

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

Факультет прикладної математики

Кафедра прикладної математики

«На правах рукопису»
УДК 519.688

«До захисту допущено»
Завідувач кафедри

О. Р. Чертов
(підпис)

«___» 2015 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 8.04030101 «Прикладна математика»

на тему: Модель адаптивної системи регулювання дорожнього руху

Виконав: студент 2 курсу, групи КМ-31М

Душин Олександр Сергійович

(підпис)

Науковий керівник доцент, канд. техн. наук Сирота С. В.

(підпис)

Консультант із нормоконтролю старший викладач Мальчиков В. В.

(підпис)

Рецензент професор, д-р техн. наук, проф.
Симоненко В. П.

(підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

**Національний технічний університет України
«Київський політехнічний інститут»**

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти – другий (магістерський)

Спеціальність 8.04030101 «Прикладна математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О. Р. Чертов

«____» 2015 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Душину Олександру Сергійовичу

1. Тема дисертації «Модель адаптивної системи регулювання дорожнього руху»,

науковий керівник дисертації Сирота Сергій Вікторович., к.т.н., доцент, затверджені наказом по університету від «20» березня 2015р. № 785-С.

2. Термін подання студентом дисертації: «18» червня 2015 р.

3. Об'єкт дослідження: методи, моделі, системні підходи для реалізації адаптивних систем регулювання дорожнього руху.

4. Предмет дослідження: математична та програмна модель адаптивної системи регулювання дорожнього руху.

5. Перелік завдань, які потрібно розробити:

- проаналізувати існуючі рішення та встановити переваги і недоліки існуючих моделей адаптивних систем регулювання дорожнього руху;
- розробити уніфіковану модель системи адаптивного регулювання;
- удосконалити існуючий метод керування фазами руху транспортних засобів на перехресті на основі апарату нечіткої логіки;

- проаналізувати отримані результати, провести оцінку адекватності застосованих моделей на основі нечіткої логіки для системи регулювання дорожнього руху;
- провести порівняння ефективності роботи розробленої моделі адаптивної системи з вже існуючими та з моделлю не адаптивної системи.

6. Орієнтовний перелік ілюстративного матеріалу

- організаційне представлення моделі;
- модель роботи вдосконаленої підсистеми керування фазами руху;
- результати проведених експериментів.

7. Орієнтовний перелік публікацій

1. Сирота С.В., Душин О.С. Модель адаптивної систем регулювання дорожнього руху на перехресті з використанням нечіткої логіки // Прикладна математика та комп’ютинг. ПМК, 2015 : сьома наук. конф. магістрантів та аспірантів, Київ, 15—17 квіт. 2015 р. : зб. пез доп. / [редкол.: Дичка І. А. та ін.]. — К. : Просвіта, 2015.
2. Сирота С.В., Душин О.С. Модель синхронізації роботи адаптивних систем регулювання дорожнього руху на суміжних перехрестях // Системний аналіз та інформаційні технології: матеріали 17-ї Міжнародної науково-технічної конференції SAIT 2015, Київ, 22-26 червня 2015 р. / ННК “ПСА” НТУУ “КПІ”. – К.: ННК “ПСА” НТУУ “КПІ”, 2015.

8. Дата видачі завдання «25» жовтня 2013 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Вивчення літератури за тематикою магістерської дисертації	15 вересня – 30 жовтня 2013	
2	Ознайомлення з предметною областю	5 листопада 2013 – 15 лютого 2014	
3	Вивчення літератури	15 лютого–1 вересня 2014	
4	Підготовка матеріалів другого розділу дипломного проекту	1 вересня 2014–1 березня 2015	
5	Завершення роботи над основною частиною МД, переддипломна практика, робота над публікаціями	1 березня–1 травня 2015	
6	Оформлення текстової і графічної частини МД	1 травня–1 червня 2015	
7	Попередній захист МД	1 червня–15 червня 2015	
8	Оформлення дипломного проекту	1 червня–15 червня 2015	

Студент _____

О.С. Душин

Науковий керівник дисертації _____

С.В. Сирота



РЕФЕРАТ

Актуальність теми. Оптимальне керування дорожнім рухом є нагальною проблемою багатьох сучасних міст. Запобігання заторів крім того, що зменшує час, який витрачає людина в транспорті, ще й підвищує безпеку руху, зменшує кількість шкідливих викидів та поліпшує якість життя в місті в цілому.

Тому розробка моделі системи адаптивного регулювання дорожнього руху, яка дозволить враховувати поточну ситуацію в дорожній мережі в режимі реального часу є важливою та актуальною задачею для поліпшення регулювання дорожнього руху та зменшення часу, який витрачає окрема людина кожен день в дорозі використовуючи наземний транспорт.

Об'єктом дослідження є методи, моделі, системні підходи для реалізації адаптивних систем регулювання дорожнього руху.

Предметом дослідження є математична та програмна модель адаптивної системи регулювання дорожнього руху.

Мета роботи: розробка і дослідження математичної та програмної моделі адаптивної системи регулювання дорожнього руху та методів керування фазами руху транспортних засобів на перехрестях.

Методи дослідження: В роботі використовуються методи математичного моделювання, методи системного аналізу, апарат нечіткої логіки.

Наукова новизна роботи полягає в наступному:

- вдосконалено існуючий метод роботи підсистеми керування фазами руху на перехресті в рамках адаптивної системи регулювання дорожнього руху;
- розроблено уніфіковану програмну модель функціонування дорожньої ділянки руху в реальному часі, яка дає змогу оцінювати

та порівнювати ефективність роботи будь-яких підсистем чи контролерів керування фазами руху та проводити моніторинг поточної ситуації.

Практична цінність отриманих у роботі результатів полягає у тому, що запропонована модель дає змогу збільшити пропускну спроможність на окремих ділянках дорожнього руху, а розроблене програмне забезпечення проводити тестування нових контролерів керування та поліпшувати вже існуючі. Таким чином, розробник зможе створювати новий програмний комплекс регулювання дорожнього руху та оцінювати адекватність його роботи не заглиблюючись в деталі апаратної складової всієї адаптивної системи регулювання дорожнього руху.

Апробація роботи. Основні положення і результати роботи були представлені та обговорювались на VII науковій конференції магістрантів та аспірантів «Прикладна математика та комп’ютинг» ПМК-2015 та опубліковані у збірнику тез доповідей, Просвіта, 2015, а також на 17-й Міжнародній науково-технічній конференції SAIT 2015 (Київ) та опубліковані у збірнику «Системний аналіз та інформаційні технології» ННК “ПСА” НТУУ “КПГ”, 2015.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, чотирьох розділів, висновків та додатків.

У вступі надано загальну характеристику роботи, виконано оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень, сформульовано мету і задачі досліджень, показано наукову новизну отриманих результатів і практичну цінність роботи.

В першому розділі виконується огляд існуючих методів та засобів реалізації систем адаптивного регулювання дорожнього руху, підсистем керування фазами руху, а також подається їх порівняльна характеристика.

У другому розділі представлене обґрунтування методів та засобів для вирішення поставленої задачі побудови моделі.

В третьому розділі будуються діаграми основних сутностей уніфікованої моделі. Також тут продемонстровано математичну модель вдосконаленої підсистеми керування фазами.

В четвертому розділі розглядається архітектура програмної реалізації системи для моделювання функціонування перехрестя в реальному часі та оцінки роботи системи в цілому.

У висновках проаналізовано отримані результати роботи.

У додатах наведено блок-схему модифікованого алгоритму реалізації підсистеми керування фазами руху за допомогою апарату нечіткої логіки, діаграму класів та фрагменти коду програмного продукту, скріншоти результатів роботи програми та слайди презентації для супроводу доповіді.

Робота виконана на 81 аркуші, містить 2 додатки та посилання на список використаних літературних джерел з 21 найменування. У роботі наведено 23 рисунки та 6 таблиць.

Ключові слова: адаптивна система регульовання, математична модель, нечітка логіка, нейронечіткий підхід.

ABSTRACT

Theme urgency. Optimal traffic control is an urgent problem of many modern cities. Preventing congestion not only reduces the time a person spends in transport, but also increases safety, reduces harmful emissions and improves quality of life in the city in general.

Therefore, the development of a model of adaptive traffic control system is an important and urgent task to improve traffic control.

The object of research is a complex of the methods, models, systematic approaches for implementing adaptive traffic control systems.

The subject of research is a mathematical and programming model of an adaptive traffic control systems.

Research objective: development and research of the mathematical and programming models of an adaptive traffic control systems and control methods of traffic signal on the intersections.

Scientific novelty is:

- improving the existing subsystem with control methods of traffic signal within the adaptive traffic control system;
- unified programming model for traffic simulation in real time that allows to evaluate and compare the performance of the subsystems.

Practical value is obtained results and the fact that the proposed model allows to increase the performance of adaptive traffic control system. It was developed a software to test new traffic signal management subsystems and to improve existing ones. Thus, a developer can create a new software for adaptive traffic control and integrate it, without getting acquainted with the details of all the hardware components and architecture of the adaptive traffic control systems.

Approbation. The main points and the results were presented and discussed at the VII Scientific Conference "Applied Mathematics and Computing" AMC 2015 and published in the book of handouts, Prosvita , 2015, as well as 17th International Scientific and Technical conference SAIT 2015 and published in the "System Analysis and information Technologies" ESC " IASA " "KPI", 2015 .

Structure and content of the thesis. The master's thesis consists of the introduction, four chapters, conclusions and appendixes.

In the introduction were provided a general characterization of the existing methods and means adaptive traffic control systems, formulate goals and objectives of research.

In the first chapter the existing methods and tools for implementation the adaptive traffic control system were reviewed, their comparison is provided.

In the second chapter_the rationale of methods and means to construct the adaptive traffic model are presented.

In the third chapter created the main entity diagrams to define the unified adaptive traffic control model. The mathematical model for improving subsystem of control traffic signal was also proposed.

The fourth section_deals with the implementation of the software architecture of the modulation traffic control system in real-time and estimation the whole system in general.

In the conclusion the obtained results are analyzed.

The thesis contains 81 pages, 2 appendixes and 21 references to the used information sources. 23 drawings and 6 tables are also presented.

Keywords: adaptive control system, mathematical model, fuzzy logic, neuro-fuzzy approach.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	12
ВСТУП	13
1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ МОДЕЛЕЙ ТА ЗАСОБІВ РЕАЛІЗАЦІЇ АДАПТИВНИХ СИСТЕМ РЕГУЛЮВАННЯ ДОРОЖНІМ РУХОМ.....	15
1.1 Огляд існуючих моделей адаптивних систем регулювання дорожнім рухом	15
1.1.1 Метод адаптивного корегування сигналу світлофору з використанням нечіткої логіки.....	17
1.1.1 Нейронечіткий підхід для корегування сигналу світлофору	20
1.2 Порівняльний аналіз існуючих методів	22
Висновки до розділу	23
2 ОБГРУНТУВАННЯ ОБРАНИХ МЕТОДІВ ТА ЗАСОБІВ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ	24
2.1 Вдосконалення моделі адаптивної системи регулювання дорожнім рухом	24
2.2 Модель з урахуванням пріоритетності руху.....	25
2.3 Апарат нечіткої логіки для блоків керування системи.....	26
2.4 Контролер на базі нечіткої логіки.....	28
Висновки до розділу	32
3 РОЗРОБКА УНІФІКОВАНОЇ МОДЕЛІ АДАПТИВНОЇ СИСТЕМИ РЕГУЛЮВАННЯ ДОРОЖНЬОГО РУХУ	33
3.1 Суть і складові уніфікованої моделі	33
3.1.1 Діаграми компонентів адаптивної системи.....	34
3.1.2 Діаграма процесу нечіткого виведення	36

3.1.3	Діаграма класів моделі регульованого перехрестя.....	37
3.2	Математична модель підсистеми керування фазами руху з урахуванням пріоритетності напрямків руху.....	38
3.3	Оцінка адекватності моделі	42
	Висновки до розділу	46
4	ПРОГРАМНА РЕАЛІЗАЦІЯ УНІФІКОВАНОЇ МОДЕЛІ ФУНКЦІОНУВАННЯ ПЕРЕХРЕСТЯ В РЕАЛЬНОМУ ЧАСІ	47
4.1	Опис програмного забезпечення.....	47
4.2	Реалізація контролерів керуванням фазами руху.....	51
4.2.1	Контролер з фіксованим часом перемикання фаз руху	51
4.2.2	Нечіткий контролер	52
4.2.3	Нечіткий контролер з врахуванням пріоритетності руху	53
4.3	Інструкція користувача.....	55
	Висновки до розділу	58
	ВИСНОВКИ.....	59
	СПИСОК ЛІТЕРАТУРИ.....	61
	ДОДАТОК А.....	64
	ДОДАТОК Б	73

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

АСРДР – адаптивна система регулювання дорожнього руху

МАТЕМАТИЧНА МОДЕЛЬ — система математичних співвідношень, які описують досліджуваний процес або явище.

