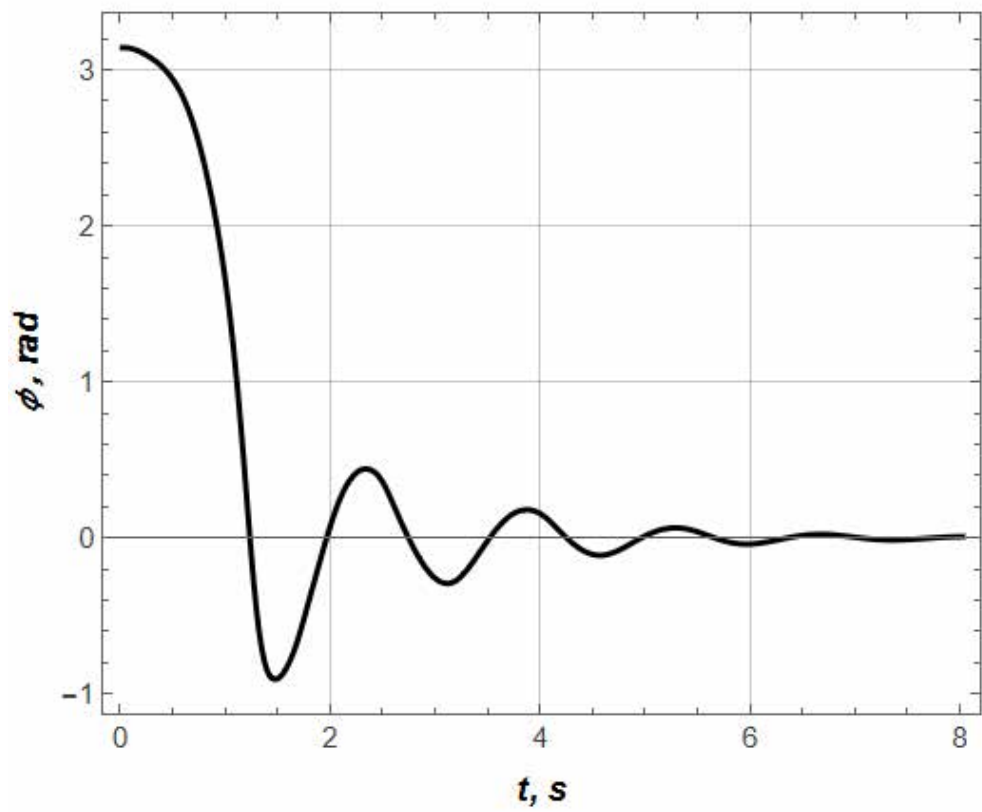


Рисунок 3.22 – Графік зміни кутової координати

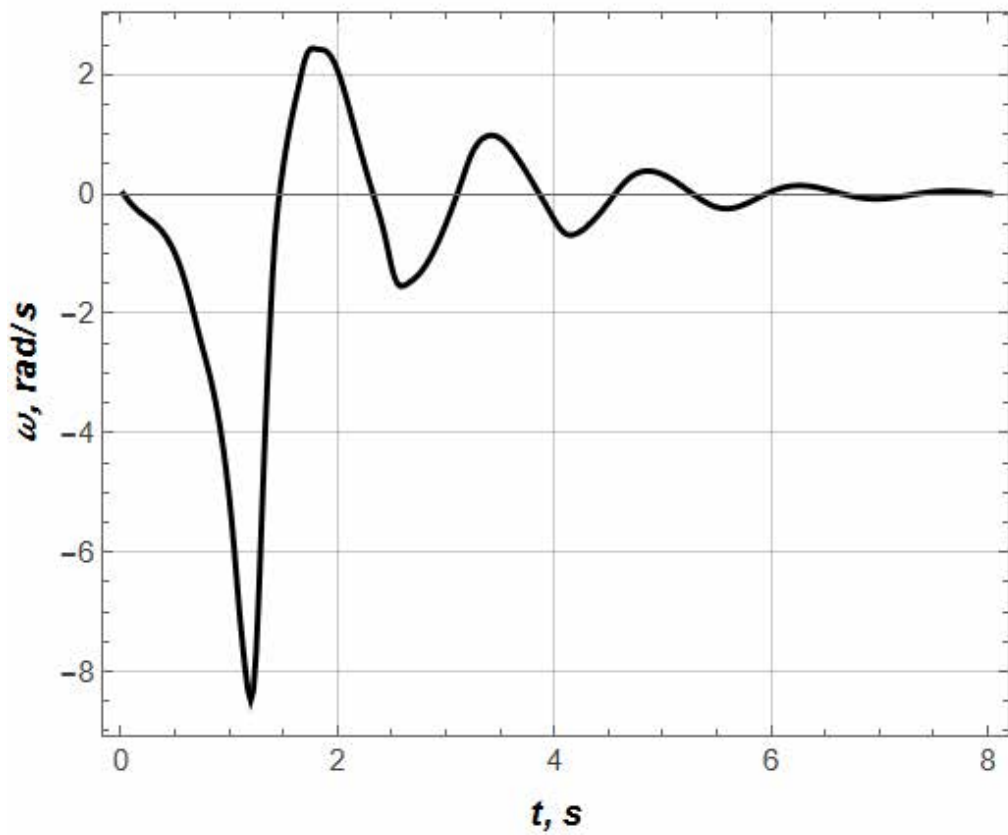


Рисунок 3.23 – Графік зміни кутової швидкості

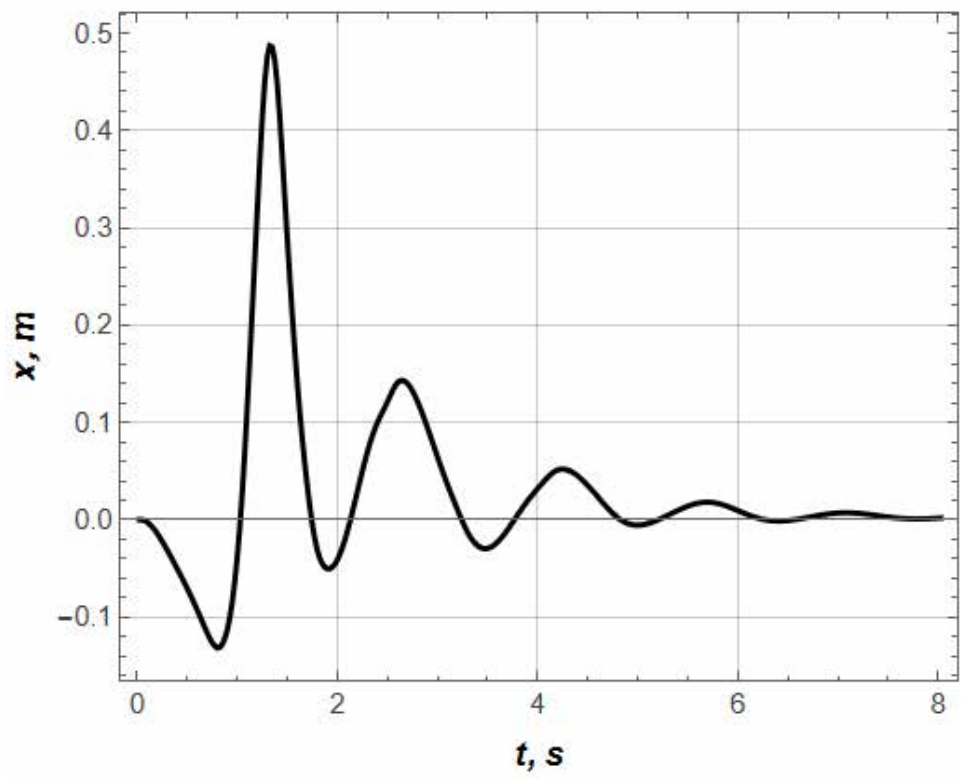


Рисунок 3.24 – Графік зміни положення

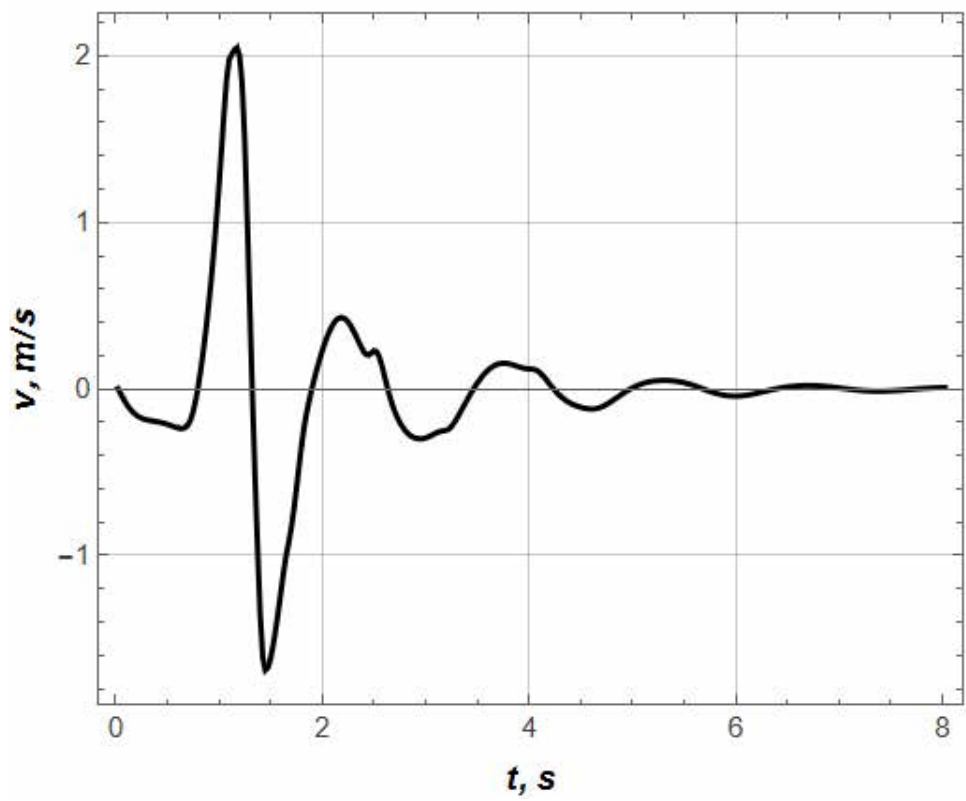


Рисунок 3.25 – Графік зміни швидкості візка.

Як видно з графіків, а саме з графіку зміни кутової координати(Рисунок 3.22). Ми починаємо рух з числа π , тобто з положення вниз, далі рухом візок розгойдується та викидається вгору.

І в кінці розрахунків маємо такі цифри(Рисунок 3.26).

```
Величина штрафної функції  $P_{goal} - 0$   
Величина штрафної функції  $P_{ineq} - 0$   
Величина комплексного критерію  $Cr - 0.607759$ 
```

Рисунок 3.26 – Величини штрафів та критерію

3.5 Підсумок тренувань

Нелінійна модель оберненого маятника продемонструвала здатність адекватно відображати складну динаміку системи та дозволила сформувати оператор керування на основі ШНМ, що забезпечує переведення системи у стійкий режим.

