

ШИФР: «Інтелектуальне керування»

НАУКОВО-ДОСЛІДНА РОБОТА

на тему: «**МЕТОДИ ТА ЗАСОБИ МОДЕЛЮВАННЯ ОБРАЗУ ЗАДАЧІ  
КЕРУВАННЯ**»

2023

## АНОТАЦІЯ

Обсяг роботи складає 55 сторінок, 27 рисунків, 2 таблиці, 15 використаних джерел, 3 додатки.

**Актуальність.** Прийняття рішень у проблемно-орієнтованих інформаційних системах та системах керування здійснюється в умовах апріорної невизначеності, обумовленої неточністю або неповнотою вхідних даних, стохастичною природою зовнішніх впливів, відсутністю адекватної математичної моделі функціонування, нечіткістю мети, людським фактором та ін. На жаль, існуючі засоби не завжди можуть якісно вирішити завдання керування, алгоритмічно відрізняються від традиційних завдань. Ці складнощі вимагають нових ідей, більш досконалих методів та інструментів, які можуть в реальному часі допомагати приймати рішення. Тому є доцільним створення системи методів та засобів моделювання образу задачі в керуючих процесах.

**Мета роботи** – довести можливість або неможливість реалізації методами і засобами ШІ предметної дії людини-оператора в задачах керування.

Для досягнення поставленої мети було виділено наступні **задачі дослідження**: аналіз предметної дії оператора; підбір відповідників структурних складових та інформаційних процесів предметної дії; моделювання штучного інтелекту; організація взаємодії між структурними складовими інформаційних процесів; перевірка адекватності отриманого образу.

У ході проведення дослідження проаналізовано можливості математичних моделей для вирішення задач керування. На основі проведеного аналізу було визначено, що можливості математичних моделей є точними тільки в заданих умовах на відміну від людини, яка повно враховує внутрішні властивості та зовнішні впливи в умовах невизначеності для прийняття рішень.

Розроблено програмне забезпечення у вигляді m. файлів в середовищі Matlab для моделювання образу задачі керування з використанням напрацьованих психологів, як основа для моделювання предметної дії засобами штучного

інтелекту. Під час розробки було використана рекурсивна багатошарова нейронна мережа з методом зворотного поширення помилки.

В якості тестування та проведення експериментальних досліджень було обрано задачу удару по м'ячу. Використовуючи нечітку логіку та нейронну мережу аналізується стан футболіста та пропонується інструкція для досягнення образу цілі. Адекватність отриманих результатів перевіряється шляхом співставлення інструкції програми та інструкції футбольного тренера. В результаті проведення досліджень розроблених моделей виявлено, що програма видає точні результати, тобто доведено реалізацію методами і засобами ШІ предметної дії людини-оператора в задачах керування.

**Ключові слова:** образ задач, задачі керування, Matlab, нейронна мережа, ШІ.

## ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Аналіз проблем керування в ергатичних системах.....	8
1.2 Організація предметної дії людини-оператора.....	10
1.3 Організація інформаційних процесів в предметній дії.....	10
1.4 Аналіз та синтез образів цілі та задачі .....	13
1.5 Невизначеність задачі, її ієрархізація, редукція та конкретизація .....	14
1.6 Мета та завдання роботи.....	15
РОЗДІЛ 2 АНАЛІЗ І СИНТЕЗ МОДЕЛЕЙ ТА АЛГОРИТМІВ .....	16
2.1 Вибір програмного комплексу .....	16
2.2 Проектування системи вирішення задачі на основі формування образу .....	16
2.3 Порівняльні властивості спроектованих моделей.....	18
2.4 Алгоритм навчання нейронної мережі методом зворотного поширення помилки .....	19
2.5 Моделювання навколишнього простору об'єкта керування.....	21
2.6 Висновки по розділу 2.....	22
РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ РОЗРОБЛЕНОГО МЕТОДУ МОДЕЛЮВАННЯ ОБРАЗУ ЗАДАЧІ КЕРУВАННЯ .....	23
3.1 Розробка структурних складових інформаційних процесів предметної дії у футболіста .....	23
3.2 Розробка методики і планування експериментального дослідження..	24
3.3 Проведення експериментальних дослідження адекватності розроблених моделей .....	24
3.4 Аналіз адекватності отриманих результатів .....	25
3.5 Висновки по розділу 3.....	27
ВИСНОВКИ .....	28
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	29
ДОДАТОК А Matlab скрипт .....	31

ДОДАТОК Б Проектування системи вирішення задачі шляхом математичної моделі.....	44
ДОДАТОК В Структура моделей у Matlab.....	52

## ВСТУП

**Актуальність теми.** Прийняття рішень у проблемно-орієнтованих інформаційних системах та системах керування здійснюється в умовах апріорної невизначеності, обумовленої неточністю або неповнотою вхідних даних, стохастичною природою зовнішніх впливів, відсутністю адекватної математичної моделі функціонування, нечіткістю мети, людським фактором та ін. Невизначеність системи призводить до зростання ризиків від прийняття неефективних рішень, результатом чого можуть бути негативні економічні, технічні та соціальні наслідки.

В ситуаціях, коли обчислювальні можливості людини недостатні для обробки інформації, необхідної для прийняття рішень, потрібні сучасні інструментальні засоби і методи. Як впливає з аналізу операторської діяльності і досліджень, такі оцінки здійснюються в такий спосіб: виникле уявлення про завдання (його образ) оператор співвідносить з інформацією, що актуалізована з пам'яті, про те, наскільки успішно і надійно вдавалося йому вирішувати подібні завдання в минулому. Попутно актуалізуються правила та інші обмеження, що накладаються на рішення таких задач, а також побічні результати минулих рішень (наприклад, отримані при цьому вигоди або, навпаки, аварії та інші неприємності). Швидкість же реагування впливає з поставленої мети, ситуації, що склалася, умовної задачі. В ситуаціях які вимагають швидких відповідних дій, найбільш вагомими елементами опиняються не швидкість моторної реакції, а повнота і швидкість оцінки наявного стану речей, тобто повнота і швидкість аферентного синтезу. Головною причиною помилкових дій операторів в подібних ситуаціях є поспішність з відповідними реакціями.

На жаль, існуючі засоби не завжди можуть якісно вирішити завдання керування, алгоритмічно відрізняються від традиційних завдань. Ці складнощі вимагають нових ідей, більш досконалих методів та інструментів, які можуть в реальному часі допомагати приймати рішення. Таким інструментом можуть бути засоби інтелектуального моделювання(складних алгоритмів на основі

нейронних мереж, нечіткої логіки), розробка яких для систем управління є проблемою.

Тому є доцільним створення системи методів та засобів моделювання образу задачі в керуючих процесах.

**Практична проблема:** вирішення задач керування в умовах невизначеності та не стаціонарності. **Невирішена наукова задача:** моделювання предметних дій оператора, методами штучного інтелекту.

**Мета дослідження:** підвищення ефективності тренування операторів за рахунок розроблення методів та засобів моделювання образів задач керування.

**Об'єкт дослідження:** процеси керування людиною-оператором рухомими об'єктами. **Предмет дослідження:** методи та засоби моделювання операторської діяльності людини на основі методів штучного інтелекту.

**Задачами дослідження** відповідно до поставленої мети є:

1. Аналіз предметної дії оператора з позиції інженерної психології.
2. Підбір відповідників структурних складових та інформаційних процесів предметної дії в методах і засобах штучного інтелекту.
3. Моделювання структурних складових інформаційних процесів предметної дії.
4. Організація взаємодії між структурними складовими інформаційних процесів.
5. Формування образу задачі.
6. Перевірка адекватності отриманого образу.

**Наукова новизна дослідження:** вперше створена імітаційна модель предметної дії оператора на основі інженерно-психологічних уявлень шляхом застосування методів та засобів штучного інтелекту.

**Практична цінність дослідження:** розроблені моделі та методи можуть бути використані в наукових дослідженнях шляхом масштабування, накопичення даних та використаний в області робототехніки, тренажерній підготовці операторів для вирішення задач тренування.

Автор висловлює подяку Гнідашу Віталію за допомогу у написанні роботи.

# РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Аналіз проблем керування в ергатичних системах

Ергатична система (система людина-машина) – система, складовим елементом якої є людина-оператор або кілька людей-операторів. У загальному випадку ергатичні системи – це складні ієрархічні системи керування, в яких людина може брати участь на будь-якому рівні. Прикладом ергатичної системи можуть бути: керування засобом транспорту, екскаватором, диспетчерська служба шахт, збагачувальних фабрик тощо. Дослідження ергатичної системи проводиться на основі системного підходу, що розглядає об'єкт як систему, в якій виділяються окремі елементи, внутрішні та зовнішні зв'язки, що впливають на функціонування елементів, при цьому цілі кожного з елементів формуються залежно від загального призначення системи. Підвищення ефективності ергатичних систем пов'язано зі зростанням їх технічної та програмної складності. У зв'язку із цим посилюються вимоги до працездатності (надійності) людини-оператора та надійності компонентів, що входять до складу ергатичної системи. Для розв'язання цієї проблеми потрібно вміти оцінювати рівні надійності всіх компонентів та їх внесок у рівень надійності всієї системи. Нині питанням проблеми працездатності складних людино-машинних систем із сучасним програмним забезпеченням приділяється недостатньо уваги, немає єдиного концептуального підходу до дослідження таких систем. Також недостатньо вивчено проблеми оператора і працездатність програмних комплексів.

Основною проблемою керування в ергатичних системах є формалізація психологічних аспектів розумової діяльності людини в процесі вироблення рішень. Проблема формалізації основних схем поведінки та характеристик людини-оператора пов'язана зі спробою створення математичних моделей діяльності людини, але описати їх за допомогою рівнянь можна лише наближено. Це обумовлено тим, людина не мислить рівняннями, вона сприймає образ.



Мислення – це активний процес відображення об’єктивного світу в людському мозку у формі суджень, понять і висновків. Розрізняють наочно-образне, словесно-логічне і оперативне мислення. Для діяльності оператора характерний особливий тип мислення – оперативне мислення. Оперативне мислення – це такий шлях вирішення практичних завдань, який здійснюється на основі моделювання оператором об’єктів трудової діяльності, в результаті чого формується суб’єктивна модель передбачуваної сукупності дій, що забезпечують вирішення поставленого завдання.

Виділяють три основні функції оперативного мислення: рішення задач, планування, декодування.

1. Рішення задач – це функція, що реалізується за різних і непередбачуваних змін у процесі управління, коли відомі засоби регулювання не спрацьовують. Типова форма її прояву – це прийняття рішення. Ця функція оперативного мислення вважається основною і детальніше розглядатиметься далі;

2. Планування зумовлене необхідністю впорядкування та організації дій оператора з управління процесом у нормальних умовах. Результатом планування є уявлення про послідовність дій, необхідних засобів і часу на їх виконання. Для цього операторові важливо знати закономірності перебігу процесу управління, передбачати послідовність змін стану системи, враховувати ймовірність їхнього прояву. Однією з форм планування є оптимальне планування, за якого обирається оптимальний варіант регулювання системи з кількох можливих на основі їхнього зіставлення;

3. Сутність функції декодування полягає в переведенні образу сигналу в оперативний образ керованого об’єкта або процесу і пов’язана з тим, що інформація про процес управління на технічних засобах відображається у закодованому вигляді.

Оперативне мислення завжди починається з усвідомлення задачі, аналізу ситуації, в результаті чого окреслюється відоме і визначається шукане. За

змістом цей процес являє собою рух думок оператора від згорнутого формулювання виробничої задачі до детального розкриття її вимог, підзадач.