НЕЙРОНЕЧІТКА МЕРЕЖА – це подання системи нечіткого виведення у вигляді нейронної мережі, зручної для вивчення, поповнення, аналізу та використання [1].

НЕЧІТКА ЛОГІКА – наука, що була започаткована американським вченим азербайджанського походження Лотфі А. Заде (Lotfi A. Zadeh). На відміну від булевої алгебри, у котрій існує лише дві величини (0 та 1, правда чи неправда), у нечіткій логіці існують також перехідні величини (стани) [2].

ТЗ – транспортний засіб

ФАЗА РУХУ – час, від початку роботи поточних сигналів світлофора і до їх перемикання на наступні.

UML-МОДЕЛЬ – абстрактна модель системи, що створена за допомогою відкритого стандарту графічних позначень мови уніфікованої мови моделювання UML.

ВСТУП

Проблема моніторингу, контролю та регулювання дорожнього руху є дуже актуальною у зв'язку із постійним збільшенням кількості транспортних засобів на дорогах. Дорожній рух на регульованих ділянках руху в автоматичному режимі контролюється світлофором з фіксованим часом зміни сигналу або світлофором, режим перемикання якого змінюється в залежності від часу доби. Контролери даних світлофорів не є чутливими до постійних змін у дорожній ситуації на перехресті. Очевидним і основним недоліком такого варіанту є неможливість автоматично адаптуватись до дорожньої ситуації й змінювати час перемикання сигналу відповідно до дорожньої ситуації в поточний момент часу [3-4]. Таким чином, всі транспортні засоби для усіх напрямків руху мають фіксований час на перетин регульованих перехресть та одинаковий пріоритет на рух в дорожньо-транспортній мережі. В разі частих змін інтенсивності руху транспортних засобів, фіксований контроль призводить до накопичення в черзі великої кількості транспортних засобів, не оптимально розподіляючи час на рух відповідних напрямків. Це призводить до невимушено збільшення кількості транспортних засобів на окремих напрямках руху, збільшення середнього часу затримки транспортних засобів на перехрестях та створення заторів.

Останнім часом проведено багато досліджень даної проблеми. Все частіше на завантажених дорожнім рухом перехрестях починають використовуватися адаптивні контролери для світлофорів, які дозволяють працювати в режимі реального часу. Адаптивний контроль дорожнім рухом - стратегія управління рухом, в якій час роботи сигналу світлофору змінюється відповідно до дорожньої ситуації. Існують різні підходи для їх проектування

та реалізації, однак найбільший акцент досліджень був зроблений на підхід із застосуванням нечіткої логіки та нейронних мереж [5].

Тому в даній роботі були розглянуті моделі АСРДР, робота яких базується на використанні нечіткої логіки та нейронних мереж. Вони дозволяють збільшити пропускну спроможність перехрестя та зменшити час затримки ТЗ на перехресті. Для моніторингу поточної ситуації на перехресті мають використовуватися дані із встановлених камер спостереження, аналіз яких дозволяє регулювати рух на перехресті в режимі реального часу.



1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ МОДЕЛЕЙ ТА ЗАСОБІВ РЕАЛІЗАЦІЇ АДАПТИВНИХ СИСТЕМ РЕГУЛЮВАННЯ ДОРОЖНІМ РУХОМ

1.1 Огляд існуючих моделей адаптивних систем регулювання дорожнім рухом

Нагальною проблемою багатьох сучасних міст є оптимальне керування дорожнім рухом. Зменшення часу, який витрачається щоб дістатися з однієї точки міста в іншу, дозволить поліпшити роботу міських транспортних сервісів та життя кожного окремого мешканця міста [6].

Адаптивна система регулювання дорожнім рухом - це система управління дорожнім рухом з центрально-розділеним інтелектом, яка складається з наступних елементів:

- центральний пункт управління (ЦПУ)
- точки адаптивного управління дорожнім рухом, обладнаних інтелектуальними контролерами і детекторами транспорту

Детектори транспорту забезпечують локальне адаптивне управління найбільш складними і важливими перетинами і ділянками транспортної мережі, інформаційну взаємодію з ЦПУ та системних детекторів і контролерів, керованих з ЦПУ постійно або періодично, які повідомляють в ЦПУ відомості про транспортні потоки.

Основними завданнями АСРДР є:

1. Забезпечення безпеки та комфорності дорожнього руху
2. Реалізація максимальної пропускної здатності дорожньої мережі
3. Підвищення техніко-економічної ефективності управління дорожнім рухом

4. Підвищення ефективності використання вже встановленого апаратного забезпечення
5. Скорочення викидів шкідливих речовин в атмосферу

Для виконання даних завдань постає необхідність забезпечення злагодженого управління всіма елементами системи.

Сучасна модель адаптивної системи регулювання дорожнього руху складається з двох основних підсистем: активної та пасивної (Рис. 1.1) [7].



Рисунок 1.1 – Організація адаптивного регулювання дорожнього руху

Активна підсистема виконує безпосереднє регулювання дорожнього руху за рахунок головного контролера керування фазами руху. Головний контролер регулює рух через програмний інтерфейс переключення сигналів світлофора [8].

Пасивна підсистема виконує аналіз даних отриманих детектором з перехрестя для подальшого вдосконалення роботи адаптивного контролера.

Основними параметрами, які характеризують роботу системи регулювання дорожнім рухом є середня кількість транспортних засобів, які простоюють на окремому перехресті в очікуванні дозволяючого сигналу та середній час затримки на перехресті кожного з них [8].

Оптимальні параметри дорожнього руху в транспортній мережі представляють собою сукупність умов, найбільш сприятливих для руху транспортних засобів чи умов для правильного протікання технологічного процесу. Адаптивне управління перехрестям дозволяє це забезпечити, завдяки своєчасній зміні режимів роботи світлофорного об'єкта в реальному часі в залежності від реальних транспортних запитів.

1.1.1 Метод адаптивного корегування сигналу світлофору з використанням нечіткої логіки

В [9] було запропоновано метод побудови АСРДР з виростанням нечіткої логіки. Схематично принцип роботи контролера, що використовує цей методу зображений на рисунку 1.1.

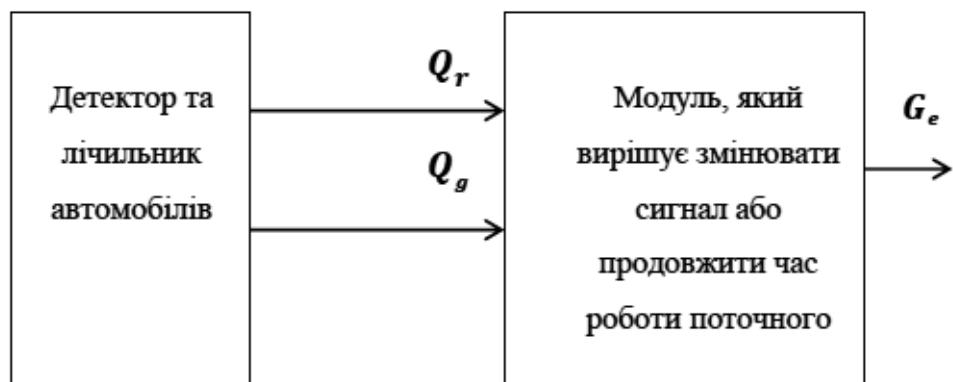
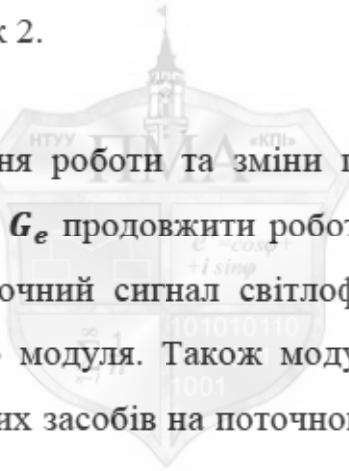


Рисунок. 1.2 – Схема принципу роботи методу

Нижче наведено алгоритм роботи контролера.

1. Виставити мінімальний час роботи зеленого сигналу світлофора відповідно до поточної дорожньої ситуації.
2. Встановити мінімальний час роботи зеленого сигналу для поточних напрямків з сигналом світлофора, що дозволяє рух транспортним засобам.
3. Визначити на який час можна продовжити роботу поточного зеленого сигналу світлофора, використовуючи нечітке прогнозування та базуючись на кількість транспортних засобів в черзі на поточний та наступний сигнал світлофора, що дозволяє рух.
4. Ввімкнути зелений сигнал світлофора для наступного напрямку.

Перейти на крок 2.

Модуль подовження роботи та зміни поточного сигналу світлофора вирішує на скільки часу G_e продовжити роботу поточного зеленого сигналу світлофору. Змінює поточний сигнал світлофору, базуючись на параметрі «важливості» з першого модуля. Також модуль отримує на вхід параметр Q_g – черга транспортних засобів на поточному зеленому напрямку та Q_r – черга транспортних засобів на наступний зелений напрямок.

Значення параметрів Q_g та Q_r лежать в інтервалі $[0; 30]$ та поділене на наступні нечіткі підмножини: {дуже мала черга - VS , мала черга - S , середня черга - M , велика черга - B , дуже велика черга - VB }.

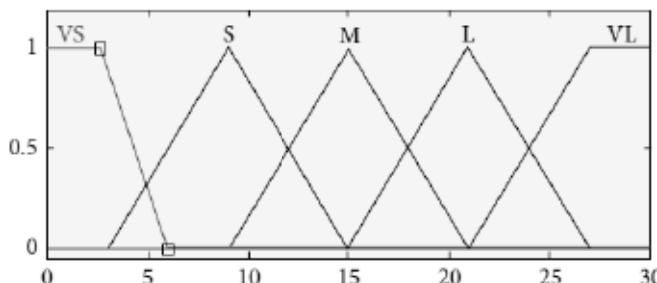


Рисунок 1.3 – Функції належності для Q_g та Q_r

Значення параметру G_e лежить в інтервалі $[0; 25]$ та поділено на наступні нечіткі підмножини: {дуже мало часу - VS, мало часу - S, середньо часу - M, багато часу - L, дуже багато часу - VL}.

На рисунку 1.4 та в таблиці 1.1 представлено відповідно функцію належності та нечіткі правила для визначення параметру затримки. Наприклад: якщо $\{Q_g = VS\}$ і $\{Q_r = L\}$, тоді $\{G_e = VS\}$.

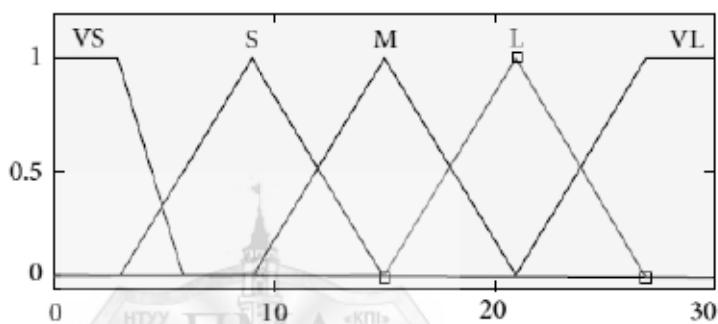


Рисунок 1.4 - Функція належності для визначення параметру G_e

Таблиця 1.1 – Нечіткі правила для визначення параметру G_e

$Q_r \backslash Q_g$	VS	S	M	L	VL
VS	VS	VS	VS	VS	VS
S	S	S	S	VS	S
M	M	M	M	M	S
B	L	L	L	M	M
VB	VL	VL	L	L	L

Інтервали, на яких визначені нечіткі множини, можуть бути налаштовані індивідуально під особливості перехреся перед початком роботи контролера.