Отримані результати засвідчили, що нейронна мережа здатна адаптуватися до різних параметрів та початкових умов, мінімізуючи витрати енергії та дотримуючись фізичних обмежень.

Узагальнюючи проведені дослідження, можна стверджувати, що використання ШНМ у задачах керування нелінійними системами є ефективним та перспективним напрямом, який забезпечує стабільність і гнучкість у складних сценаріях роботи.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Загальні положення

Охорона праці є ключовою частиною безпечної організації виробничих процесів. Під час роботи з обладнанням, що містить рухомий обернений маятник із рухомою точкою підвісу, необхідно дотримуватися спеціальних заходів безпеки, щоб уникнути травматизму та забезпечити здорові умови праці. Усі роботи повинні виконуватися відповідно до затверджених регламентів та інструкцій, де визначені вимоги до безпечної експлуатації обладнання та дії в разі аварії.

Планування робіт має включати попереднє виявлення можливих небезпек, визначення ризикових зон та розробку заходів для їх усунення або мінімізації. До роботи допускаються лише працівники, які пройшли навчання, мають підтверджену кваліфікацію та регулярно проходять повторні інструктажі.

Все обладнання повинно бути справним і відповідати технічним стандартам. Регулярні огляди й технічне обслуговування маятника проводяться за встановленим графіком. На обладнанні повинні бути наявні захисні елементи: обмежувачі руху, аварійні вимикачі, огорожі. Система контролю має забезпечувати моніторинг стану маятника, його положення та швидкості, а всі відхилення — фіксуватися й негайно передаватися відповідальним особам.

Робоче місце повинно бути обладнане з урахуванням безпеки: позначені небезпечні зони, встановлені знаки заборони, забезпечене достатнє освітлення та вентиляція, а також доступні евакуаційні маршрути.

Працівники повинні бути забезпечені засобами індивідуального захисту (каски, окуляри, рукавички, спецвзуття, спецодяг) та вміти їх правильно використовувати.