## **1.2 Організація предметної дії людини-оператора**

Перш ніж приступити до конкретного аналізу структури предметної дії, слід обумовити деякі використовувані при цьому поняття. Саму дію можна визначати як процес підпорядкованого представлення проміжної мети в проміжному результаті діяльності. У психологічній літературі, поряд з поняттям «дія», використовується близьке до нього поняття «вчинок». С. Л. Рубінштейн [1] визначає вчинок людини як «дія, яке сприймається і усвідомлюється самим діючим суб'єктом». Розглянувши твердження Рубінштейна можна зробити висновок, що дія є більш широким поняттям, а вчинок виявляється лише його різновидом. Коли керуючий вплив здійснюється усвідомлено, то цілком справедливо називати його і керуючим вчинком.

З позиції кібернетики і фізіології предметну дію аналізували П. К. Анохін [2] і Н. А. Бернштейн; Всі дослідження, незважаючи на різноманітність підходів до трактування цього питання, виходили з одного загального положення, висловленого І. М. Сеченовим про те, що «відчуття всюди має значення регулятора руху».

Інакше кажучи, головною складовою будь-якої цілеспрямованої предметної дії є перетворення того, що існує на даний момент, в те, що має стати. Це і є оцінкою того, що має бути. Саме існуючий стан речей, що ми можемо відчувати, зумовлює і автоматизовані рухи у відповідь та дії у відповідь, що є результатом вирішення розумових задач та проблемних ситуацій.

## **1.3 Організація інформаційних процесів в предметній дії**

Інформацію оператор отримує з інформаційних моделей, що вимагає її декодування, а це, своєю чергою, потребує від нього значної розумової

активності. Функція декодування полягає в переведенні образу сигналу в оперативний образ об'єкта або керованого процесу. Звідси випливає, що оперативне мислення в основному є образним. Оперативний образ виникає в результаті співвіднесення отриманої інформації з інформацією, яку зберігає в своїй пам'яті оператор.

Оперативні образи характеризуються:

1. Вони прагматичні, оскільки формуються у процесі дій з об'єктами, під час виконання конкретних практичних завдань і відповідно до них;
2. Вони адекватні конкретній меті дії і, залежно від самого завдання, умов його виконання і психологічних особливостей людини-оператора, можуть змінюватися, залишаючись адекватними самому завданню управління;
3. Вони впорядковані, тому що інформація в них структурно організована в єдиний комплекс, у якому окремі складові мають певний взаємозв'язок;
4. Вони специфічні, адже відображають тільки ту інформацію, котра необхідна для розв'язування конкретного завдання.

Слід зауважити, що адекватність образу інформаційній моделі ще не означає його адекватності завданню, що вирішується, оскільки на базі однієї інформаційної моделі можуть виникати різні концептуальні моделі, які породжують різні оперативні образи. Неадекватність оперативного образу тій ситуації, яку він відображає, може зумовлюватися різними факторами як зовнішнього, так і внутрішнього характеру.

Оперативні образи, що адекватні ситуації, лежать в основі прийняття оперативного рішення і тому є важливими регуляторами в системі предметної дії оператора. Більше того, за ускладнення умов ефективність використання оперативних образів значно зростає. Роль оперативних образів у загальній схемі предметної дії оператора може бути представлена у вигляді схеми (рис. 1.1) [3], в якій робиться спроба врахувати різні аспекти цього питання. На цій схемі у вигляді окремих блоків представлені складові процесу.

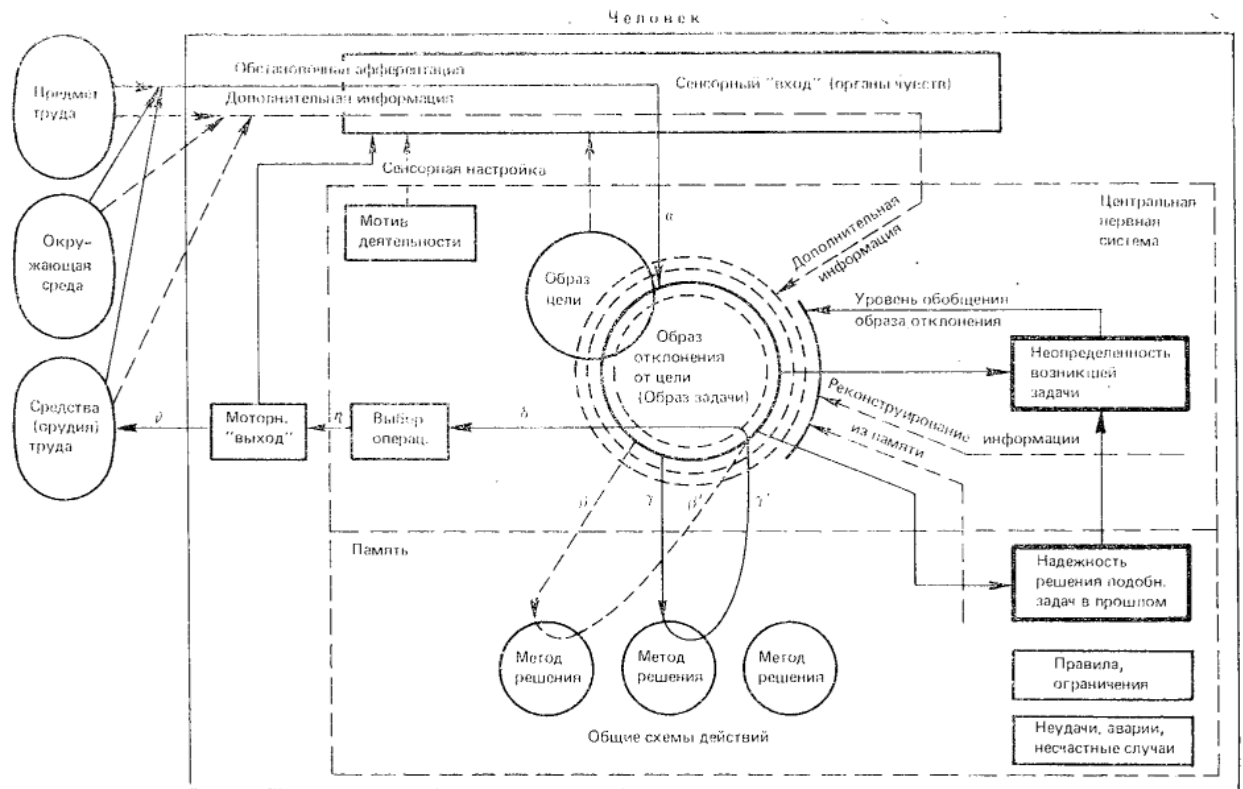


Рисунок 1.1 – Схема організації інформаційних процесів в предметній дії

Як будь-яка предметна діяльність, так і предметна дія завжди починається з усвідомлення мети - актуалізації образу бажаного результату, який виступає як «модель потрібного майбутнього». У блоці ЦНС угорі ліворуч зображене коло, який названий образ цілі. Фактором же, котрі спрямовують дію до його мети, як уже зазначалося, є загальний мотив даної діяльності.

У свідомості оператора складається оперативний образ цілі, під впливом якого формується певна оперативна готовність, котра орієнтує сенсорний вхід оператора на сприйняття тих параметрів функціонування системи, що є типовими для даного стану системи. Завдяки такому налаштуванню оператор знімає з приладів не абсолютні значення параметрів, а їхні розбіжності між заданими і фактичними. Ці розбіжності накладаються на образ-еталон і породжують новий оперативний образ – образ-відхилення, котрий відповідає новій концептуальній моделі, що виникла через зміну стану об'єкта. З цим образом-відхиленням в оператора пов'язана певна загальна схема дій, підсилена

конкретними засобами її практичної реалізації. Вибір алгоритму керуючих дій здійснюється під впливом образу-відхилення і його концептуальної моделі.

#### **1.4 Аналіз та синтез образів цілі та задачі**

Образ задачі є складним багатофакторним уявленням суб'єкта про неузгодженість між тим, що є на цей час, і тим, що повинно стати. Воно відображає його ставлення до цієї неузгодженості і шляхи до його усунення. Даний образ в предметній дії виступає як деякий центральний пункт, який об'єднує в собі все те, що потрібно знати і враховувати при пошуку рішення завдань, що виникають. Тому він повинен бути перш за все достатньо інформативним. І якщо тих відомостей, які безпосередньо сприймаються ззовні і актуалізуються з пам'яті, виявляється недостатньо, ведеться активний пошук додаткової інформації у зовнішньому середовищі, в пам'яті, з наявних відомостей реконструюються нові - і так до отримання такої інформаційної основи образу, яка буде визнана достатньою для знаходження шуканого рішення.

І тут, природно, виникає питання: на основі чого оператор робить висновок, що та чи інша інформативність образу завдання є достатньою або недостатньою для знаходження його рішення? Як впливає з аналізу операторської діяльності і досліджень цього питання, такі оцінки здійснюються в такий спосіб: виникло уявлення про завдання (його образ) оператор співвідносить з інформацією, що актуалізована з пам'яті, про те, наскільки успішно і надійно вдавалося йому вирішувати подібні завдання в минулому. Попутно актуалізуються правила та інші обмеження, що накладаються на рішення таких задач, а також побічні результати минулих рішень (наприклад, отримані при цьому вигоди або, навпаки, аварії та інші неприємності).

Швидкість же реагування впливає з поставленої мети, ситуації, що склалася, умовної задачі. І тут доречно зазначити, що в ситуації, що вимагає швидких відповідних дій, найбільш істотним елементом стають не швидкість

моторної реакції, а повнота і швидкість оцінки наявного стану речей, тобто повнота і швидкість аферентного синтезу. Досвід показує, що головною причиною помилкових дій операторів в подібних ситуаціях є поспішність з відповідними реакціями.

### **1.5 Невизначеність задачі, її ієрархізація, редукція та конкретизація**

З співвідношення образу задачі з даними актуалізованими з пам'яті, у оператора формується твердження про ступінь невизначеності завдання щодо бажаного результату. Якщо в минулому подібні задачі виникли та вирішувалися успішно, то дана задача сприймається як досить певна, яку можна вирішити на не великій інформаційній базі. Якщо ж такі завдання раніше ніколи не вирішувалися або вирішувалися невдало, то це є свідченням високої невизначеності даного завдання і вказує на необхідність розширення її інформаційної бази до досить високого рівня, на якому стане можливим вирішення цієї невизначеності. Причому, в залежності від того, наскільки невизначена виникла задача для оператора, настільки він розширює рівень узагальнення її образу. Зі сказаного випливає, що інформаційне забезпечення образу задачі залежить від того, наскільки невизначеною видається завдання оператору, а ця невизначеність, в свою чергу, оцінюється на основі його минулого досвіду, його уявлення про власні можливості.

Так, було встановлено, що, коли оператори вирішували завдання, аналогічні відомим їм за власним минулому досвіду, результати рішень були значно краще, ніж при вирішенні об'єктивно настільки ж складних, але маловідомих їм завдань.

Ієрархізація полягає в розкладанні завдання на частини, виділення її найбільш істотних елементів, встановленні зв'язку між ними, в їх ранжуванні.

Редукція спрямована на спрощення способу, виділення в ньому лише тих відомостей, які дозволяють сформулювати обмежене число найбільш реальних робочих гіпотез дозволу завдання.

Конкретизація образу полягає в використанні в ньому тільки тих даних, які представлені в явному вигляді, дані ж, які потрібно вишукувати, додатково виводити, ігноруються.

Названі вище перетворення фактично направлені на спрощення способу завдання, на те, щоб зробити його більш зручним для використання і пошуку рішення.

## **1.6 Мета та завдання роботи**

Мета даної роботи – довести можливість або неможливість реалізації методами і засобами ШІ предметної дії людини-оператора в задачах керування.

Розроблена модель має видавати адекватні отримані результати – інструкцію, щодо дії людини-оператора.

Для досягнення поставленої мети було виділено наступні задачі дослідження:

- аналіз предметної дії оператора;
- підбір відповідників структурних складових та інформаційних процесів предметної дії;
- моделювання штучного інтелекту;
- організація взаємодії між структурними складовими інформаційних процесів;
- перевірка адекватності отриманого образу.