1.1.1 Нейронечіткий підхід для корегування сигналу світлофору

В роботі [10] в якості для корегування сигналу світлофору використовується нейронечітка мережа. Мережа складається з 5 шарів (Рис. 1.8). На вхід подається три параметри: q_g – черга автомобілів на зеленому сигналі світлофора, q_r – черга автомобілів на наступному зеленому сигналі світлофора та Ar – інтенсивність руху транспортних засобів на напрямку з зеленим сигналом світлофору. Вихідний параметр нейронної мережі Δt – час затримки поточного сигналу світлофору.

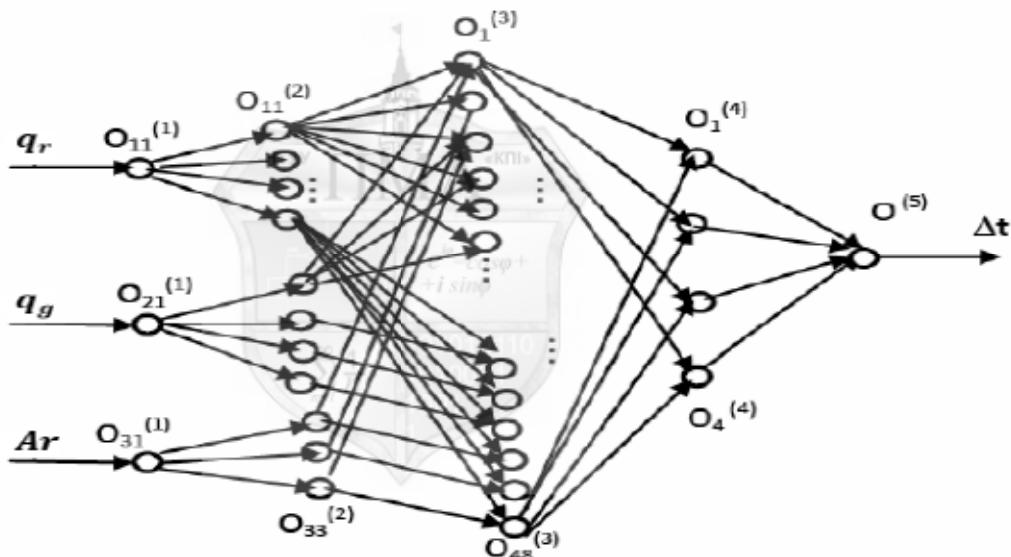


Рисунок 1.5 – Нейронечітка мережа

Перший шар нейронної мережі представляє собою чіткі вхідні значення змінних. На виході першого шару наступні значення: $O_{11}^{(1)} = qr_i$, $O_{21}^{(1)} = qg_i$, $O_{31}^{(1)} = Ar_i$.

Другий шар – шар функцій належності. Кожен нейрон цього шару представляє собою функцію належності для вхідного параметру. Вихідне значення для кожного нейрону цього шару лежить в інтервалі [0, 1] та дорівнює:

$$O_{jk}^{(2)} = e^{-\frac{(o_{jk}^{(1)} - a_{jk})^2}{b_{jk}^2}},$$

де $j = 1, 2; k = 1, 2, \dots, 4$ та для $j=3; k=1, 2, 3$; a_{jk} та b_{jk} параметри, які використовуються в методі зворотнього поширення помилки.

Третій шар – базується на застосування нечітких правил. Вихідними значенням нейронів третього шару будуть:

$$O_p^{(3)} = \min(\mu q_r, \mu q_g, \mu Ar),$$

де p – номер нечіткого правила.

Четвертий шар – акумуляція змінних.

$$O_j^{(4)} = \max(\alpha_p \cdot O_p^{(3)}), \quad j=1, \dots, 4.$$

П'ятий шар – дефазифікація змінних. Вихідне значення:

$$O_j^{(5)} = \frac{\sum_{j=1}^4 (O_j^{(4)} a_j b_j)}{\sum_{j=1}^4 (O_j^{(4)} a_j)}$$

В якості алгоритму для навчання було обрано метод градієнтного спуску. $\frac{(t_i - O^{(5)})^2}{2}$ – значення помилки, яку необхідно мінімізувати.

1.2 Порівняльний аналіз існуючих методів

Кожний з вище наведених адаптивних методів керування показує значно кращі результати [9, 10] у порівнянні з методами неадаптивного керування дорожнім рухом в транспортній мережі. Також, моделі АСРДР залишають перспективи подальшого їх вдосконалення та оптимізації, на відміну від моделей неадаптивних систем. Єдиним недоліком адаптивних методів регулювання є необхідність встановлення додаткової апаратури та датчиків в транспортній мережі. Тому є актуальним використовувати дані методи лише на окремо визначених перехрестях.

Порівнюючи вище наведені методи між собою, було отримано різні результати для різних ступенів завантаженості перехрестя.

Для випадку слабкої та середньої завантаженості перехрестя найкращі результати показав метод, який застосовує нечіткий підхід, для високої - нейронечіткий.

Таким чином, вибір підходу для побудови адаптивної системи регулювання дорожнього руху залежить від характеристик та параметрів перехрестя. Також необхідно відмітити, що підхід з використанням нейронної мережі вимагає довшого попереднього налаштування та більш складний у технічній підтримці [11].

В даній роботі, акцент зроблений на перехрестя з малим та середнім ступенями завантаженості, тому для подальшого дослідження було обрано моделі адаптивного регулювання з контролером керування на базі нечіткої логіки.

Висновки до розділу

У розділі проведено огляд існуючих методів для реалізації та оцінки роботи адаптивних систем регулювання дорожнього руху, розглянуто два основні підходи для побудови підсистем керування фазами руху, проаналізовано літературні джерела [1-10].

Встановлено:

1. Сучасні адаптивні системи організаційно складаються з двох основних частин: активної та пасивної. Активна частина безпосередньо відповідає за регулювання дорожнього руху, пасивна – за оптимізацію та вдосконалення даної системи.
2. Існує два основних підходи для побудови підсистеми керування фазами руху: нечіткий та нейронечіткий

2 ОБГРУНТУВАННЯ ОБРАНИХ МЕТОДІВ ТА ЗАСОБІВ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Вдосконалення моделі адаптивної системи регулювання дорожнім рухом

На основі досліджень існуючих моделей адаптивних систем було встановлено, що їх застосування дає значно кращі результати в регулюванні дорожнього руху на перехресті ніж звичайні, з фіксованим часом перемиканням. Однак, на відміну від моделей не адаптивних систем, адаптивні можна намагатися оптимізувати, завдяки урахуванню більшої кількості дорожніх характеристик. Дослідивши два основних підходи для побудови моделей АСРДР, було встановлено, що вони не враховують пріоритетність перемикання зеленого сигналу, що дозволяє окремим напрямкам почати рух. Тобто, послідовність руху всіх напрямків на перехресті фіксовано визначена та не залежить від дорожньої ситуації. В [9] також підкреслюється відсутність врахування пріоритетності руху в існуючих АСРДР.

В роботі пропонується побудувати вдосконалену модель АДСРД з використанням нечіткої логіки, контролер керування фазами якої враховує поточну ситуацію на перехресті для визначення пріоритетності його напрямків та порівняти дану систему з існуючими. Також, запропонована модель контролера використовує значення параметру пріоритетності руху, не тільки для визначення напрямку, для якого необхідно увімкнути дозволяючий рух сигнал наступним, але й для визначення часу подовження роботи поточного сигналу світлофора. Такий підхід, зменшить кількість запитів від контролера до детектора та підвищить ефективність роботи системи в цілому.

Для вдосконалення було обрано нечіткий підхід, бо в порівнянні з нейро-нечітким моделлю не вимагає час на навчання, тому система швидша у налаштуванні та простіша у підтримці.

2.2 Модель з урахуванням пріоритетності руху

Для визначення пріоритетності напрямків руху та їх врахування під час обчислення затримки поточного сигналу світлофору, необхідно інтегрувати додатковий модуль в дослідженню моделі з нейро-нечітким підходом.

Схематично модель адаптивної системи регулювання дорожнього руху представлена на рисунку 2.1.



Рисунок 2.1 Схема принципу керування перемиканням сигналу світлофора

Таким чином, в даній моделі кожний модуль представляє собою систему нечіткого логічного виведення. Перший модуль визначає пріоритетність напрямків руху. Другий модуль отримує на вхід значення пріоритетності та

дані з детекторів. На основі цих даних визначається час затримки поточного сигналу.

2.3 Апарат нечіткої логіки для блоків керування системи

Нечітка логіка – розділ математики, який є узагальненням класичної логіки і теорії множин, та базується на понятті нечіткої множини, вперше введеної Лотфі Заде у 1965 році як об'єкта з функцією належності елемента до множини, яка приймає будь-які значення в інтервалі $[0, 1]$, а не тільки 0 або 1. Нечітка логіка є розширенням для роботи з концептом часткової правди, де змінна правда може набувати значення між повна неправда та абсолютна правда [12].

Нечітка множина A – це сукупність впорядкованих пар, складених з елементів x універсальної множини X і відповідних ступеней належності $\mu_A(x)$. $A = \{(x, \mu_A(x)) \mid x \in X\}$, де $\mu_A(X)$ – функція належності, яка вказує в якій ступені(мірі) елемент x належить множині A . Функція $\mu_A(X)$ приймає значення в деякій лінійно впорядкованій множині M . Множину M називають множиною належності, часто в якості M обирається відрізок $[0, 1]$. Якщо $M = \{0, 1\}$ (тобто складається тільки з двох елементів), то нечітка множина може розглядатися як звичайна, чітка множина.

Функція належності нечіткої множини – узагальнення характеристичної функції класичної множини. В термінах нечіткої логіки вона представляє собою ступінь належності кожного члена простору X до даної нечіткої множини. Функція належності $\mu_A(X)$ кількісно градуює належність елементів фундаментальної множини простору $x \in X$ нечіткої множини A . Значення 0

показує, що елемент не належить нечіткій множині, 1 – належить. Значення між 0 та 1 характеризують елементи, які «нечітко» належать множині.



Рисунок 2.2 – Приклад функції належності

Нечітка логіка найчастіше працює з лінгвістичним змінними, які використовуються для полегшення формування правил та фактів для задачі. Лінгвістична зміна – це зміна, яка може приймати значення у формі та точності, які є достатніми для вирішення визначеної задачі. Це є вагомою відмінністю від розв'язання задач методами класичної математики, в якій зміні, зазвичай, приймають числові значення. Наприклад, така лінгвістична зміна як вік, може приймати значення молодий та старий.

Нечітка логіка є багатозначною логікою, що дозволяє визначити проміжні значення для таких загальноприйнятих оцінок, як так | ні, істинно | хибно, чорне | біле і т.п. Вирази подібні таким, як повільно або досить швидко можливо формулювати математично і обробляти на комп'ютерах [13].

Основою для проведення операції нечіткого логічного висновку є база правил, яка містить нечіткі висловлювання у формі "ЯКЩО – ТО" і функції

принадлежності для відповідних лінгвістичних термів. При цьому повинні виконуватися наступні умови:

У загальному випадку механізм логічного висновку включає три етапи: введення нечіткості (фазифікація), композиція і приведення до чіткості, або дефазифікації (Рис. 2.1) [14].



Рисунок 2.3 – Система нечіткого логічного виведення

Контролери на нечіткій логіці – найважливіший додаток теорії нечітких множин. Їх функціонування трохи відрізняється від роботи звичайних контролерів; для опису системи використовуються знання експертів замість диференціальних рівнянь. Ці знання можуть бути виражені природною чином за допомогою лінгвістичних змінних, які описуються нечіткими множинами.