Перед роботою слід оглянути обладнання, перевірити ЗІЗ та інструменти, провести інструктаж. Під час роботи необхідно суворо дотримуватися

встановлених процедур, уникати різких рухів і перебування в небезпечних зонах. Після завершення — оглянути обладнання, прибрати робоче місце та утилізувати відходи згідно з нормами.

На робочому місці мають бути засоби пожежогашіння, працівники повинні знати правила їх використання. Регулярно проводяться профілактичні заходи, перевірки електрообладнання та контролюється правильне зберігання легкозаймистих матеріалів. Робота з електричним обладнанням проводиться лише після його відключення.

Аптечки першої допомоги повинні бути укомплектовані, а працівники — навчати надавати допомогу та проходити періодичні тренування.

Керівники відповідають за створення безпечних умов праці, проведення аудитів та контроль за дотриманням вимог охорони праці. Працівники зобов'язані виконувати інструкції та повідомляти про виявлені порушення.

Дотримання цих положень гарантує безпечну роботу з оберненим маятником і мінімізує ризики.

В Україні охорона праці регулюється законом «Про охорону праці», КЗпП (ст. 153–174), Законом «Про забезпечення санітарного та епідемічного благополуччя населення» та ДСТУ, які визначають вимоги до умов праці, безпеки обладнання та відповідальність роботодавця.

4.2 Вимоги безпеки перед початком роботи

Перед початком роботи необхідно провести оцінку можливих ризиків, визначити небезпечні ситуації та способи їх запобігання. Працівники повинні пройти підготовку щодо роботи з оберненим маятником, знати правила безпеки та порядок дій у разі аварії.

Потрібно забезпечити працівників захисним обладнанням (окуляри, рукавички, каски) та перевірити його справність. Обладнання має бути попередньо оглянуте на наявність дефектів; усі несправності усуваються перед початком роботи.

Також слід мати готовий план дій на випадок аварії: шляхи евакуації, порядок виклику екстреної допомоги, наявність засобів першої допомоги.

4.3 Вимоги безпеки під час виконання роботи

Під час роботи необхідно контролювати правильність використання обладнання та дотримання працівниками інструкцій. Персонал повинен вміти керувати рухом маятника й точки підвісу, а також стежити за тим, щоб обладнання не створювало загрози.

Працівники повинні використовувати ЗІЗ (окуляри, рукавички, каски). Перед роботою необхідно переконатися у стабільності та справності маятника. Заборонено працювати в зонах з обмеженою видимістю або значною кількістю перешкод.

У разі появи нестандартних ситуацій або аномального руху маятника роботу негайно припиняють і усувають проблему.

4.4 Вимоги безпеки в аварійних ситуаціях

У разі аварії роботу необхідно негайно зупинити, зупинити рух обладнання та відключити електроживлення. Далі — викликати екстрені служби та надати їм повну інформацію про подію.

Працівників слід евакуювати безпечними шляхами. Якщо це безпечно, потерпілим надають першу допомогу. Джерело небезпеки необхідно ізолювати (відключити живлення, перекрити доступ).

Місце аварії слід залишити недоторканим до прибуття спеціалістів. Після ліквідації наслідків проводять розслідування причин події.

РОЗДІЛ 5. ЕКОНОМІЧНА ДОЦІЛЬНІСТЬ РОЗРОБКИ ТА СФЕРИ ПРАКТИЧНОГО ЗАСТОСУВАННЯ

5.1 Економічна ефективність розробки

Розробка оптимальних систем керування для обернених маятникових систем має значну економічну цінність завдяки:

- Зниженню енергоспоживання. Оптимальні регулятори та ШНМ дозволяють мінімізувати витрати енергії у системах транспортного типу — гіроборди, сигвеї, мобільні платформи.
- Зменшенню зносу компонентів. Стабільність керування та плавність траєкторій знижує навантаження на двигуни, приводи та механічні з'єднання.
- Оптимізації виробничих витрат. Нейромережевий підхід може замінити складні оптимізаційні алгоритми, що зменшує час розробки та спрощує налаштування обладнання.
- Збільшенню конкурентоспроможності продукту. Сучасні системи керування є ключовою частиною робототехніки, і якісне керування напряду підвищує вартість і функціональність продукту

5.2 Сфери використання розробленої системи

Розглянута система керування та сам принцип регулювання на основі моделі оберненого маятника може використовуватися в:

- Персональному електротранспорті (гіроборди, електроскутери, сигвеї);
- Робототехніці — автономні платформи, роботи-гуманоїди, балансувальні маніпулятори;
- Авіації та дронобудуванні — стабілізаційні системи;
- Навчальному обладнанні — лабораторні стенди для студентів і дослідників;
- Медичних та реабілітаційних системах — екзоскелети, балансувальні тренажери.