## РОЗДІЛ 2 АНАЛІЗ І СИНТЕЗ МОДЕЛЕЙ ТА АЛГОРИТМІВ

### 2.1 Вибір програмного комплексу

На сьогоднішній день існують велика кількість програмних засобів для роботи з моделюванням, нейромережами, різноманітним обчисленням та тестуванням. Наприклад, такі, як Statistica (фірма StatSoft), NeuroSolutions (компанія NeuroDimension), MatLab (компанія MathWorks).

Вибір на користь MatLab зроблений з зв'язку з великими можливостями даного середовища. MatLab – інструмент, що забезпечує взаємодію оператора (часто навіть не програміста) з усіма доступними можливостями аналізу, збору і представлення даних. У нього є очевидні плюси – інтерактивне середовище для програмування, чисельних розрахунків і візуалізації результатів. За допомогою MatLab можна аналізувати дані, розробляти алгоритми, створювати моделі і додатки. Вбудовані математичні функції дозволяють досліджувати різні підходи і отримувати рішення швидше, ніж з використанням електронних таблиць або традиційних мов програмування, таких як C/C++ або Java.

MatLab широко використовується для вирішення широкого спектра наукових і прикладних задач, в таких областях як: моделювання об'єктів і розробка систем управління, проектування комунікаційних систем, обробка сигналів і зображень, вимірювання сигналів і тестування, фінансове моделювання, обчислювальна біологія та ін.

### 2.2 Проектування системи вирішення задачі на основі формування образу

На відміну від машини людина не може мислити числами, змінами, вона мислить образами (рис. 2.1).



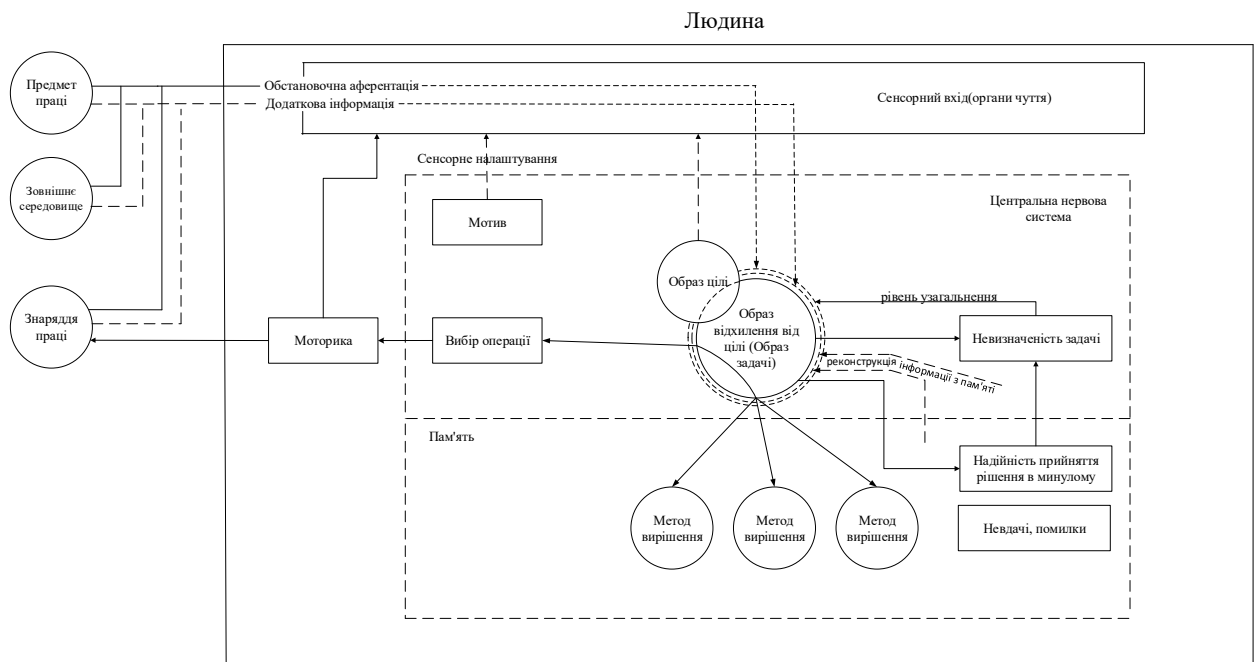


Рисунок 2.1 – Формулювання образу задачі людиною

В будь-якій дії людини є певний мотив, який формує образ цілі. Наприклад візьмемо футболіста, мотивом якого є забити гол, внаслідок чого формуються образи та дія. Для цього представимо наприклад для найнижчої класифікації футболіста такі рішення футболіста щодо параметрів удару по м'ячу табл. 2.1. Моторику виходу можна представити у вигляді набору параметрів що формуються у людини у вигляді образу, так як людина не мислить символами та числами, а мислить образами. Подальший процес навчання людини можна змодельовати в вигляді штучної нейронної мережі оскільки штучна нейрона мережа є аналогом біологічної нейронної мережі.

Таблиця 2.1 – Параметри удару

Сильно	Середнє	Слабо
Зліва	По центру	Справа
Зліва	По центру	Справа
Нижнє ліво	Нижче центру	Нижнє право

Оскільки кількість входів при п'яти параметрах и трьох варіантів вибору дорівнює 30 для лінійного вектору то вважаємо, що кількість нейронів в шарі повинна бути як мінімум в 2 рази більше, тому виходить 60, оскільки задача являється складною то кількість шарів повинна бути як мінімум 3, наприклад 5.

### 2.3 Порівняльні властивості спроектованих моделей

Побудувавши математичну модель та модель на базі образів проведемо порівняльні характеристики.

Таблиця 2.2 – Порівняльні властивості

<b>Математична модель</b>	<b>Модель на базі образу</b>
Являє собою систему рівнянь, яка описує перетворення вхідних даних на вихідні.	Являє собою набір відповідників, що зберігаються, змінена методами і засобами штучного інтелекту.
Створюється до початку керування шляхом теоретичних та експериментальних досліджень. Використовується на протязі всього часу керування.	Створюється в реальному часі безпосередньо перед використанням шляхом спостереження впливів на об'єкт керування, та постійно вдосконалюється та змінюється в процесі діяльності
Зовнішні збудження описуються окремо від об'єкта керування (як правило випадковими процесами).	Об'єкт керування та зовнішні збурення сприймаються як єдине ціле. Модель будується як детермінований процес.
Модель є точною в межах заданих припущень але не повною(враховує лише основні чинники).	Модель є повною – враховує внутрішні властивості та зовнішні впливи, але неточною (моделює дію збурень лише «в середньому»).

Порівнюючи ці дві моделі можна сказати, що математична модель є точною лише в заданих припущеннях та складною, на відміну від моделі на базі образів, яка є простою та працює в реальному часі.

## 2.4 Алгоритм навчання нейронної мережі методом зворотного поширення помилки

Розроблена нейронна мережа являє собою комп'ютерну програму, яка призначена, для прогнозування удару м'яча в задану точку, що є процесом, схожим з роботою людського мозку. Крім того, на рис. 2.3 показано, як штучний нейрон обробляє вхідні дані в вихідний сигнал. Проте, найбільш важливою особливістю нейронної мережі є те, як вона вчиться. Навчання побудовано на методі навчання алгоритму зворотного поширення помилки. Наступний висновок узятий з пояснення, представленому Дхаррм і Стеїном (1996). Всі математичні формули відносяться до рис. 2.2. Показаний один штучний нейрон, який вчиться за допомогою алгоритму навчання зворотного поширення помилки.

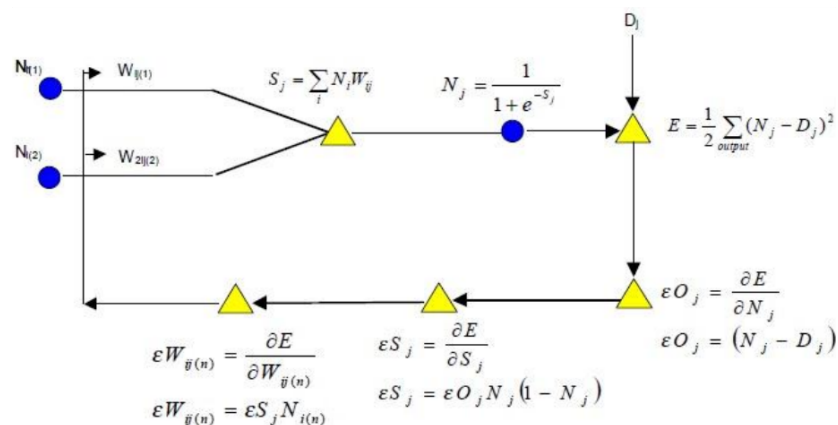


Рисунок 2.2 – Штучний нейрон. Використання зворотного поширення навчання

Алгоритм зворотного поширення прагне звести до мінімуму член помилки між входом нейронної мережі і фактичним необхідним вихідним значенням. Термін помилки обчислюється порівнянням корисної потужності до бажаного виходу, а потім подається назад через мережу в результаті чого синаптичні ваги змінюються, щоб звести до мінімуму помилку. Процес повторюється, поки

помилка не досягає мінімального значення. Мережа використовує рівняння (2.1) для оновлення ваг  $W_{ij}$  з даного вузла  $N_i$  до поточного вузла  $N_j$ ; де  $t$  відноситься до кількості і  $\lambda$  відноситься до параметру навчання. Навчання параметру або швидкість навчання, контролює швидкість виміру вагів, так проходить навчання. Чутливість вузла  $N_j$  до зміни ваг  $W_{ij}$  представлена  $(\varepsilon W_{ij})$  та буде представлена по формулі:

$$W_{ij,(t+1)} = W_{ij,t} + (\lambda)(\varepsilon W_{ij})(N_j), \quad (2.1)$$

Загальний вхід вузла описаний в рівнянні:

$$S_j = \sum_i N_i W_{ij} \quad (2.2)$$

де  $S_j$  – сума всіх входів в вузлі,

$N_i$  – вихід попереднього вузла

$W_{ij}$  – з'єднання по масі між вузлами попереднього шару.

Цей вихідний сигнал перетворюється за допомогою функції активації, рівняння:

$$N_j = \frac{1}{1+e^{-S_j}} \quad (2.3)$$

Операція триває на вузлах нижнього шару, дозволяючи вузлів нижнього прихованого шару, грати роль вихідного вузла. Всі помилки від всіх входів в прихований шар повинні бути підсумовані. Спрощене рівняння для розрахунку ваги поновлення надається в рівняння нижче, де індекс  $J$  представляє сукупність вузлів в шарі над прихованому шаром.

$$\varepsilon H_i = \sum_j \varepsilon S_i W_{ij} \quad (2.4)$$

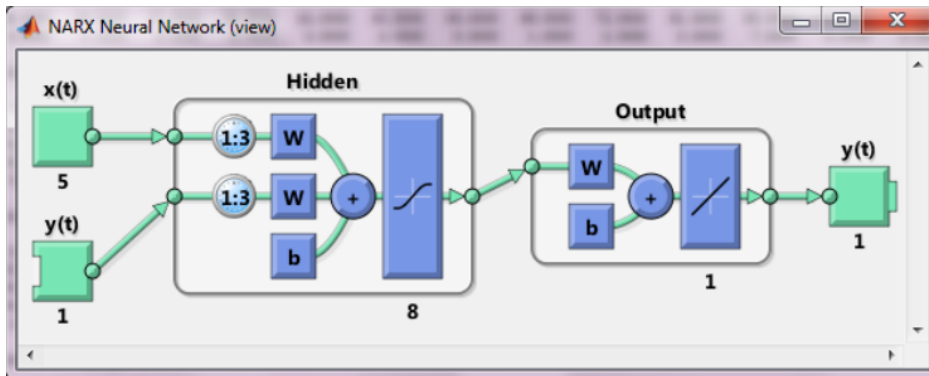


Рисунок 2.3 – Паралельний варіант архітектури NARX-мережі

Таким чином, помилка в мережі поширюється рекурсивно назад через всю мережу і всі ваги коректуються, щоб звести до мінімуму загальну похибка мережі.