2.4 Контролер на базі нечіткої логіки

Для роботи з контролером на базі нечіткої логіки потрібно [15]:

1. Визначити вхідні лінгвістичні змінні

2. Визначити лінгвістичну зміну, яку необхідно отримати на виході
3. Задати правила визначення результуючої змінної з вхідних

Математична складова роботи контролера.

Основу роботи контролера складає система нечіткого виведення. Зазвичай, щоб отримати значення результуючої змінної в системі нечіткого виведення, виконуються наступні етапи [16]:

- Фазифікація;
- Композиція;
- Дефазифікація;

Фазифікація. На етапі фазифікації виконується процес знаходження значень функції належності нечітких множин на основі звичайних (не нечітких) вхідних даних. Метою фазифікації є визначення відповідності між конкретною вхідною змінною системи нечіткого виведення та значенням функції належності, що відповідає терму вхідної лінгвістичної змінної. Після завершення даного етапу для всіх вхідних змінних мають бути визначені конкретні значення функції належності по кожному з лінгвістичних термів, які використовуються у підумовах бази правил системи нечіткого виведення.

Етап композиції часто розбивають на два етапи: агрегація та акумуляція.

Агрегація. Агрегація представляє собою процедуру визначення істинності належності умов по кожному з правил системи нечіткого виведення. При цьому використовуються отримані на етапі фазифікації значення функцій належності термів лінгвістичних змінних.

Якщо умова нечіткого правила є простим нечітким виразом, тоді ступінь його істинності відповідає значенню функції належності відповідного терма лінгвістичної змінної.

Якщо умова являє собою складний вираз, тоді ступінь істинності складного виразу визначається на основі відомих значень істинності елементарних виразів, які його складають, за допомогою введених раніше

нечітких логічних операцій. При цьому для визначення результату кон'юнкції або логічного «і» може бути використана наступна формула:

$$T(A \cap B) = \min\{T(A) \cap T(B)\},$$

а для визначення результату нечіткої диз'юнкції чи логічного «або» може бути використана формула:

$$T(A \cup B) = \max\{T(A) \cup T(B)\}.$$

Активізація. Активізація в системі нечіткого виведення – це процедура чи процес знаходження ступені істинності кожного елементарного логічного виразу. Якщо нечітке правило є простим виразом, то ступінь його істинності дорівнює добутку вагового коефіцієнту та ступені істинності умови даного нечіткого правила. Якщо заключення представляє собою складений вираз, то ступінь істинності кожного з елементарних виразів дорівнює декартовому добутку вагового коефіцієнта та ступені істинності умови даного нечіткого правила. Якщо вагові коефіцієнти не вказані явно на етапі формування правил, то їх значення по замовчуванню дорівнюють одиниці.

Акумуляція. Акумулювання – це процес знаходження функції належності дляожної з вихідних лінгвістичних змінних. Ціллю етапу акумуляції є об'єднання всіх ступенів істинності підзаключень для визначення функції належності кожної з вихідних змінних.

Дефазифікація. Дефазифікація в системах нечіткого виведення – це процес переходу від функції належності вихідної лінгвістичної змінної до її чіткого(числового) значення. Мета дефазифікації полягає в тому, щоб використовуючи результати акумуляції всіх вихідних лінгвістичних змінних, отримати кількісне значення кожної із вихідних змінних. Це значення є результатом роботи системи нечіткого виведення та контролеру в цілому.

Для виконання чисельних розрахунків на етапі дефазифікації можуть бути використані наступні методи:

Метод центру ваги. Цей метод зведення до чіткості називається центроїдним та розраховується по наступній формулі:

$$y = \frac{\int_{\min}^{\max} x \mu(x) dx}{\int_{\min}^{\max} \mu(x) dx},$$

де y – результат дефазифікації, x – зміна, яка відповідає вихідній лінгвістичній змінній та приймає значення від $x = \min$ до $x = \max$. \min та \max – ліва та права точки інтервалу значень змінної x , $\mu(x)$ – функція належності.

Метод бісектриси площини. Даний метод зведення до чіткості визначаються з наступного рівняння:

$$\int_{\min}^u \mu(x) dx = \int_u^{\max} \mu(x) dx,$$

де бісектриса площини $y = u$.

Метод лівого модального значення. Ліве модальне значення визначається по формулі:

$$y = \min\{x_m\},$$

де x_m – модальне значення нечіткої множини.

Метод правого модального значення. Праве модальне значення визначається по формулі:

$$y = \max\{x_m\},$$

де x_m – модальне значення нечіткої множини.

В даній роботі в якості методу дефазифікації використовується метод центру ваги та використано в усіх математичних моделях, які базуються на використані нечіткої логіки.

Висновки до розділу

У розділі обрано методи та засоби для вирішення задачі створення моделі адаптивної системи регулювання дорожнього руху.

Розглянуто наступні питання:

- вдосконалення існуючого нечіткого підходу для побудови підсистеми керування фазами завдяки інтегрування додаткового модулю визначення пріоритетності напрямків руху
- алгоритми для реалізації системи нечіткого виведення

3 РОЗРОБКА УНІФІКОВАНОЇ МОДЕЛІ АДАПТИВНОЇ СИСТЕМИ РЕГУЛЮВАННЯ ДОРОЖНЬОГО РУХУ

3.1 Суть і складові уніфікованої моделі

Уніфікована модель складається з набору діаграм:

- класів;
- діяльності;
- пакетів;
- компонентів;

Діаграма класів - статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм [17, 18]. Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

Діаграма діяльності - візуальне представлення графу діяльності. Граф діяльності є різновидом графу станів скінченного автомата, вершинами якого є певні дії, а переходи відбуваються по завершенню дій [19].

Діаграма пакетів - діаграма, що відображає залежності між пакетами, що утворюють модель.

Діаграма компонентів - відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись. Модуль програмного

забезпечення може бути представлено в якості компоненти. Деякі компоненти існують під час компіляції, деякі — під час компонування, а деякі під час роботи програми. Діаграма компонент відображає лише структурні характеристики [19].

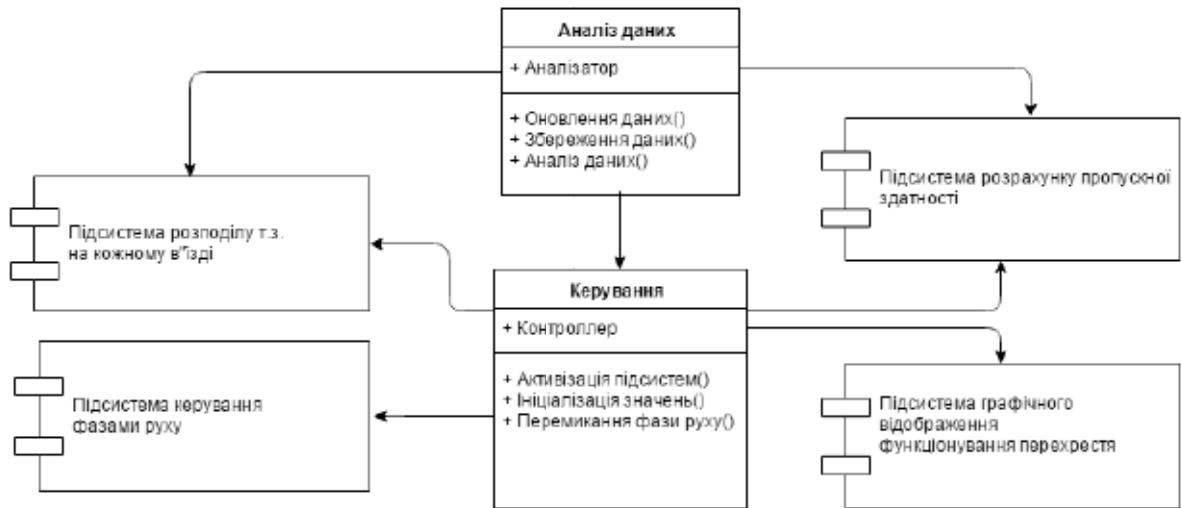
В наступних підрозділах представлена основні діаграми для побудови уніфікованої моделі адаптивної системи регулювання дорожнім, які відображають її структурне представлення.

3.1.1 Діаграми компонентів адаптивної системи

Діаграма компонентів показує розбиття програмної системи на структурні компоненти та зв'язку (залежності) між компонентами. Компоненти зв'язуються через залежності, коли з'єднується необхідний інтерфейс одного компонента з наявним інтерфейсом іншого компонента. Таким чином ілюструються відносини клієнт–вихідні дані між двома компонентами.

Залежність показує, що один компонент надає сервіс, необхідний іншого компоненту. Залежність зображується стрілкою від інтерфейсу або порту клієнта до імпортованого інтерфейсу.

На рисунку 3.1 представлена реалізація діаграми компонентів системи для моделі АСРДР [20].



Рисинок 3.1 – Діаграма компонентів адаптивної системи у нотації UML

Основним компонентом діаграми є підсистема керування, яка здійснює контроль за усіма іншими підсистемами і відповідає за активізацію, ініціалізацію та взаємообмін даними між іншими підсистемами, для отримання необхідних параметрів керування дорожнім рухом.

Підсистема аналізу даних здійснює функцію накопичення та аналізу даних про поточну ситуацію на перехресті. Періодично оновлює дані та сповіщає про це підсистему керування.

Підсистема розподілу транспортних засобів на кожному в'їзді відповідає за моделювання транспортного руху на перехресті. В залежності від параметрів налаштувань може моделювати різні ступені завантаженості ділянки транспортного руху.

Результатом роботи підсистеми керування фазами руху є подовження чи зупинка роботи поточного сигналу світлофору, а також будь-які дії пов'язані з безпосереднім керуванням роботою світлофора.

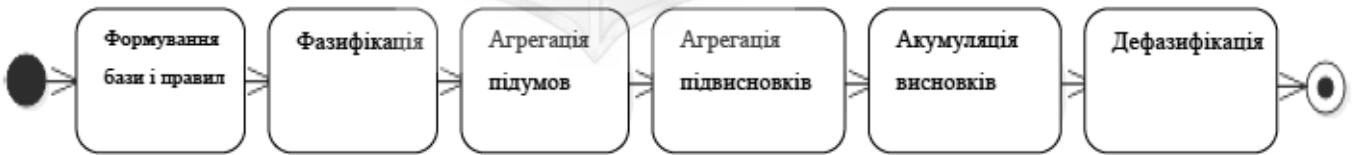
Підсистема розрахунку пропускної здатності здійснює моніторинг поточної ситуації на ділянці дорожнього руху та обчислює основні параметри для характеристики ситуації на перехресті.

Підсистема графічного відображення функціонування перехрестя дає можливість побудови візуального представлення поточної ситуації на ділянці руху. Це дозволяє проводити моніторинг та візуально оцінювати дорожню обстановку.

3.1.2 Діаграма процесу нечіткого виведення

В роботі досліджується нечіткий підхід, в якості основного для адаптивного регулювання дорожнього руху. В основі підходу лежить алгоритм нечіткого виведення.

Даний алгоритм описує кілька етапів, що виконується послідовно. При цьому кожний наступний етап отримує на вхід значення отримані на попередньому кроці. Діаграма діяльності алгоритму представлена на рисунку 3.3.



Рисинок 3.2 – Діаграма діяльності процесу нечіткого виведення у нотації UML

Алгоритм відрізняється тим, що він працює за принципом «чорного ящику» [21]. На вході і на виході мають місце кількісні значення. На проміжних етапах використовується апарат нечіткої логіки і теорія нечітких множин. У цьому і полягає елегантність використання нечітких алгоритмів для підсистем керування. Можна маніпулювати звичними числовими даними, але

при цьому використовувати гнучкі можливості, які надають системи нечіткого виведення.

3.1.3 Діаграма класів моделі регульованого перехрестя

На рисунку 3.3 представлена діаграма класів, яка показує зв'язки між основними сутностями системи регулювання дорожнього руху на перехресті.

Головними задачами при проектуванні були майбутня гнучкість, розширеність та легкість у підтримці системи. Двома основними сутностями моделі є класи «Перехрестя» (Interesection) та «Контролер» (Controller).