ВИСНОВКИ

На основі проведеного дослідження в магістерській кваліфікаційній роботі було отримано такі ключові висновки:

1. Проведено комплексний кількісний та змістовний аналіз науково-технічних публікацій, який підтвердив високу актуальність тематики синтезу оптимального керування оберненим маятником. Загальна кількість публікацій, знайдених у базах Scopus та Web of Science за ключовими запитами, перевищує 16 000 документів, що свідчить про стійкий інтерес до цієї теми.
2. Для лінійної моделі оберненого маятника успішно синтезовано LQR-регулятор, на основі якого сформовано навчальну вибірку з 50 000 розв'язків. Ця вибірка була розділена у пропорції 90/10 на навчальну (44 934 пар) та тестову (5066 пар).
3. Розроблена Штучна Нейронна Мережа (ШНМ) для керування лінійною моделлю має архітектуру з двома прихованими шарами, кожен з яких містить по 5 нейронів. Мережа приймає 2 вхідних параметри (довжина та маса маятника) і видає 4 керуючі коефіцієнти. Тренування ШНМ тривало 2290 секунд (близько 38 хвилин) і досягло високої точності: похибка на тренувальній вибірці склала 2.74464×10^{-6} , а на валідаційній — 2.90286×10^{-6} .
4. Для нелінійної моделі системи оберненого маятника розроблено оператор керування на основі ШНМ, оптимізація якого відбувалася за допомогою алгоритму рою частинок (VCTPSO). Модель продемонструвала здатність переводити маятник із нижнього положення (π) у стійке вертикальне, дотримуючись фізичного обмеження на переміщення візка не більше 1.5 метри відносно початкового положення. Типова тривалість керування (гальмування/стабілізація) у тестових сценаріях становила близько 40 секунд.
5. Нейромережевий підхід до синтезу оптимального керування є ефективним і адаптивним інструментом, що дозволяє досягти заданих

кінцевих умов (наприклад, похибка 0.01 для кутової координати) при мінімізації витрат енергії.

6. Розроблена система керування має значну економічну доцільність завдяки зниженню енергоспоживання, зменшенню зносу механічних компонентів та оптимізації виробничих витрат за рахунок спрощення алгоритмів налаштування.
7. Результати роботи мають високу практичну значущість і можуть бути безпосередньо використані при розробці систем стабілізації у персональному електротранспорті (гіроборди, сігвеї), робототехніці (платформи, маніпулятори), авіації (дрони) та у навчальному обладнанні

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. G Technology Ltd., Experiment of Inverted Pendulum and Automatic Control Principle, G Technology Ltd., Shenzhen, China, 2005.
2. C. Hui Feng, "Research on automatic swing up control of inverted pendulum system," M.S. thesis, Dalian University of Technology, China, 2009.
3. S. Kim and S. Kwon, "Nonlinear optimal control design for underactuated two-wheeled inverted pendulum mobile platform," IEEE/ASME Trans. Mechatronics, 2017.
4. V. Santibanez, R. Kelly, and J. Sandoval, "Control of the inertia wheel pendulum by bounded torques," in: Proc. 44th IEEE Conf. on Decision and Control and European Control Conf. (2005)
5. D. Somwanshi, M. Srivastava, and R. Panch. Control of Inverted Pendulum using LQR Controller. In Proceedings of the 2016 International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN).
6. V. S. Loveikin, Yu. A. Romasevich, S. A. Khoroshun, and A. G. Shevchuck, "Time-optimal control of a simple pendulum with a movable pivot. Part 1," Int. Appl. Mech., 54, No. 3, 358–365 (2018).
7. <https://www.scopus.com/record/display.uri?eid=2-s2.0-85104831246&origin=resultslist&sort=plf-f&src=s&sid=b41c6851ecc4b0c80ebd8d8ad49805bf&sot=b&sdt=cl&s=TITLE-ABS-KEY%28Inverted+pendulum+optimal+control%29&sl=48&sessionSearchId=b41c6851ecc4b0c80ebd8d8ad49805bf&relpos=9>
8. M. Hasan, C. Saha, M. M. Rahman, M. R. I. Sarker, and S. K. Aditya, "Balancing of an inverted pendulum using PD controller," Dhaka University Journal of Science, vol. 60, pp. 115-120, 2012.