## 2.5 Моделювання навколишнього простору об'єкта керування.

Опрацювання інформації, тобто сигналів, що виникають в організмі й надходять із зовні потрібно також враховувати. Для футболіста будуть характерні наступні фактори рис. 2.4.

### Зовнішнє середовище



Рисунок 2.4 – Навколишній простір під час удару

Такі фактори як погодні умови, характеристика поля, розташування воріт і характеристика м'яча відносяться до зовнішніх факторів – зовнішньої аферентації.

Крім зовнішніх факторів, на об'єкт керування (футболіст) також впливають внутрішні фактори, оцінювання свого внутрішнього стану, чи зможе

він виконати дію (завдати удар по воротах), чи ні. Вплив внутрішніх та зовнішніх факторів називають обстановочною аферентацією (рис. 2.5).

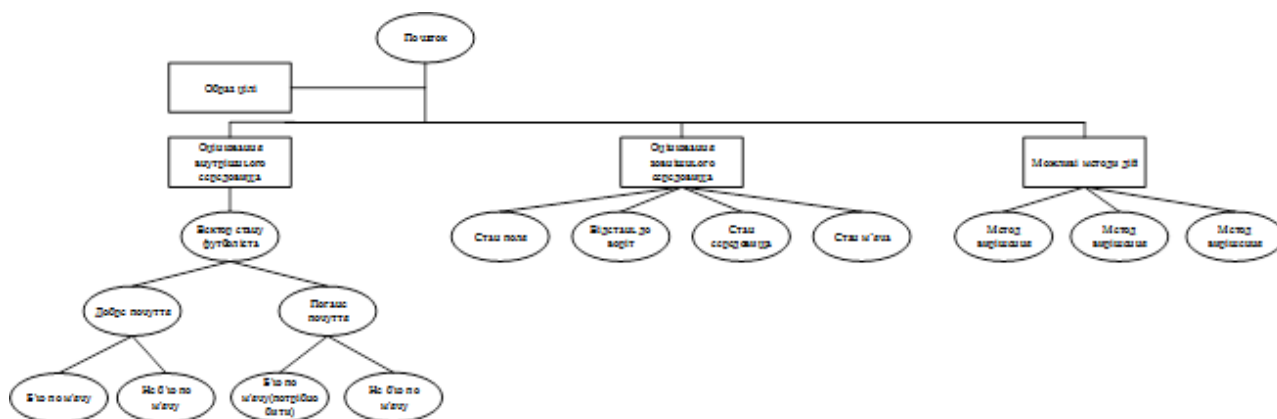


Рисунок 2.5 – Обстановочної аферентація

## 2.6 Висновки по розділу 2

На базі аналізу проблем керування в ергатичних системах та аналізу предметної дії людини-оператора було обрано програмний комплекс. Спроектвані модулі вирішення задач шляхом математичної моделі та на основі формування образу. Здійснено огляд подібних систем в задачах керування. Проаналізовано застосування нейронних мереж при моделюванні, проаналізовано та обраний метод навчання нейронної мережі, а саме алгоритм зворотного поширення помилки. Побудована нейрона мережа. Побудована функціональна структура комплексу, змодельований навколишній простір та об'єкт керування, побудовано обстановочну аферентацію на базі нечіткої логіки.

## РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ РОЗРОБЛЕНОГО МЕТОДУ МОДЕЛЮВАННЯ ОБРАЗУ ЗАДАЧІ КЕРУВАННЯ

### 3.1 Розробка структурних складових інформаційних процесів предметної дії у футболіста

Базуючись на загальній схемі предметної дії, будуємо структурну схему предметної дії футболіста (рис. 3.1) та створюємо модель в середовищі Matlab.

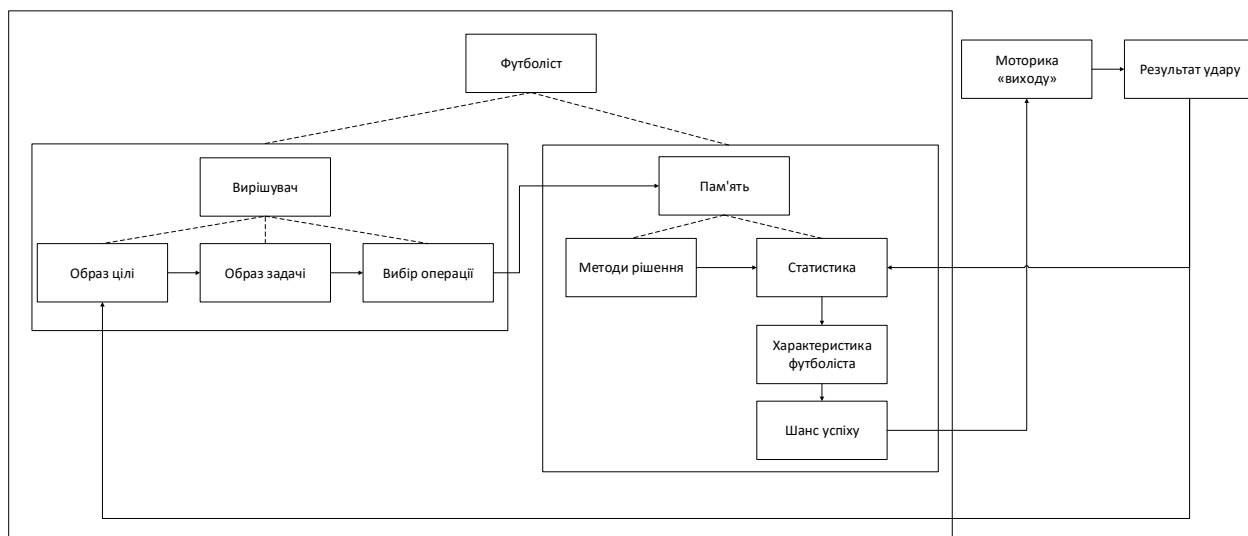


Рисунок 3.1 – Предметна дія футболіста

В першу чергу футболіст оцінює своє самопочуття, тобто проходить аферентацію для розуміння буде він завдавати удар по воротах, чи ні (рис. 3.2.).

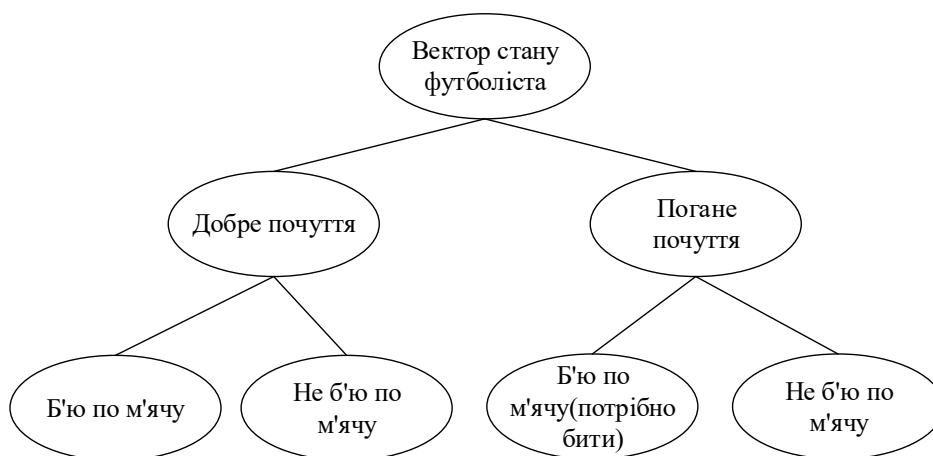


Рисунок 3.2 – Вектор стану футболіста

### **3.2 Розробка методики і планування експериментального дослідження**

Для тестування моделювання образу задачі керування проводяться експериментальні дослідження, що можна представити у вигляді подання інструкцій:

1. Інструкція футбольного тренера:
  - Усна інструкція під час навчання футболістів техніці удару.
2. За допомогою методичних вказівок:
  - Методика навчання студентів вищих навчальних закладів елементів гри в футбол.

Кращі показники роботи керування зможуть бути досягнуті шляхом навчання нейронної мережі для можливих параметрів удару та порівняння результатів інструкцій тренера та програми.

### **3.3 Проведення експериментальних дослідження адекватності розроблених моделей**

Для експериментального дослідження було обрано дії удару по м'ячу. Якщо футболіст знаходиться в гарній формі, тобто задовольняє умови аферентації, то ми можемо завдавати удар по воротам (рис. 3.3).

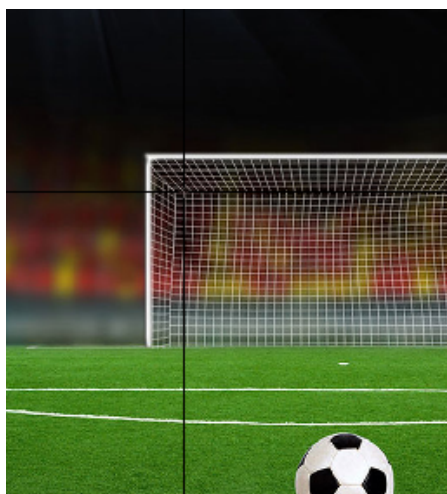


Рисунок 3.3 – Вибір точки забиття м'яча



Загально прийнято розподіляти ворота на певні зони (рис. 3.4), для тестування було обрано ліву дев'ятку.

9	8	7	7	8	9
6	5	4	4	5	6
3	2	1	1	2	3

Рисунок 3.4 – Схема розподілу воріт на зони

Виконавши дані етапи, користувач отримує інструкцію, як потрібно діяти щоб влучити в обрану ним ціль (рис. 3.5).

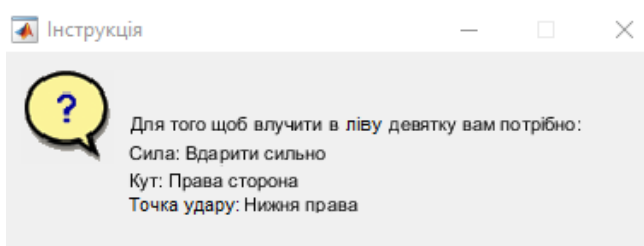


Рисунок 3.5 –Інструкція для футболіста програми

### 3.4 Аналіз адекватності отриманих результатів

Адекватність отриманих результатів перевіряється шляхом співставлення інструкції програми та інструкцією футбольного тренера. За основу було взято методика навчання студентів вищих навчальних закладів елементів гри в футбол (рис. 3.6) [15], та урок з точності удару футбольної школи (рис. 3.7) [16]. В результаті проведення досліджень розроблених моделей виявлено, що програма видає точні результати, тобто доведено реалізацію методами і засобами ШІ предметної дії людини-оператора в задачах керування.

Під час співставлення інструкції перевіряються на еквівалентність, тобто при одних й тих самих вхідних даних, ми отримуємо ті самі результати.

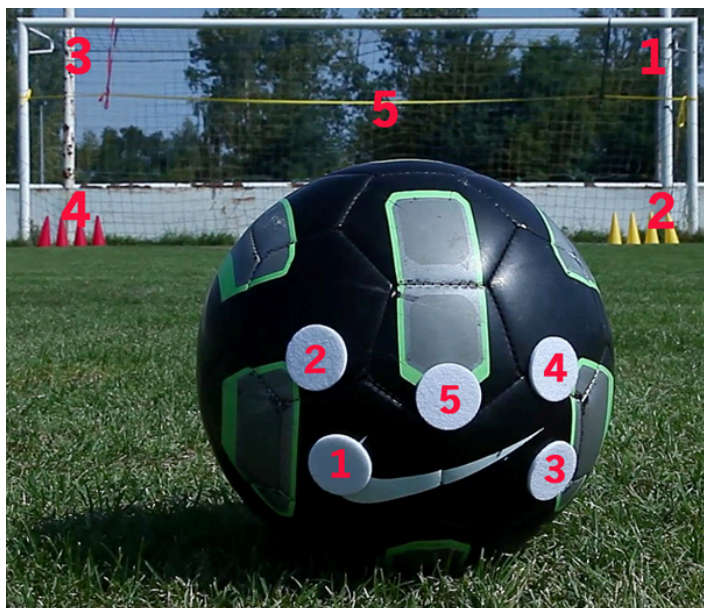


Рисунок 3.6 – Навчання удару студентів

Для того щоб потрапити в лівий верхній кут потрібно вдарити в точку 3.