Клас «Контролер» здійснює централізоване керування усіма іншими класами. Містить в собі основні параметри для контролю перехрестям. Для створення специфічного контролера для дослідження (наприклад з використанням нечіткої логіки) необхідно створити новий клас та унаслідуватись від базового класу «Контролер».

Клас «Перехрестя» дозволяє створювати та ініціалізувати перехрестя довільної структури, з вказаною кількістю напрямків та фаз руху. Це дозволяють досліджувати та аналізувати дорожню ситуацію на будь-якій ділянці руху дорожньої мережі.

Клас «Ступінь завантаженості» (ArrivalRate) дозволяє регулювати швидкістю прибуття транспортних засобів. Таким чином, можна створювати різні ступені завантаженості дорожнього руху.

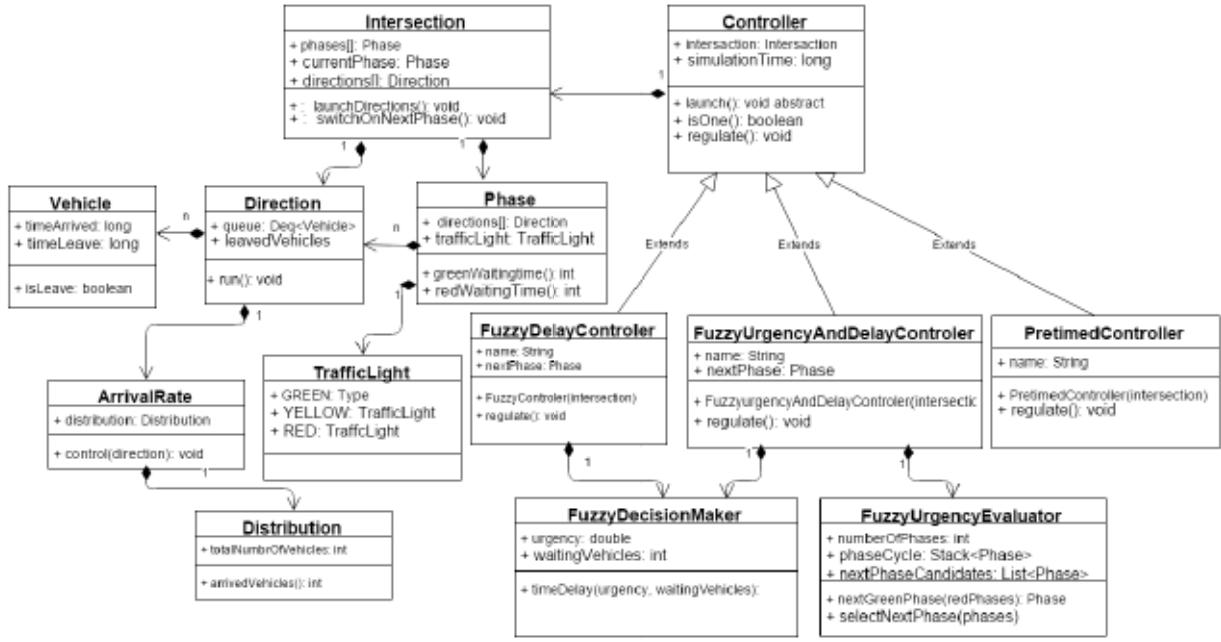


Рисунок 3.3 – Діаграма класів підсистеми керування у нотації

3.2 Математична модель підсистеми керування фазами руху з урахуванням пріоритетності напрямків руху

Модульна схема керування фазами руху на перехресті з урахуванням пріоритетності напрямків руху зображена на рисунку 1.4.

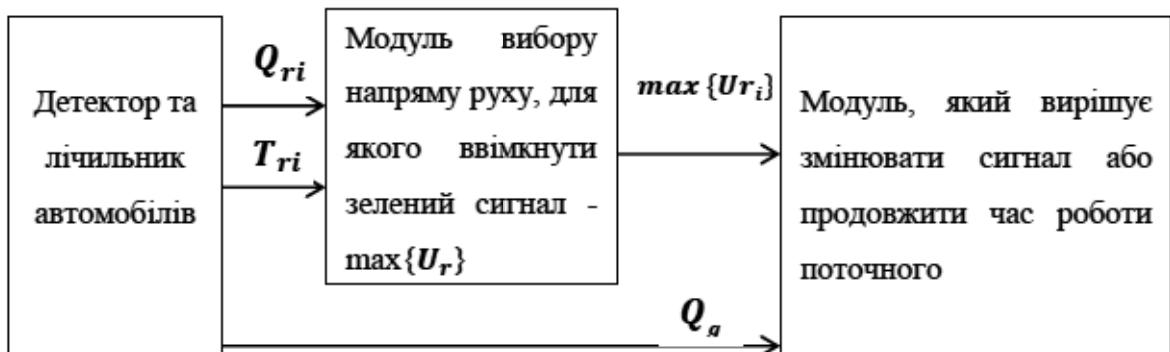


Рисунок 3.4 – Схема принципу керування перемиканням сигналу

Алгоритм роботи контролера на базі даної математичної моделі.

1. Виставити мінімальний час роботи зеленого сигналу світлофора відповідно до поточної дорожньої ситуації.
2. Встановити мінімальний час роботи зеленого сигналу для поточних напрямків з сигналом світлофора, що дозволяє рух транспортним засобам.
3. Визначити параметр «важливості» руху U_r для всіх напрямків з сигналом, що забороняє рух транспортних засобів, використовуючи функції належності на рисунках 3 та 4 і нечіткі правила з таблиці 1.
4. Обрати $\max\{U_r\}$.
5. Визначити на який час можна продовжити роботу поточного зеленого сигналу світлофора, використовуючи нечітке прогнозування та базуючись на кількості транспортних засобів в черзі на поточний та наступний сигнал світлофора, що дозволяє рух.
6. Увімкнути зелений сигнал світлофора для наступного напрямку.

Перейти на крок 2.

Зі схеми принципу керування та алгоритму можна побачити, що даний метод, не тільки корегує тривалість роботи сигналу світлофора, а також визначає пріоритет вмикання дозволяючого сигналу для всіх напрямків руху. Для цього було введено параметр «важливості» напрямку. Параметр «важливість» показує, для якого напрямку зелене світло має бути увімкненим наступним.

Контролер з нечіткою логікою для керування дорожнім рухом, в основі роботи якого лежить запропонований метод складається з двох модулів.

Модуль вибору напряму руху, для якого ввімкнути зелене світло визначає параметр «важливості» руху U_r для всіх напрямків з сигналом, що забороняє рух транспортних засобів. На вхід модуль отримує два параметри: Q_{ri} – черга

в кожній червоній фазі та T_{ri} – час, впродовж якого «горіло» червоне світло в кожній фазі.

Значення параметру Q_{ri} лежить в інтервалі $[0; 30]$ та поділене на наступні нечітки підмножини: {дуже мала черга - VS , мала черга - S , середня черга - M , велика черга - B , дуже велика черга - VB } (Рис. 3.5).

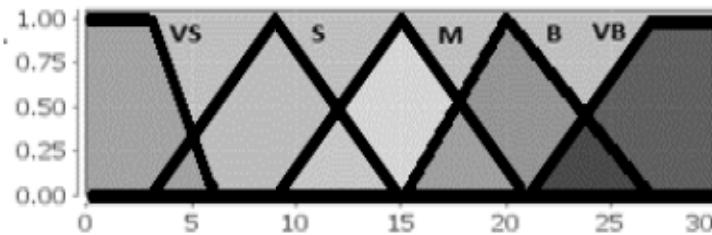


Рисунок 3.5 – Функція належності для Q_{ri}

Значення параметру T_{ri} лежить в інтервалі $[0; 150]$ та поділене на наступні нечітки підмножини: {дуже мало часу - VS , мало часу - S , середньо часу - M , багато часу - L , дуже багато часу - VL } (Рис. 3.6).

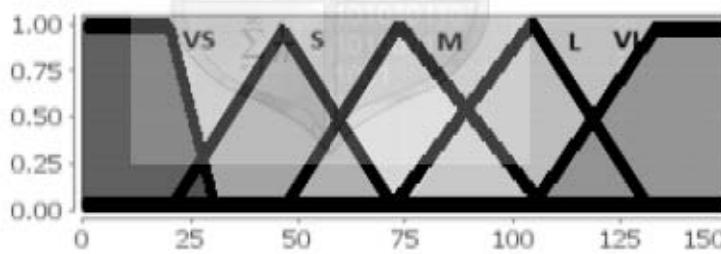


Рисунок 3.6 – Функція належності для T_{ri}

Значення параметру U_r лежить в інтервалі $[0; 1]$ та поділене на наступні нечітки підмножини: {дуже мала важливість - VS , мала важливість - S , середня важливість - M , велика важливість - B , дуже велика важливість - VB }.

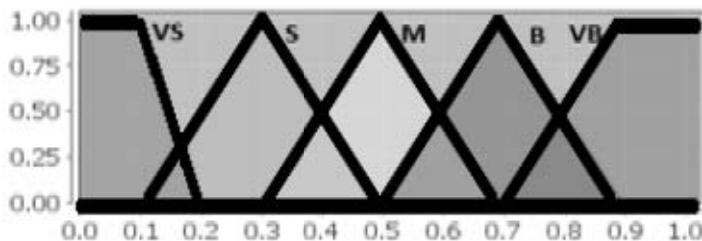


Рисунок 3.7. – Функція належності для параметру U_r

В таблиці 1.2 представлено нечіткі правила для визначення «важливості» кожного з напрямків на перехресті. Наприклад: якщо черга дуже мала $\{Q_{ri} = VS\}$ та час, впродовж якого ввімкнений сигнал світлофора, забороняючий рух, – великий $\{T_{ri} = L\}$, то параметр «важливості» малий $\{U_{ri} = S\}$.

Таблиця 3.1 – Нечіткі правила для визначення параметру U_{ri}

T_{ri} Q_{ri}	VS	S	M	L	VL
VS	VS	VS	S	S	M
S	VS	S	$\frac{c}{c + s}$	S	M
M	M	S	M	M	B
B	B	M	M	B	VB
VB	B	M	B	VB	VB

Вихідний параметр даного модуля - це максимальне значення параметра U_r серед усіх напрямків руху. Це значення вказує напрям руху, для якого треба ввімкнути зелений сигнал.

Робота модуля подовження роботи та зміни поточного сигналу світлофора аналогічна розглянутому в розділі 1.1.1.

3.3 Оцінка адекватності моделі

В ході використання запропонованої моделі виконані розрахунки основних параметрів оцінки завантаженості дорожньої мережі. Також проведено порівняння з існуючими моделями АСРДР та з моделлю регулювання руху, де час перемикання сигналів світлофору є фіксованим. В якості дорожньої мережі для тестування було обрано чотирьох напрямлене перехрестя із дозволеними поворотами ліворуч (Рис. 3.8).

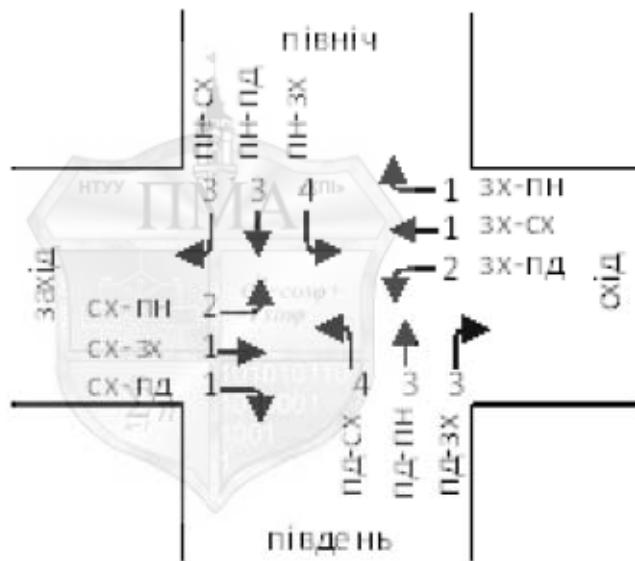


Рисунок 3.8. – Регульоване перехрестя з 4-ма фазами руху

Розрахунки основних параметрів та порівняльний аналіз було проведено для ступенів завантаженості руху на перехресті: слабкий, середній, високий. Також, був проведений аналіз для випадку, коли фази руху на перехресті мають різні ступені завантаження. В якості параметрів оцінки пропускної спроможності перехрестя розглядалися середня кількість автомобілів, яка знаходиться в очікувані своєї черги руху D та середній час затримки окремого транспортного засобу T на перехресті.

Для перевірки адекватності моделі, було проведено 10 моделювань для кожного типу контролера.

Тривалість кожного моделювання 1 година.

Для моделювання кількості транспортних засобів на в'їздах перехрестя використовувався розподіл Пуасона:

$$P(x) = \frac{(\lambda t)^x e^{-\lambda t}}{x!},$$

де $P(x)$ – ймовірність появи x транспортних засобів за час t , λ – середня кількість появи ТЗ за одиницю часу.

На рисунках нижче зображені графіки залежності параметру D від часу для кожного з контролерів.

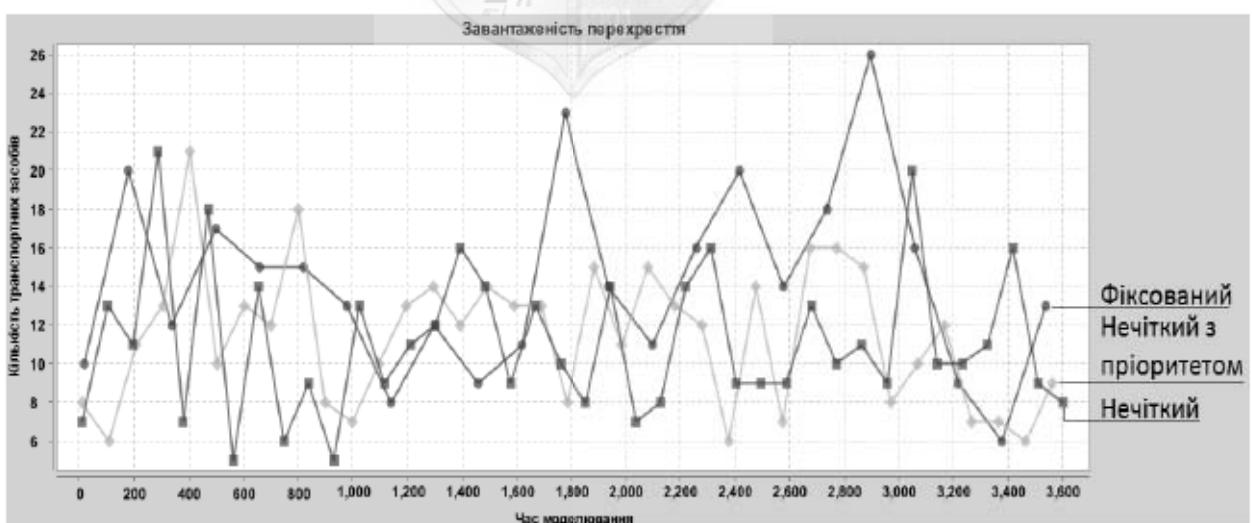
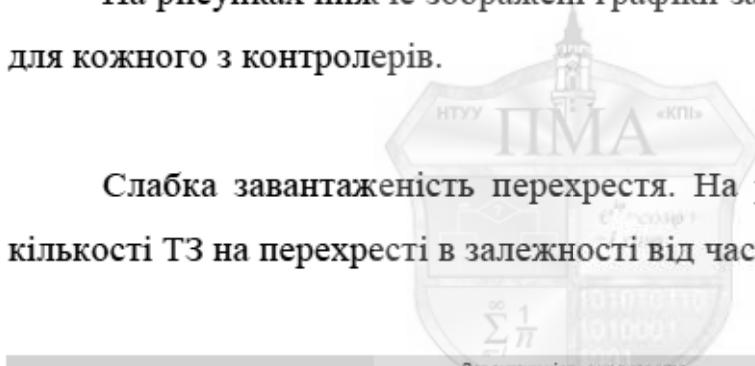


Рисунок 3.9 – Порівняльний аналіз, слабка завантаженість перехрестя

Таблиця 3.2 – Порівняльна таблиця, слабка завантаженість перехрестя

Контролер	Середня к-сть ТЗ на перехресті	Середній час очікування
Фіксований	17.4	27.1
Нечіткий	13.3	22.6
Нечіткий з пріоритетністю руху	11.9	18.6

Середня завантаженість перехрестя. На рисунку 3.10 зображене графік кількості ТЗ на перехресті в залежності від часу.

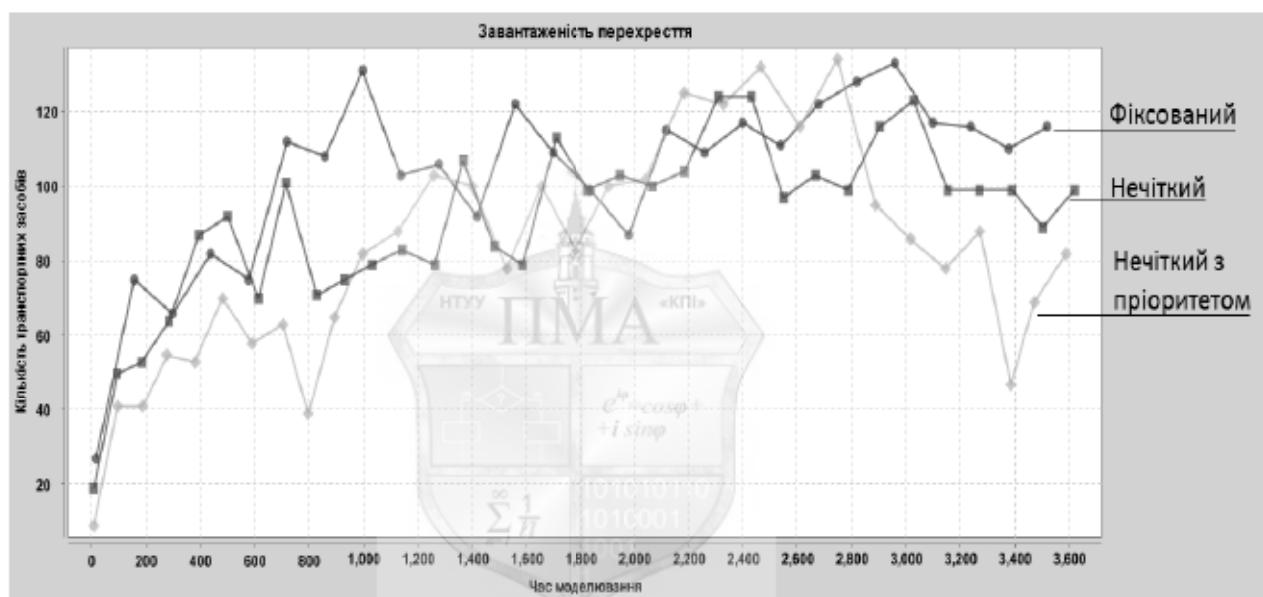


Рисунок 3.10 – Порівняльний аналіз, середня завантаженість перехрестя

Таблиця 3.3 – Порівняльна таблиця, середня завантаженість перехрестя

Контролер	Середня к-сть ТЗ на перехресті	Середній час очікування
Фіксований	95.6	70.1
Нечіткий	91.9	65.1
Нечіткий з пріоритетністю руху	85.8	53.6

Висока завантаженість перехрестя. На рисунку 3.11 зображене графік кількості ТЗ на перехресті в залежності від часу.

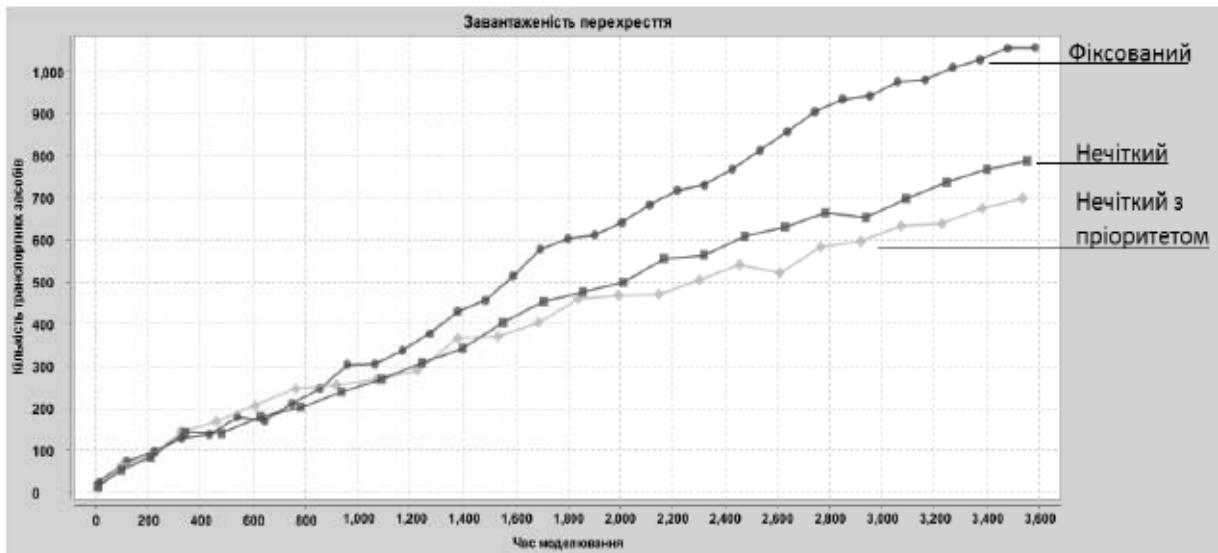


Рисунок 3.11 – Порівняльний аналіз, висока завантаженість перехрестя

Таблиця 3.4 – Порівняльна таблиця, висока завантаженість перехрестя

Контролер	Середня к-сть ТЗ на перехресті	Середній час очікування
Фіксований	475.8	246.7
Нечіткий	460.7	231.0
Нечіткий з пріоритетністю руху	431.4	220.5

Різні ступені завантаженості фаз руху на перехресті. В даному досліді, дві фази руху напрямків транспортних засобів мають слабку ступень завантаженості, а інші дві – високу. На рисунку 3.13 зображене графік кількості ТЗ на перехресті в залежності від часу.