Рисунок 3.7 – Урок з точності удару футбольної школи

Коли йдеться про рухову діяльність в галузі фізичного виховання і спорту, то ефективним засобом корекції може виступати професійне мовлення, тобто спортивно-термінологічний словник, як то: словесні інструкції щодо вихідних положень у виконанні вправ, назви рухів.

### 3.5 Висновки по розділу 3

Розроблено програмне забезпечення у вигляді m. файлів в середовищі Matlab для моделювання образу задачі керування використавши напрацювання психологів, як основу для моделювання предметної дії засобами штучного інтелекту. Під час розробки було використана рекурсивна багатошарова нейронна мережа з методом зворотного поширення помилки.

В якості тестування та проведення експериментальних досліджень було обрано задачу удару по м'ячу. Використовуючи нечітку логіку та нейронну мережу аналізується стан футболіста та пропонується інструкція для досягнення образу цілі. Адекватність отриманих результатів перевіряється шляхом співставлення інструкції програми та інструкцією футбольного тренера. В результаті проведення досліджень розроблених моделей виявлено, що програма видає точні результати, тобто доведено реалізацію методами і засобами ШІ предметної дії людини-оператора в задачах керування.

## ВИСНОВКИ

В ході дослідження було розглянуто актуальну наукову проблему, а саме створення імітаційної моделі предметної дії людини-оператора в задачах керування.

В процесі написання даної роботи отримані наступні результати:

1. Проаналізована предметна дія оператора з позиції інженерної психології.
2. Проаналізовані та підібрані відповідники структурних складових та інформаційних процесів предметної дії в методах і засобах штучного інтелекту.
3. Змодельована структурна складова інформаційних процесів предметної дії.
4. Організовано взаємодію між структурними складовими інформаційних процесів.
5. Сформовано образ задачі керування, як трійку що складається з того що є, що має бути та що для цього потрібно зробити.
6. Перевірено адекватності отриманого образу.

Практичною значимістю даної роботи є: розроблені моделі та методи можуть бути використані в наукових дослідженнях шляхом масштабування, накопичення даних та використаний в області робототехніці, тренажерній підготовці операторів для вирішення задач тренування.

Науковою новизною виконаної роботи є: вперше створена імітаційна модель предметної дії оператора на основі інженерно-психологічних уявлень шляхом застосування методів та засобів штучного інтелекту.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рубинштейн С. Л. Основы общей психологии, 1998. – 674с.
2. Анохин П. К. Принципиальные вопросы общей теории функциональных систем, 1973. – 61с.
3. Котик М.А., Емельянов А.М. Ошибки управления, 1985. – 392с.
4. Моделирование полёта футбольного мяча [Электронный ресурс]. – Режим доступа:  
[http://genius.pstu.ru/file.php/1/pupils\\_works\\_2013/Ilenko\\_Djakov.pdf](http://genius.pstu.ru/file.php/1/pupils_works_2013/Ilenko_Djakov.pdf)
5. Васильев А.Н., Тархов Д.А. Применение искусственных нейронных сетей, 2005. – 37-47с.
6. Субботін С. О Нейронні мережі : теорія та практика: навч. посіб. / С. О. Субботін. – Житомир : Вид. О. О. Євенок, 2020. – 184 с.
7. Managing Functional Biases in Organizational Forecasts: A Case Study of Consensus Forecasting in Supply Chain Planning [Электронный ресурс]. – Режим доступа: <http://www.hbs.edu/research/pdf/07-024.pdf>
8. Day-ahead electricity prices forecasting based on time series models: a comparison [Электронный ресурс]. – Режим доступа: [http://www.pscs-central.org/uploads/tx\\_ethpublications/s15p06.pdf](http://www.pscs-central.org/uploads/tx_ethpublications/s15p06.pdf)
9. Basaran Filik U.,Kurban M. A New Approach for the Short-Term Load Forecasting with Autoregressive and Artificial Neural Network, 2007. – 71с.
10. Chuchueva I. The time series extrapolation model based on maximum likeness set, 2010. – 283с.
11. H. Haken. Principles of Brain Functioning: A Synergetic Approach to Brian Activity, Behavior and Cognition, 1999. – 59с.
12. Yarbrow JW. Conference on prognostic factors and staging in cancer management. contributions of artificial neural networks and other statistical methods, Arlington, Virg, 2001. – 91с.
13. Юсупова Ш. А. Система прогнозирования на базе нейронных сетей, 2014. – 354-358с.

14. Стасюк Р. М., Левковець О. Г. Методика навчання студентів вищих навчальних закладів елементів гри в футбол [Електронний ресурс]. – Режим доступу: [http://lib.sumdu.edu.ua/library/docs/rio/2017/Stasiuk\\_metodyka.pdf](http://lib.sumdu.edu.ua/library/docs/rio/2017/Stasiuk_metodyka.pdf)

15. Футбольная школа. Урок 3. Точность удара [Електронний ресурс]. – Режим доступу: <https://youtu.be/Npl-Bcmnt1A>

## ДОДАТОК А Matlab скрипт

```
function varargout = my_graf_out(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @my_graf_out_OpeningFcn, ...
    'gui_OutputFcn', @my_graf_out_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before my_graf_out is made visible.
function my_graf_out_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure

% Choose default command line output for my_graf_out
handles.output = hObject;
```

```

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes my_graf_out wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = my_graf_out_OutputFcn(hObject, eventdata, handles)
% Get default command line output from handles structure
varargout{1} = handles.output;

function btnPush_Callback(hObject, eventdata, handles)
figure(1);
Vrata;
%figure(my_graf);

function radio_btn3_net_Callback(hObject, eventdata, handles)

function axes_pol_CreateFcn(hObject, eventdata, handles)
Stadion=imread('stadion.jpg');
image(Stadion)
set(gca, 'XTick', []);
set(gca, 'YTick', []);

function popupmenu2_Callback(hObject, eventdata, handles)
contents=cellstr(get(hObject, 'String'));
pop_choice=contents{get(hObject, 'Value')};
if(strcmp(pop_choice, '0.6'))
    Tuck=0.6

```



```
elseif(strcmp(pop_choice, '0.8'))
```

```
    Tuck=0.8
```

```
elseif(strcmp(pop_choice, '1.1'))
```

```
    Tuck=1.1
```

```
end
```

```
function popupmenu2_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```
function popupmenu3_Callback(hObject, eventdata, handles)
```

```
contents=cellstr(get(hObject, 'String'));
```

```
pop_choice=contents{get(hObject, 'Value')};
```

```
if(strcmp(pop_choice, '410'))
```

```
    Baga=410
```

```
elseif(strcmp(pop_choice, '430'))
```

```
    Baga=430
```

```
elseif(strcmp(pop_choice, '450'))
```

```
    Baga=450
```

```
end
```

```
function popupmenu3_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```

function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
switch(get(eventdata.NewValue,'Tag'));
    case 'radio_btn1_yus'
        contents=cellstr(get(hObject, 'String'));
        pop_choice=contents{get(hObject, 'Value')};
        if(strcmp(pop_choice, 'Високий'))
            Gazon=1
        end
    case 'radio_btn1_nuz'
        contents=cellstr(get(hObject, 'String'));
        pop_choice=contents{get(hObject, 'Value')};
        if(strcmp(pop_choice, 'Низький'))
            Gazon=0
        end
end

```

```

function uibuttongroup2_SelectionChangedFcn(hObject, eventdata, handles)
switch(get(eventdata.NewValue,'Tag'));
    case 'radio_btn2_da'
        contents=cellstr(get(hObject, 'String'));
        pop_choice=contents{get(hObject, 'Value')};
        if(strcmp(pop_choice, 'Гарна'))
            Pogoda=1
        end
    case 'radio_btn2_net'
        contents=cellstr(get(hObject, 'String'));
        pop_choice=contents{get(hObject, 'Value')};
        if(strcmp(pop_choice, 'Погана'))
            Pogoda=0
        end
end

```

end

```
function check_goal(y, t_start, t_final, dt, goal_length, goal_height)
```

```
%initializing variables
```

```
n=1;
```

```
temp_n = 0;
```

```
n_total = ceil((t_final - t_start)/dt);
```

```
t = zeros(1, n_total);
```

```
t(1) = t_start;
```

```
%finding n where ball crosses axes
```

```
while t(n) < t_final
```

```
    % Check for overshooting
```

```
    if t(n)+dt > t_final
```

```
        % If new time t(n+1)=t(n)+dt is greater than t_final we reduce dt
```

```
        % so that that new time t(n+1) would be exactly t_final
```

```
        dt = t_final-t(n);
```

```
    end
```

```
    % Calculate new time
```

```
    t(n+1) = t(n) + dt;
```

```
%checking what n will be when y crosses the goal
```

```
if((y(2,n)>0) && (y(2,n+1)<0))
```

```
    temp_n = n;
```

```
end
```

```

    n= n+1;
end

% finding x, y, z positions for n
if (temp_n ~= 0)
    n = temp_n;
    time_cross = -y(2,n)*(t(n+1)-t(n))/(y(2,n+1)-y(2,n)) + t(n);
    x_cross = y(1,n) + (time_cross - t(n))*(y(1,n+1)-y(1,n))/(t(n+1)-t(n));
    z_cross = y(3,n) + (time_cross - t(n))*(y(3,n+1)-y(3,n))/(t(n+1)-t(n));

end

%checking to see if a goal was made
if(abs(x_cross) <= goal_length/2 && z_cross <= goal_height)
    disp('made the goal');
else
    disp('missed the goal');
end

end

function [error2] = calculate_max_error(y, y_exact, t_start, t_final, dt)

```

```

%error checking code bellow
n_total = ceil((t_final - t_start)/dt);
n = 1;
error = 0;
total = 0;
test = zeros(1, n_total);
while n < n_total

[rows,columns] = size(y);
m=1;
while m < rows

    total = total + (y_exact(m,n)-y(m,n))^2;
    test(n) = test(n) + (y_exact(m,n)-y(m,n))^2;
    m = m+1;

end

% max_error = sqrt(total);
%
% %checking for highest error
% if max_error > error
%     error = max_error;
% end

%fixed error
    error2 = sqrt(max(test));
    n = n + 1;
end

```

```

clear;
clc;
close all;

%time
t_start = 0;
t_final = 1;

%?t = 0.1, 0.05, 0.025, 0.0125, 0.00625
i=1;
dt = [0.1; 0.05; 0.025; 0.0125; 0.00625];

% initial conditions
y_start = [ 0; 1; 0; 0; 2; 1 ];

f = @(t, y) [y(2);
             sin(1-exp(y(3)));
             1/(1-log(y(5) -1));
             (1/2*t^2+t)*y(2)+exp(y(3))*y(1);
             1-y(5);
             -3*((exp(y(3))-1)/1+t^3))^2; ];

%exact solution
exact_solution = @(t) [ sin(t);
                       cos(t);
                       log(1+t);
                       (1/2*(t.^2)+t).*sin(t);
                       1+exp(-t);
                       1./(1+(t.^3)); ];

```

```

while i < 6
[t, y] = solve_ode(f, y_start, t_start, t_final, dt(i));

y_exact = exact_solution(t);

%calculating order
[error] = calculate_max_error(y, y_exact, t_start, t_final, dt(i));
e1 = error;

%this also allows me to get an additional error for an additional order
dt(i) = dt(i)/2;
[t, y] = solve_ode(f, y_start, t_start, t_final, dt(i));
y_exact = exact_solution(t);
[error] = calculate_max_error(y, y_exact, t_start, t_final, dt(i));
e2 = error;

order = log((e1)/(e2))/log(2);
fprintf('dt = %1.5f \nerror = %i \norder of accuracy = %1.5f\n\n', dt(i)*2, e1, order);

i = i + 1;
end

function check_wall(y, t_start, t_final, dt, wall_x_start, wall_y_start, wall_x_end,
wall_y_end, wall_height)

%initializing variables