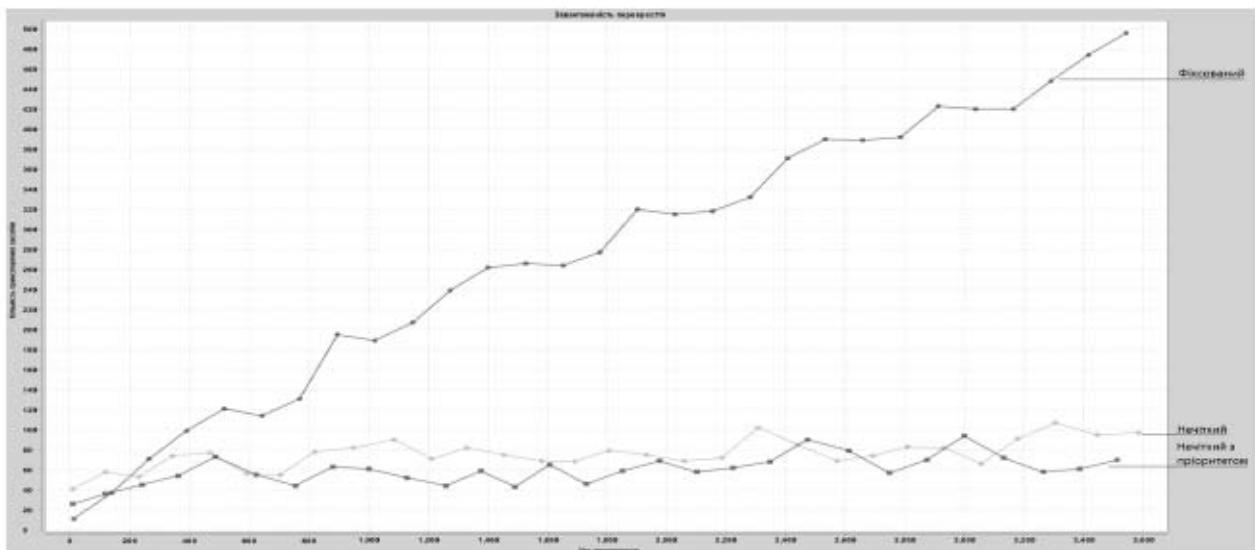


Рисунок 3.11 – Порівняльний аналіз, змішана завантаженість перехрестя

Таблиця 3.5 – Порівняльна таблиця, висока завантаженість перехрестя

Контролер	Середня к-сть ТЗ на перехресті	Середній час очікування
Фіксований	259.8	162.2
Нечіткий	75.1	58.1
Нечіткий з пріоритетністю руху	59.9	40.0

Висновки до розділу

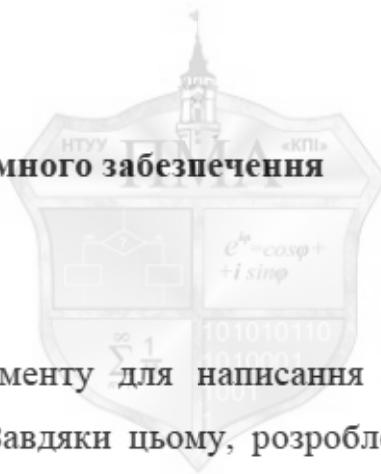
В розділі було описано та побудовано основні діаграми для розробки уніфікованої моделі адаптивної системи регулювання дорожнього руху, а саме діаграми: класів, діяльності, пакетів та компонентів.

Також, розроблена математична модель вдосконаленої підсистеми керування фазами руху та проведено відповідні досліди для оцінки адекватності даної моделі та її порівняння з існуючими моделями.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ УНІФІКОВАНОЇ МОДЕЛІ ФУНКЦІОНУВАННЯ ПЕРЕХРЕСТЯ В РЕАЛЬНОМУ ЧАСІ

В ході проведення дослідження, для оцінки розробленої моделі та її порівняння зі вже існуючими було розроблено та реалізовано програмне забезпечення для моделювання функціонування перехрестя в реальному часі. Програмне забезпечення має користувальський інтерфейс, тому крім моделювання, також може виконувати функцію моніторингу та відображення поточної ситуації на реальному перехресті.

4.1 Опис програмного забезпечення



В якості інструменту для написання програми було обрано мову програмування Java. Завдяки цьому, розроблене програмне забезпечення є кросплатформеним.

Згідно діаграми класів, основними є класи Контролер та Перехрестя. Клас Контролер (Controller) представляє собою абстрактним клас та містить основні параметри, які є спільними для побудови будь-якого контролера.

```
public abstract class Controller {
    // час початку роботи контролера
    private static long startTime;
    // загальний час моделювання
    private static long simulationDurationTime =
        Constants.SIMULATION_DURATION_TIME;
    protected String name;
    protected Intersection intersection;
```

```

protected Statistics statistics;

public Controller(Intersection intersection) {
    this.intersection = intersection;

    // встановлення специфічних для контролера налаштувань
    specificSettings();
}

public void launch() throws InterruptedException{
    startTime = Timer.currentTime();
    statistics = new Statistics(this);
    // початок моделювання
    intersection.launchDirections();
    while(isRunning && isOn()){
        // основним метод керування фазами руху
        regulate();
    }
    intersection.applyDefaultSetting();
    // завершення
}

```

Для побудови специфічного контролера, достатньо унаслідуватись від класу Controller та реалізувати всі його абстрактні методи.

Метод regulate() – виконує безпосереднє керування фазами руху.

Метод specificSettings() – дає змогу встановити специфічні для контролера налаштування.

Конструктор класу Controller приймає лише одне значення – Intersection.

Клас Перехрестя (Intersection) служить програмний представленим реальної ділянки дорожнього руху.

```

public abstract class Intersection {

    private Direction directions[]; // напрямки руху
    protected Phase phases[]; // фази руху
    protected Phase currentPhase; // поточна фаза

    public Intersection(int phases, int directions) {
        this.phases = new Phase[phases];
        this.directions = new Direction[directions];
        init();
    }

    public void launchDirections() {
        startDirections();
    }

    protected abstract void init();

    public void switchOnSpecifiedPhase(Phase nextPhase) {
        showYellowLight(currentPhase, nextPhase);
        this.currentPhase.deactivate();
        showGreenLight(nextPhase, currentPhase);
        this.currentPhase = nextPhase;
        this.currentPhase.activate();
    }

    private synchronized void showYellowLight(Phase
currentPhase, Phase nextPhase) {
        currentPhase.light = TrafficLight.YELLOW;
        nextPhase.light = TrafficLight.YELLOW;
        try {
            this.wait(3000);
        } catch (InterruptedException e) {
            e.printStackTrace(); }
    }
}

```

```

private synchronized void showGreenLight(Phase nextPhase,
Phase previousPhase) {
    nextPhase.light = TrafficLight.GREEN;
    previousPhase.light = TrafficLight.RED;
}

```

Конструктор класу Intersection приймає два параметри: кількість напрямків та кількість фаз руху. Така ініціалізація дозволяє моделювати будь-яке перехрестя.

Для створення специфічного перехрестя необхідно унаслідуватись від класу Intersection та реалізувати абстрактний метод init(). В цьому методі описується для якої фази, які напрямки руху відносяться та яка фаза має бути активною на початку роботи програми. Для створення чотирьох направленого перехрестя був написаний наступний код:

```

public class FourWayIntersection extends Intersection {

    public FourWayIntersection(int phases, int directions) {
        super(name, phases, directions);
    }

    @Override
    protected void init() {

        phases[0].setDirections(getDirection(3),
            getDirection(4), getDirection(9), getDirection(10));

        phases[1].setDirections(getDirection(5),
            getDirection(11));

        phases[2].setDirections(getDirection(6),
            getDirection(7), getDirection(0), getDirection(1));
    }
}

```

```

        phases[3].setDirections(getDirection(2),
getDirection(8));

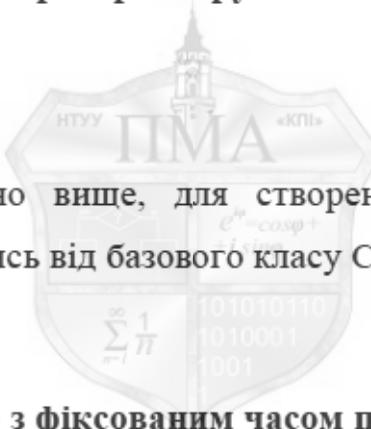
        setCurrentPhase(phases[0]);
    }

}

```

Клас FourWayIntersection унаслідуваний від базового класу Intersection. Напрямки руху 3, 4, 9, 10 відносять до фази 0, напрямки 5, 11 до фази 1 і т.д. Першою активною фазою буде фаза 0 - setCurrentPhase (phases [0]).

4.2 Реалізація контролерів керуванням фазами руху



Як було описано вище, для створення специфічного контролера необхідно унаслідуватись від базового класу Controller.

4.2.1 Контролер з фіксованим часом перемикання фаз руху

Програмна реалізація має наступний вигляд:

```

public class PretimedController extends Controller {

    public PretimedController(Intersection intersection) {
        super(intersection);
    }

    @Override
    protected synchronized void regulate() throws InterruptedException
    {
        this.wait(intersection.getCurrentPhase().getPhaseTime());
    }
}

```

```

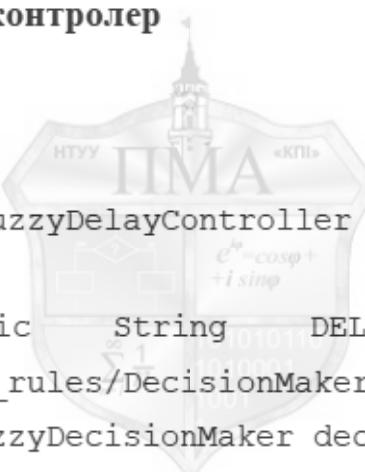
        statistics.update();
        intersection.switchOnNextPhaseByDefault();
    }

    @Override
    protected void specificSettings() { }
}

```

Головний метод regulate() перемикає поточну фазу руху на наступну кожні intersection.getCurrentPhase().getPhaseTime() секунд, при цьому не виконуючи більше ніяких обчислень чи аналізу.

4.2.2 Нечіткий контролер



```

public class FuzzyDelayController extends Controller {
    public static String DELAY_CONTROL_RULES_PATH =
        "fuzzy_control_rules/DecisionMaker.fcl";
    private FuzzyDecisionMaker decisionMaker;
    private Phase nextPhase;
    private long delay;

    public FuzzyDelayController(Intersection intersection) {
        super(intersection);
    }

    @Override
    protected void specificSettings() {
    }

    @Override

```

```

protected synchronized void regulate() throws
InterruptedException {
    this.wait(intersection.getCurrentPhase().getPhaseTime());
    this.nextPhase = intersection.getNextPhaseByDefault();
    Phase curPhase=intersection.getCurrentPhase();
    this.delay=decisionMaker.getTimeDelay(nextPhase.average
WaitingVehiclesAtDirection(),intersection.getCurrentPha
se().averageWaitingVehiclesAtDirection());
    this.wait(delay);
    statistics.update();
    intersection.switchOnSpecifiedPhase(nextPhase);
}

```

Метод regulate() перемикає поточну фазу руху на наступну в залежності від поточної ситуації на перехресті. Час затримки визначається завдяки окремому модулю, який містить в собі всі нечіткі правила та в якому реалізовано систему нечіткого виведення. Час затримки delay = decisionMaker.getTimeDelay(...).

4.2.3 Нечіткий контролер з врахуванням пріоритетності руху

```

public class FuzzyUrgencyAndDelayController extends
Controller{
    public static String URGENCY_CONTROL_RULES_PATH =
"fuzzy_control_rules/UrgencyEvaluator.fcl";
    public static String DELAY_CONTROL_RULES_PATH =
"fuzzy_control_rules/DecisionMaker.fcl";
}

```

```

private FuzzyDecisionMaker decisionMaker;
private FuzzyUrgencyEvaluator urgencyEvaluator;
private Phase nextPhase;
private long delay;

public FuzzyUrgencyAndDelayController(Intersection intersection) {
    super(intersection);
}

@Override
protected void specificSettings() { }

@Override
protected synchronized void regulate() throws
InterruptedException {
    this.wait(intersection.getCurrentPhase().getPhaseTime());
    this.nextPhase=urgencyEvaluator.nextGreenPhase(intersec-
tion.redPhases());
    Phase curPhase = intersection.getCurrentPhase();
    this.delay =
decisionMaker.getTimeDelay(nextPhase.averageWaitingVehiclesA-
tDirection(),
intersection.getCurrentPhase().averageWaitingVehiclesAtDirec-
tion());
    this.wait(delay);
    statistics.update();
    intersection.switchOnSpecifiedPhase(nextPhase); }
}

```

Метод `regulate()` перемикає поточну фазу руху на наступну в залежності від поточної ситуації на перехресті. Спочатку перший модуль контролера визначає пріоритетність кожного з 3 напрямків руху

`this.nextPhase=urgencyEvaluator.nextGreenPhase(intersection.redPhases())`. Далі другий модуль для визначеного напрямку визначає час затримки `delay = decisionMaker.getTimeDelay(...)`.

4.3 Інструкція користувача.

На рисунку 4.1 зображене початковий вигляд інтерфейсу програми одразу після запуску.

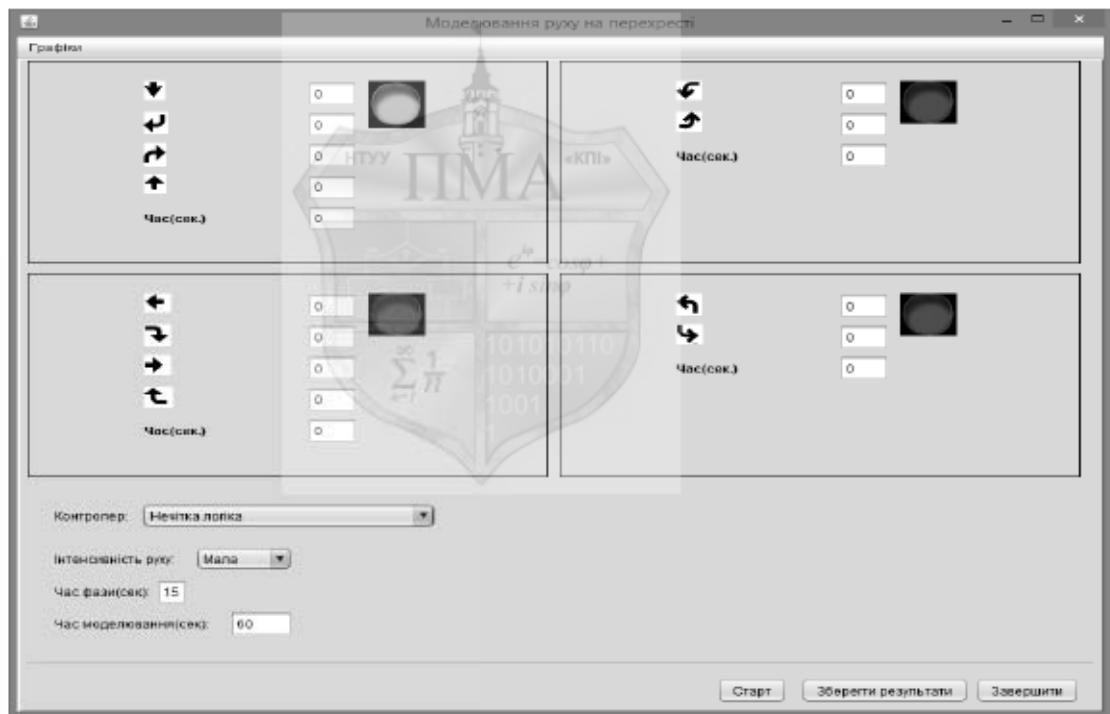


Рисунок 4.1 – Інтерфейс програми

Налаштування:

1. Контролер – вибір контролеру управління фазами руху;
2. Інтенсивність руху – мала, середня, висока;
3. Час фази – тривалість фаз руху за замовчуванням;
4. Час моделювання – час моделювання.

Після встановлення налаштувань та запуску моделювання, користувач може спостерігати за поточною ситуацією на перехресті та статистикою транспортного руху (Рис. 4.2).



Рисунок 4.2 – Моделювання

Після завершення моделювання користувач може зберегти основну статистику в файл для подальшого її аналізу (Рис. 4.3).

```

Fuzzy Delay. LPhase time: 10000. Low: 0.0-0.1 DecisionMaker3 - 22.
Average total time waiting: 18.147172701883253
Total number of phases: 314
Total time: 3611
Waiting vehicles for the last phase: 5
Average waiting vehicles for the last phase : 11.130573248407643

Phase Vehicles Time AverageVehiclesDelay
1 3 11 3.0
2 8 22 5.5
3 10 34 7.0
4 17 45 9.5
5 15 57 10.6
6 19 69 12.0
7 12 80 12.0
8 15 92 12.375
9 9 103 12.0
10 13 115 12.1
11 10 126 11.909090909090908
12 10 138 11.75
13 7 149 11.384615384615385

```

Рисунок 4.3 – Збереження статистики

В меню «Графіки», користувач може запросити графічну статистику проведеного моделювання. Для цього буде необхідним лише обрати файл зі збереженою статистикою (Рис 4.4).

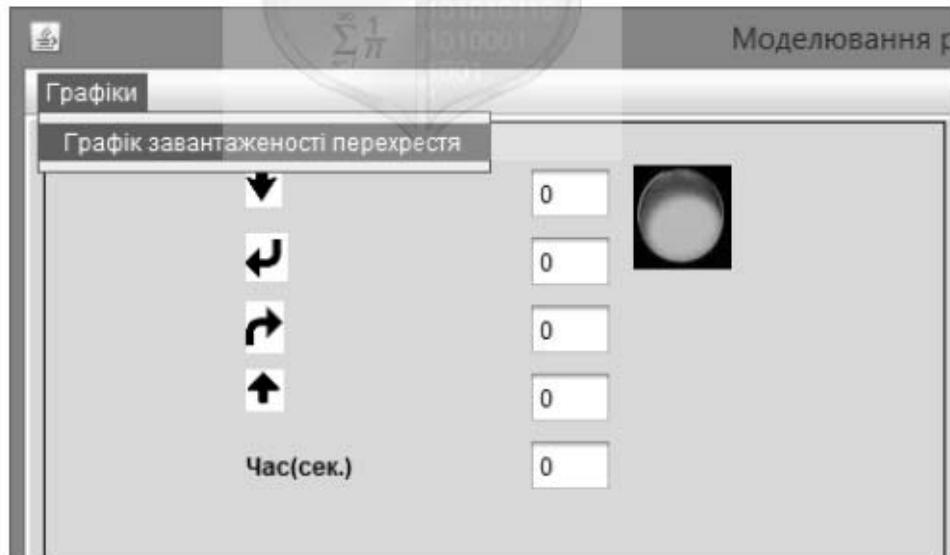


Рисунок 4.4 – Меню графіки

Висновки до розділу

У розділі приведена програмна реалізація системи для моделювання функціонування будь-якого перехрестя в режимі реального часу та оцінки ефективності роботи адаптивної системи регулювання дорожнього руху на перехресті.

Запропонована реалізація інтерфейсу користувача та продемонстровані результати роботи програмного продукту.

Описана інструкція користування, основні характеристики та можливості програмного забезпечення. Наведені фрагменти коду, для освітлення основних нюансів програмної реалізації та відображення програмного інтерфейсу для інтегрування даної програми іншими розробниками.

ВИСНОВКИ

Результатом магістерської дисертації є уніфікована модель адаптивної системи регулювання дорожнього руху, що дозволяє враховувати поточну ситуацію на перехресті та збільшити пропускну спроможність окремих ділянок дорожнього руху. Така система дає можливість зменшити кількість транспортних засобів, які очкують своєї черги на перехрестях та середній час затримки кожного з них.

Встановлено:

- сучасна система адаптивного регулювання складається з двох основних компонент: активної та пасивної;
- існує два основні підходи для реалізації адаптивного регулювання дорожнього руху: нечіткий на нейронечіткий;
- відсутність програмних засобів для порівняння ефективності роботи різних адаптивних контролерів між собою;

Розроблено:

- уніфіковану модель адаптивної системи регулювання дорожнього руху
- вдосконалений алгоритм роботи контролера керування фазами руху;
- програмне забезпечення для моделювання дорожнього руху на перехресті та порівняння ефективності роботи контролерів між собою;

Реалізовано:

- програмний інтерфейс для інтеграції та порівняння будь-яких контролерів регулювання дорожнього руху;

- користувальський інтерфейс для можливості моніторингу поточної ситуації на ділянці дорожнього руху.

Запропоновано і вдосконалено алгоритм з використанням нечіткої логіки для керування фазами руху на перехресті. Врахування пріоритетності напрямків руху дозволило збільшити пропускну спроможність перехрестя, а вдосконалена модель контролера, дозволила зменшити кількість запитів до детектора транспортних засобів. Таким чином, в порівнянні з не адаптивним підходом, пропускна спроможність перехрестя збільшилась на 46% у випадку слабкої завантаженості та на 10% у випадку високої завантаженості.

Результати роботи опубліковані в матеріалах:

- сьомої наукової конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг»;
- 17-ї Міжнародної науково-технічної конференції SAIT 2015.

Перспективи:

- розширення функціональності контролера керування фазами руху для роботи на перехрестях з пішохідним рухом
- розробка програмного забезпечення для синхронізації роботи адаптивних контролерів світлофорів суміжних перехресть

СПИСОК ЛІТЕРАТУРИ

1. Клебанова Т.С., Чаговець Л.О., Панасенко О.В., Нечітка логіка та нейронні мережі в управлінні підприємством: Монографія.-Х.: ВД «ІНЖЕК», 2011. – 240 с
2. А. В. Леоненков. Нечеткое моделирование в среде MatLab и FuzzyTech [Текст] / А. В. Леоненков. – СПб.: БХВ-Петербург, 2005. – 736 с.
3. Papas C. P., Mamdani E. H. A fuzzy logic controller for a traffic junction // IEEE Transactions on Systems, Man and Cybernetics. – 1977. – Vol. 7, – P. 707–717.
4. Niittymäki J., Turunen E. Traffic signal control on similarity logic reasoning // Fuzzy Sets and Systems, 2003. – P. 101-140.
5. Balaji P. G., Srinivasan D. Type-2 fuzzy logic based urban traffic management // Engineering Applications of Artificial Intelligence, 2011. – C. 10-19.
6. Cox, E. Adaptive fuzzy systems. Spectrum // IEEE, 30:2. - 1993 pp. 7-31
7. Тэрано, Т., Асai, К., Сугэно, М. Прикладные нёчеткие системы. — М.: Мир, 1993. — 368 с.
8. M. Koukol, L. Zajíčková, LukášMarek, P. Fuzzy Logic in Traffic Engineering: A Review on Signal Control // Modelling and Simulation in Engineering. – 2014. P. 1–15.
9. Yan Ge. A Two-Stage Fuzzy Logic Control Method of Traffic Signal Based on Traffic Urgency Degree // Modelling and Simulation in Engineering. – 2014. P. 1–6.
- 10.Методы робастного, нейро-нечёткого и адаптивного управления: Учебник / Под ред. Н.Д. Егупова; издание 2-ое, стереотипное. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. - 744 с.

- 11.T. ROYANI, J. HADDADNIA, M. ALIPOOR Traffic Signal Control for Isolated Intersections Based on Fuzzy Neural Network and Genetic Algorithm // Department of electrical engineering Tarbiat Moallem University of Sabzevar 2010. P. 1-5
- 12.Усков А. А., Кузьмин А. В. Интеллектуальные технологии управления. Искусственные нейронные сети и нечеткая логика. — М.: Горячая г
- 13.Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. — М.: Мир, 1976. — 166 с.
- 14.Тэрано, Т., Асai, К., Сугэно, М. Прикладные нёчеткие системы. — М.: Мир, 1993. — 368 с.
- 15.Новак В., Перфильева И., Мочкож И. Математические принципы нечёткой логики = Mathematical Principles of Fuzzy Logic. — Физматлит, 2006. — 352 с
- 16.Методы робастного, нейро-нечёткого и адаптивного управления: Учебник / Под ред. Н.Д. Егупова; издание 2-ое, стереотипное. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. - 744 с
- 17.James Rumbaugh, Ivar Jacobson, Grady Booch (1999). The unified modeling language reference manual (англ.). Addison Wesley Longman Inc. ISBN 0-201-30998-X
- 18.Fowler М. / Фаулер М., UML Distilled / UML Основы, 3-е издание [Текст] / Фаулер М. – СПб: Символ-Плюс, 2005 – 184 с.
- 19.James Rumbaugh, Ivar Jacobson, Grady Booch (1999). The unified modeling language reference manual (англ.). Addison Wesley Longman Inc. ISBN 0-201-30998-X.
- 20.Сирота С.В., Душин О.С. Модель адаптивної системи регулювання дорожнього руху на перехресті з використанням нечіткої логіки // Прикладна математика та комп’ютинг. ПМК, 2015 : сьома наук. конф. магістрантів та аспірантів, Київ, 15—17 квіт. 2015 р. : зб. тез доп. / [редкол.: Дичка І. А. та ін.]. — К. : Просвіта, 2015.

21.Штовба С.Д. Проектирование нечетких систем средствами MATLAB /
С. Штовба. – М: Горячая линия–Телеком, 2007. – 288 с