```

```

n=1;
temp_n = 0;
n_total = ceil((t_final - t_start)/dt);
t = zeros(1, n_total);
t(1) = t_start;

%finding n where ball crosses axes
while t(n) < t_final

    % Check for overshooting
    if t(n)+dt > t_final
        % If new time t(n+1)=t(n)+dt is greater than t_final we reduce dt
        % so that that new time t(n+1) would be exactly t_final
        dt = t_final-t(n);
    end

    % Calculate new time
    t(n+1) = t(n) + dt;

    %y position prior the wall needs to be greater than "before"
    %and y position post the wall needs to be less than "after"
    before = wall_y_start + (y(1,n) - wall_x_start)*(wall_y_end-
wall_y_start)/(wall_x_end-wall_x_start);
    after = wall_y_start + (y(1,n+1) - wall_x_start)*(wall_y_end-
wall_y_start)/(wall_x_end-wall_x_start);

    %checking logic from above
    if((y(2,n)>before) && (y(2,n+1)<after))

```



```

    temp_n = n;
end

n= n+1;
end

% finding x, y, z positions for n
if (temp_n ~= 0)
    n = temp_n;

    %eq1 and eq2 are to clean up the x,y,z position calculations
    eq1 = (y(2,n+1) - y(2,n))/(y(1,n+1) - y(1,n));
    eq2 = (wall_y_end - wall_y_start)/(wall_x_end - wall_x_start);
    x_cross = (wall_y_start - y(2,n)+y(1,n)*eq1-wall_x_start*eq2)/(eq1-eq2);

    y_cross = y(2,n) + (x_cross-y(1,n))*eq1;

    eq1 = (y(3,n+1) - y(3,n))/(y(1,n+1) - y(1,n));
    z_cross = y(3,n) + (x_cross-y(1,n))*eq1;

end

%setting up boolean to see if x and y will be within the wall bounds
check_x_cross = ((x_cross > wall_x_start) && (x_cross < wall_x_end));
check_y_cross = ((y_cross > wall_y_start) && (y_cross < wall_y_end));

%taking those booleans and checking that z will be within the wall bounds

```

```

if( ((check_x_cross) || (check_y_cross)) && (z_cross < wall_height))
    disp('hit wall, but if it made it past it would of');
else
    disp('made it over wall and');
end

end

```

```

%system of equations

```

```

y_start1 = [ -15; 23; 0; 18; -23; 8.5; ];

```

```

y_start2 = [ 25.5; 15; 0; -28; -11; 7.5; ];

```

```

y_start3 = [ -4; 35; 0; -8; -36; 5; ];

```

```

y_start4 = [ 16; 28; 0; -25; -20; 8; ];

```

```

bby_start = [y_start1 y_start2 y_start3 y_start4];

```

```

%loopoing through all 4 cases

```

```

while j < 2

```

```

h = @(t,y,s) [ y(4);

```

```

    y(5);

```

```

    y(6);

```

```

    (-c_drag*sqrt(y(4)^2+y(5)^2+y(6)^2)*y(4) +

```

```

    c_lift*sqrt(y(4)^2+y(5)^2+y(6)^2)*(bb(2,j)*y(6)-bb(3,j)*y(5)))/mass;

```

```

    (-c_drag*sqrt(y(4)^2+y(5)^2+y(6)^2)*y(5) +

```

```

    c_lift*sqrt(y(4)^2+y(5)^2+y(6)^2)*(bb(3,j)*y(4)-bb(1,j)*y(6)))/mass;

```

```

    (-mass*g-c_drag*sqrt(y(4)^2+y(5)^2+y(6)^2)*y(6) +

```

```

    c_lift*sqrt(y(4)^2+y(5)^2+y(6)^2)*(bb(1,j)*y(5)-bb(2,j)*y(4)))/mass;

```

```

    ];

```

```
my_h = @(t, y) h(t, y, bb(:,j));
```

```
%solve ode
```

```
[t,y] = solve_ode(my_h, bby_start(:,j), t_start, t_final, dt);
```

```
%3D plot
```

```
plot_trajectory(y(1,:),y(2,:),y(3,:), wall_x_start(j), wall_y_start(j), wall_x_end(j),  
wall_y_end(j), wall_height)
```

```
if j ~=1
```

```
    figure;
```

```
end
```

```
j = j+1;
```

```
end
```

## ДОДАТОК Б Проектування системи вирішення задачі шляхом математичної моделі

При вирішенні різноманітних задач з використанням обчислювальних машин вирішення задачі передбачає формування певного уявлення про систему (рис. Б.1).



Рисунок Б.1 – Загальна схема вирішення задачі

Для більш явного представлення про моделювання образу задачі керування в якості прикладу було обрано математичну модель задачі пошуку параметрів удару по м'ячу [4] (рис. Б.2).



Рисунок Б.2 – Модель удару по м'ячу

Для побудови математичної моделі приймемо наступну сукупність гіпотез:

1. М'яч будемо вважати об'єктом масою  $m$ , радіусом  $r$ , центр мас якого є постійним.

1. Рух відбувається в поле сил тяжіння з постійним прискоренням вільного падіння  $g$ , прискорення можна вважати постійною величиною (так, як висота на яку піднімається м'яч, багато менше радіуса Землі) і описується рівняннями класичної механіки Ньютона (так, як швидкість м'яча багато менше швидкості світла).

2. Рух м'яча відбувається в тривимірному просторі.

3. Врахуємо також силу опору повітря і ефект Магнуса.

Для обґрунтування справедливості прийнятої нами гіпотези, знайдемо прискорення м'яча (масою 450г, радіусом 11 см, що рухається з лінійною швидкістю  $35 \frac{m}{c}$  та кутовою швидкістю  $10 \frac{об}{c}$  в повітряному середовищі, в'язкість якої при нормальних умовах становить  $17.20 \cdot 10^{-6} \text{ Па} \cdot \text{с}$ ) [5] в двох ситуаціях: спочатку без урахування сили опору і ефекту Магнуса, а потім з урахуванням їх впливу. Прискорення тіла в першому випадку дорівнює прискоренню вільного падіння, тобто  $a = g = 9.8 \frac{m}{c^2}$ . Прискорення тіла в другому випадку (при врахуванні сили опору і ефекту Магнуса) дорівнює:

$$a = \frac{F_p}{m} = \frac{F_g - F_{\text{опор}} + F_m}{m} = \frac{4.41 - 0.0014258 + 2 \cdot 62.8 \cdot 40 \cdot 17.20 \cdot 10^{-6}}{0.45} = 9.98 \frac{m}{c^2}. \quad (\text{Б.1})$$

Таким чином, встановлено, що прискорення тіла залежить від ефекту Магнуса і сили опору повітря. Значить, для побудови адекватної моделі, їх необхідно враховувати.

Основним законом механіки є другий закон Ньютона, який зв'язує рівнодію зовнішніх сил, що діють на тіло, його масу і прискорення, отримане в результаті дії сил. Запишемо другий закон Ньютона для м'яча в векторній формі, де  $m$  - маса м'яча,  $a$  - його прискорення, що з'явилося в результаті дії сили,  $F_p$  –

рівнодіюча сила, діюча на тіло,  $F_p$  дорівнює векторній сумі силі тяжіння  $F_{тяж}$ , силі опору повітря  $F_{опору}$  та силі Магнуса  $F_M$ :

$$\vec{F}_p = m\vec{a}, \quad (\text{Б.2})$$

$$\vec{F}_p = \vec{F}_{тяж} + \vec{F}_{опору} + \vec{F}_M. \quad (\text{Б.3})$$

При цьому мається на увазі, що сила і маса - постійні величини. У такому випадку і прискорення теж буде постійною величиною. Отже, рівняння моделює рівноприскореному русі тіла з постійною масою під дією постійної сили.

Нехай в початковий момент часу  $t_0$  снаряд розташований на початку координат  $r(0)$  ( $r(0)=0$ ), а його початкова швидкість в цей момент часу дорівнює  $v$  ( $v(0)=v_0$ ). Спроекуємо на кожен з осей  $x$ ,  $y$ ,  $z$  другий закон Ньютона, розкладемо по ним вектор швидкості і прискорення (рис. Б.1):

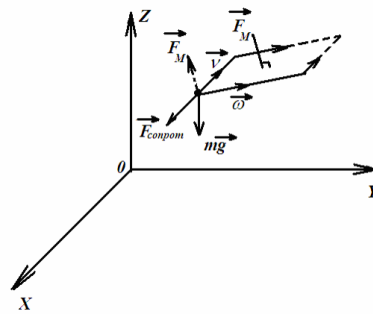


Рисунок 2.3 – Сили, що діють на матеріальну точку

В проекції на вісь  $x$ :

$$F_{px} = ma_x; \quad (\text{Б.4})$$

$$F_{px} = F_{тяжx} + F_{опоруx} + F_{Mx}; \quad (\text{Б.5})$$

$$ma_x = F_{тяжx} + F_{опоруx} + F_{Mx}; \quad (\text{Б.6})$$

В проекції на вісь  $y$ :

$$F_{py} = ma_y; \quad (\text{Б.7})$$

$$F_{py} = F_{тяжy} + F_{опоруy} + F_{My}; \quad (\text{Б.8})$$

$$ma_y = F_{тяжy} + F_{опоруy} + F_{My}; \quad (\text{Б.9})$$

В проекції на вісь  $z$ :

$$F_{pz} = ma_z; \quad (\text{Б.10})$$

$$F_{pz} = F_{\text{тяж}z} + F_{\text{опору}z} + F_{Mz}; \quad (\text{Б.11})$$

$$ma_z = F_{\text{тяж}z} + F_{\text{опору}z} + F_{Mz}; \quad (\text{Б.12})$$

Силу опору будемо розраховувати за формулою Стокса [6] (рис. Б.3)

$$F_{\text{опору}} = 6\pi\eta\nu R, \quad (\text{Б.13})$$

де  $\eta$  – це динамічна в'язкість середовища,  $\nu$  – це швидкість м'яча, а  $R$  – це радіус м'яча.

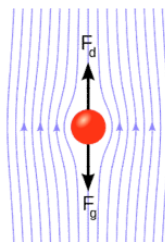


Рисунок Б.4 – Сили, які діють на тіло, що рухається в середовищі

На рис. Б.2 зображено тіло, що рухається в середовищі, сили, що діють на тіло в проекції на вертикальну вісь і лінії оточуючого потоку.

Таким чином,  $F_d$  – це шукана сила опору, діюча на тіло зі сторони в середовищі, а  $F_g$  – це сила тяжіння. Силу тяжіння приймем  $F_g = mg$  (де  $g$  – прискорення вільного падіння), сила опору визначається формулою (Б.13), силу Магнуса будемо обраховувати по формулі  $F_M = \lambda(\omega \times \nu)$  (де  $\omega$  – це кутова швидкість м'яча, а  $\nu$  – це лінійна швидкість м'яча).

Футболіст, штовхаючи м'яч, надає йому кутову швидкість, яку можна знайти, використовуючи рівняння обертального руху:

$$I \frac{\Delta\omega}{\Delta t} = M; \quad (\text{Б.14})$$

де  $M$  – момент сили удару, дорівнює  $M = (r \times F)$  (де  $r$  – плече сили удару,  $F$  – її модуль),  $\Delta t$  – інтервал часу взаємодії ноги з м'ячем,  $I$  – момент інерції, який дорівнює  $I = \frac{2}{3}mR^2$  (де  $R$  – радіус м'яча,  $m$  – його маса):

$$I \frac{\Delta\omega}{\Delta t} = M; \quad (\text{Б.15})$$

$$I \frac{d\omega}{dt} = Fr; \quad (\text{Б.16})$$

$$\Delta\omega = \omega_k - \omega_0, \omega_0 \rightarrow 0; \quad (\text{Б.17})$$

Знайдемо плече сили удару в разі, якщо вона спрямована довільно. Надалі довжина плеча буде використана для знаходження моменту сили і кутової швидкості. Для цього позначимо через  $l$  шукане плече, нехай  $R$  - радіус м'яча,  $r$  - відстань від точки прикладання сили до осі обертання,  $F_{\text{уд}}$  - сила удару,  $\gamma$  - кут між лінією дії сили та віссю,  $\alpha$  - кут між радіусом м'яча та віссю.

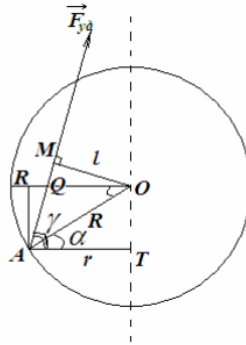


Рисунок Б.5 – М'яч в момент удару

Таким чином,

$$l = MO = R \cdot \sin(\gamma - \alpha); \quad (\text{Б.18})$$

Для знаходження початкової лінійної швидкості, скористаємося другим законом Ньютона в проекції на вісь  $x$ , де  $m$  - маса м'яча,  $a$  - прискорення,  $F$  - сила удару:

$$ma_x = F; \quad (\text{Б.19})$$

Прискорення  $a_x$  чисельно дорівнює відношенню приросту швидкості  $dv_x$  на безкінечно малому інтервалі часу  $dt$ :

$$m \frac{dv_x}{dt} = F; \quad (\text{Б.20})$$

Висловимо збільшення швидкості з попереднього виразу і підсумуємо його:

$$\sum dv_x = \sum \frac{F}{m} dt; \quad (\text{Б.21})$$

Масу м'яча будемо вважати постійною, отже, її можна винести з-під знака суми:

$$\Delta v_x = \frac{1}{m} \int_{t_0}^{t_k} F dt; \quad (\text{Б.22})$$



де вираз  $\int_{t_0}^{t_k} F dt$  дорівнює площі під графіком  $S_f$ , тобто  $\int_{t_0}^{t_k} F dt = S_F$ :

$$\Delta v_x = \frac{1}{m} S_F; \quad (\text{Б.23})$$

Розпишемо векторний добуток векторів  $\omega$  і  $v$ :

$$(\vec{\omega} \times \vec{v}) = \begin{bmatrix} i & j & k \\ \omega_x & \omega_y & \omega_z \\ v_x & v_y & v_z \end{bmatrix} = (\omega_y v_z - \omega_z v_y)i + (\omega_z v_x - \omega_x v_z)j + (\omega_x v_y - \omega_y v_x)k; \quad (\text{Б.24})$$

Тоді проекція векторного добутку на певну вісь запишеться наступним чином:

$$(\omega \times v)_x = \omega_y v_z - \omega_z v_y; \quad (\text{Б.25})$$

$$(\omega \times v)_y = \omega_z v_x - \omega_x v_z; \quad (\text{Б.26})$$

$$(\omega \times v)_z = \omega_x v_y - \omega_y v_x. \quad (\text{Б.27})$$

З огляду на формули (1), проекції прискорення на відповідні осі перепишемо у вигляді:

$$a_x = \frac{6\pi\eta R}{m} v_x - \frac{2}{m} \lambda (\omega \times v)_x; \quad (\text{Б.28})$$

$$a_y = \frac{6\pi\eta R}{m} v_y - \frac{2}{m} \lambda (\omega \times v)_y; \quad (\text{Б.29})$$

$$a_z = \frac{6\pi\eta R}{m} v_z - \frac{2}{m} \lambda (\omega \times v)_z. \quad (\text{Б.30})$$

З урахуванням формул (1), проекції прискорень на кожен з осей повторно в наступному вигляді:

$$a_x = \frac{6\pi\eta R}{m} v_x - \frac{2}{m} \lambda (\omega \times v)_x = \frac{6\pi\eta R}{m} v_x - \frac{2}{m} \lambda (\omega_y v_z - \omega_z v_y); \quad (\text{Б.31})$$

$$a_y = \frac{6\pi\eta R}{m} v_y - \frac{2}{m} \lambda (\omega \times v)_y = \frac{6\pi\eta R}{m} v_y - \frac{2}{m} \lambda (\omega_z v_x - \omega_x v_z); \quad (\text{Б.32})$$

$$a_z = \frac{6\pi\eta R}{m} v_z - \frac{2}{m} \lambda (\omega \times v)_z = \frac{6\pi\eta R}{m} v_z - \frac{2}{m} \lambda (\omega_x v_y - \omega_y v_x). \quad (\text{Б.33})$$

Покажемо застосування методу Ейлера до розрахунку руху падаючого тіла. Розрахунок проводиться від початкового моменту часу  $t = 0$  з малим кінцевим кроком по часу  $\Delta t$ . Нехай за час  $\Delta t$  швидкість зміниться на величину

$\Delta v$ . З огляду на співвідношення для прискорень, запишемо співвідношення для швидкостей:

$$v_x^1 = v_x^0 + a_x^0 \Delta t = v_x^0 + \left( \frac{6\pi\eta R}{m} v_x - \frac{2}{m} \lambda (\omega_y v_z - \omega_z v_y) \right) \Delta t \approx \frac{\Delta x}{\Delta t}; \quad (\text{Б.34})$$

$$v_y^1 = v_y^0 + a_y^0 \Delta t = v_y^0 + \left( \frac{6\pi\eta R}{m} v_y - \frac{2}{m} \lambda (\omega_z v_x - \omega_x v_z) \right) \Delta t \approx \frac{\Delta y}{\Delta t}; \quad (\text{Б.35})$$

$$v_z^1 = v_z^0 + a_z^0 \Delta t = v_z^0 + \left( \frac{6\pi\eta R}{m} v_z - \frac{2}{m} \lambda (\omega_x v_y - \omega_y v_x) \right) \Delta t \approx \frac{\Delta z}{\Delta t}. \quad (\text{Б.36})$$

Далі міркування ведеться по індукції. Маючи в своєму розпорядженні значенням  $v^1$ , можна, відштовхуючись від нього, знайти  $v^2$  – швидкість в момент часу  $2\Delta t$  і т.д. Загальний вигляд формули стосовно до даної задачі вийде таким:

$$v_x^{i+1} = v_x^i + a_x^i \Delta t = v_x^i + \left( \frac{6\pi\eta R}{m} v_x - \frac{2}{m} \lambda (\omega_y v_z - \omega_z v_y) \right) \Delta t \approx \frac{\Delta x}{\Delta t}; \quad (\text{Б.37})$$

$$v_y^{i+1} = v_y^i + a_y^i \Delta t = v_y^i + \left( \frac{6\pi\eta R}{m} v_y - \frac{2}{m} \lambda (\omega_z v_x - \omega_x v_z) \right) \Delta t \approx \frac{\Delta y}{\Delta t}; \quad (\text{Б.38})$$

$$v_z^{i+1} = v_z^i + a_z^i \Delta t = v_z^i + \left( \frac{6\pi\eta R}{m} v_z - \frac{2}{m} \lambda (\omega_x v_y - \omega_y v_x) \right) \Delta t \approx \frac{\Delta z}{\Delta t}. \quad (\text{Б.39})$$

Застосовуючи аналогічний підхід для обчислення переміщення, отримуємо формулу методу Ейлера для обчислення переміщення тіла, що рухається під кутом до горизонту з часом:

$$x^{i+1} = x^i + v_x^i \Delta t, \quad (\text{Б.40})$$

$$y^{i+1} = y^i + v_y^i \Delta t, \quad (\text{Б.41})$$

$$z^{i+1} = z^i + v_z^i \Delta t. \quad (\text{Б.42})$$

Розроблений алгоритм (Б.31) - (Б.42) був реалізований у вигляді програми для, що дозволяє визначити дальність траєкторії і відобразити у вигляді графіка траєкторію руху м'яча в тривимірному просторі протягом певного проміжку часу в залежності від поставлених користувачем параметрів. З використанням розробленої і реалізованої програми проведені наступні чисельні експерименти:

$$1. m = 0.45 \text{ кг}, R = 0.11 \text{ м}, F_{\text{уд}} = 20 \text{ Н}, \Delta t = 0.8 \text{ с}, a = 48, \beta = 60, \eta = 0.0000172 \text{ Па} \cdot \text{с}$$

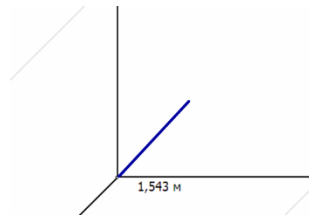


Рисунок Б.6 – Траєкторія м'яча, її кінцеве значення при параметрах, заданих в експерименті

## ДОДАТОК В Структура моделей у Matlab

Для розробки системи було обрано засіб Matlab, в якому було реалізовані всі методи для моделювання в задачі керування у вигляді m.file та побудована штучна нейрона мережа (рис. В.1, В.2).

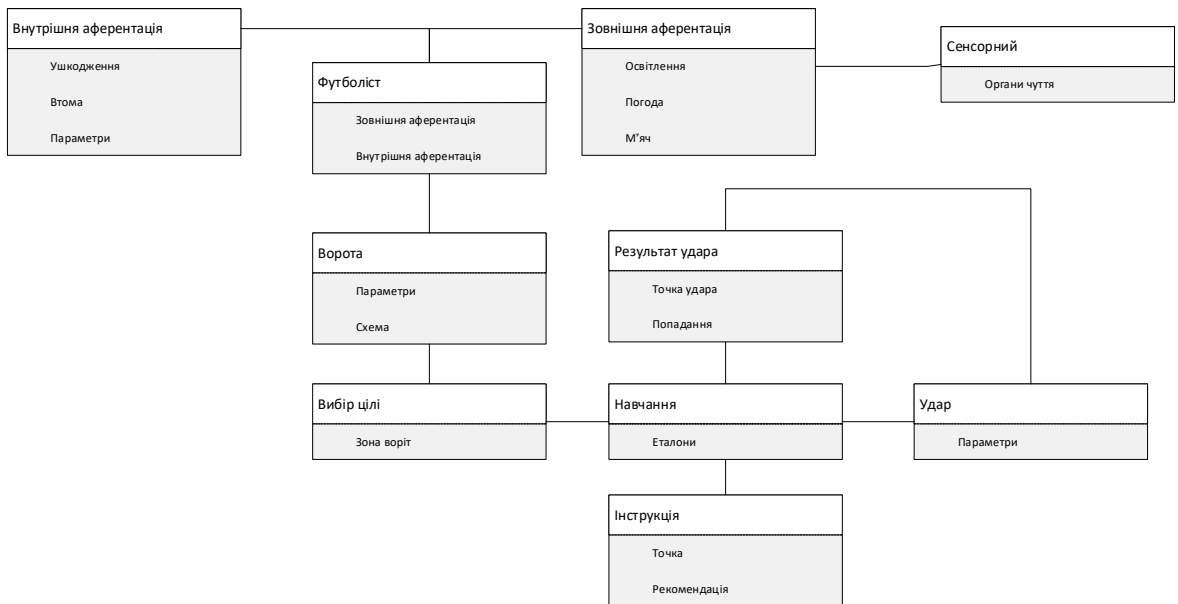


Рисунок В.1 – Функціональна структура моделі образу задачі в середовищі Matlab

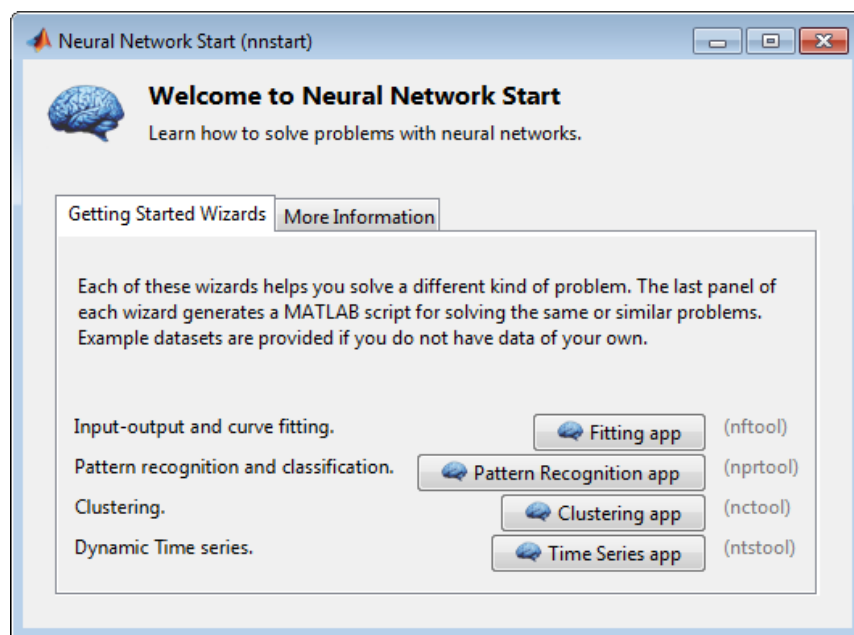


Рисунок В.2 – Вікно побудови нейронної мережі

Побудована нейронна мережа, праворуч зображено вагові коефіцієнти які відповідають за вибір методу рішення, що є аналогом вибору методу на схемі предметної дії (рис. В.3).

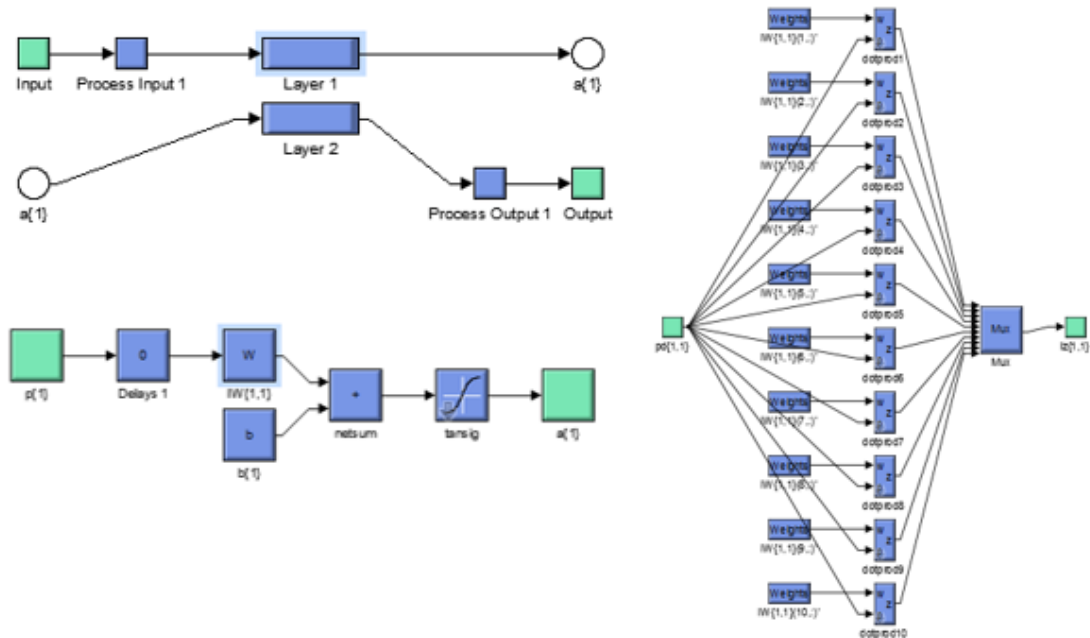


Рисунок В.3 – Побудована нейронна мережа в MatLab

Для більш точних результатів нейруну мережу потрібно навчати, постійно корегувати ваги та кількість слоїв в мережі (рис. В.4).

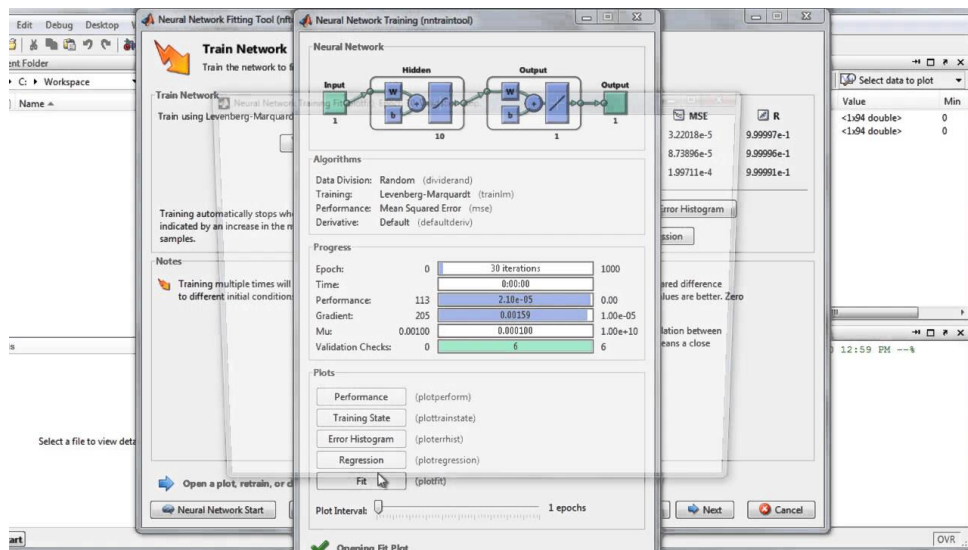


Рисунок В.4 – Навчання НМ

Результати навчання нейронної мережі: графік навчання (рис.В.5), гістограма помилок (рис.В.6), діаграма узагальнюючої спроможності (рис.В.7).

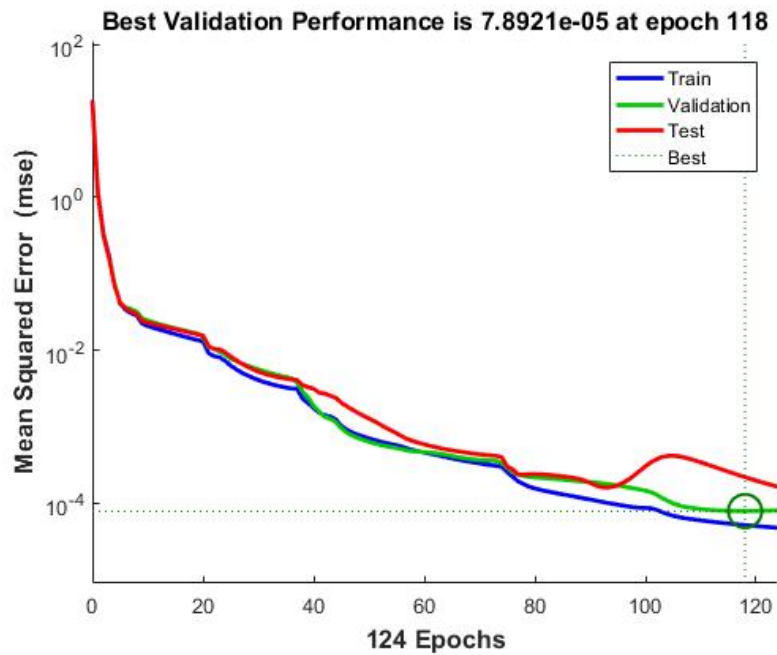


Рисунок В.5 – Графік навчання

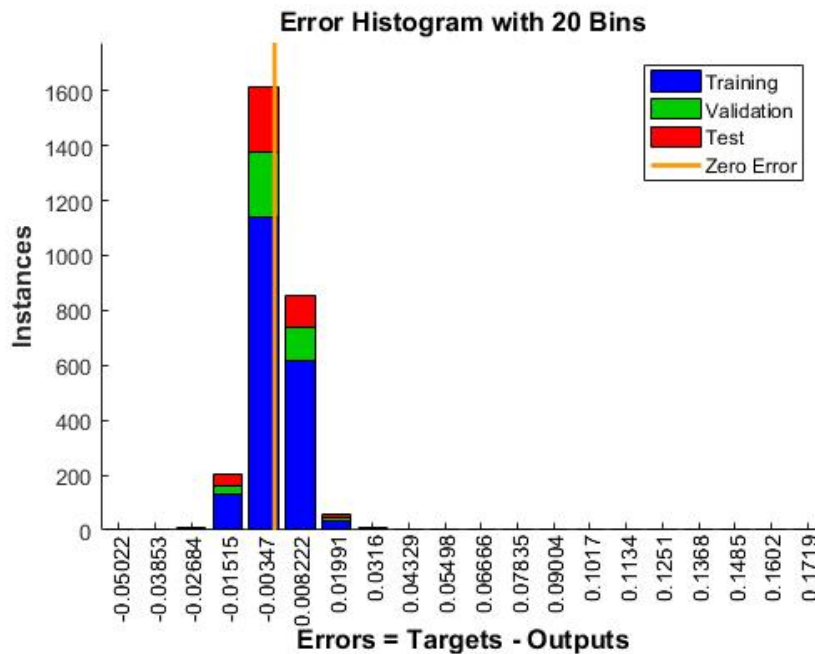


Рисунок В.6 – Гістограма помилок

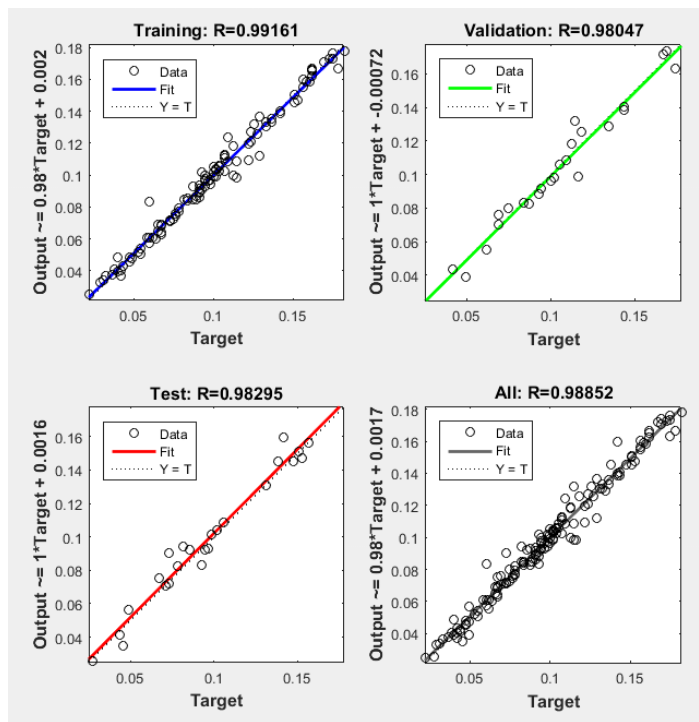


Рисунок В.7 – Діаграма узагальнюючої спроможності

Аферентація розроблена шляхом діалогу на базі нечіткої логіки (рис. Б.8).

The screenshot shows a dialog box titled 'my\_graf\_out' with two main sections:

- Внутрішня аферентація (Internal afferentation):**
  - 1. Наявні ушкодження:  Так  Ні
  - 2. Втома:  Так  Ні
- Зовнішня аферентація (External afferentation):**
  - 1. Освітлення:  Високий  Низький
  - 2. Погода:  Гарна  Погана
  - 3. М'яч:
    - Тиск: [dropdown menu]
    - Вага: [dropdown menu]

On the right side of the dialog, there is a silhouette of a soccer player and a photograph of a soccer field.

Рисунок В.8 – Обстановочна аферентація

Під час запуску в першу чергу відбувається аферентація внутрішнього та зовнішнього стану, це відповідає схемі предметної дії, адже в першу чергу футболіст повинні оцінити свій власний стан та ситуацію, що склалася