

# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

## ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Інститут Навчально-науковий інститут Комп'ютерних систем і технологій «Індустрія 4.0» ім. П.М. Платонова  
Факультет Автоматизація та робототехніка  
Кафедра Автоматизація технологічних процесів і робототехнічних систем  
Ступінь вищої освіти Магістр  
Спеціальність 151 - «Автоматизація та комп'ютерно-інтегровані технології»  
Освітньо-наукова програма Комп'ютерні системи та програмна інженерія в автоматизації

### КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

**Тема: «Підвищення ефективності автоматичного керування вакуумною деалкоголізацією вина в потоці»**

Розробив М.О. Саприкін

Керівник  
к.т.н., доцент О.В. Мазур

Зав. кафедри АТПіРС  
д.т.н., професор В.А. Хобін

*«e-версія роботи ідентична оригіналу»* М.О. Саприкін

*«e-версію роботи прийнято»* Т.В. Волик

Депозитор кафедри АТПіРС

## ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Інститут	Навчально-науковий інститут Комп'ютерних систем і технологій «Індустрія 4.0» ім. П.М. Платонова
Факультет	Автоматизації та робототехніки
Кафедра	Автоматизації технологічних процесів і робототехнічних систем
Ступінь вищої освіти	Магістр
Спеціальність	151 - «Автоматизація та комп'ютерно-інтегровані технології»
Освітня програма	Комп'ютерні системи та програмна інженерія в автоматизації програма

### ЗАТВЕРДЖУЮ

Завідувач кафедри АТП і РС  
 \_\_\_\_\_ д.т.н., проф. Хобін В.А.  
 «24» листопада 2021року

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

- Здобувач освіти **Саприкін Микита Олексійович**
- Тема роботи **«Підвищення ефективності автоматичного керування вакуумною деалкоголізацією вина в потоці»**
- Керівник кваліфікаційної роботи магістра **Мазур Олександр Васильович, к.т.н., доцент**  
 П.п. 2 і 3 затверджено наказом ОНТУ від 11 листопада 2021 року № 945-03.
- Строк подання студентом зброшурованої випускної роботи – 05 червня 2023 р.
- Вихідні дані до випускної роботи: матеріали виконаних індивідуальних завдань (ІЗ) виробничої практики, дипломної роботи бакалавра, дослідницької практики, курсових та самостійних робіт, виконаних відповідно до ІЗ.
- Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити):  
 Вступ (актуальність роботи, зв'язок з науковими напрямками робіт академії, мета та задачі дослідження, об'єкт та предмет дослідження, методи дослідження, наукова новизна отриманих результатів, практичне значення отриманих результатів, апробація результатів роботи, публікації, структура та об'єм роботи).  
 Розділ 1. Технологічний процес вакуумної деалкоголізацією вина та задачі підвищення його ефективності засобами автоматичного керування.  
 Розділ 2. Удосконалення алгоритмів, що реалізують функції регулювання САК ТП вакуумної деалкоголізацією вина.  
 Розділ 3. Удосконалення алгоритмів, що реалізують функції регулювання САК ТП, на основі застосування штучних нейронних мереж.  
 Розділ 4. Розробка та дослідження САК, що реалізує нову, інтелектуально просунуту, функцію.  
 Розділ 5. Реалізація та підготовка до впровадження на базі кафедри АТПіРС ОНТУ.  
 Додатки (допоміжні матеріали, ксерокопії програм конференцій, статей, патентів).

7. Консультанти розділів випускної кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Удосконалення алгоритмів, що реалізують функції регулювання САК ТП, на основі застосування штучних нейронних мереж	Гурський О.О. к.т.н. доцент каф. АТПіРС		
Розробка та дослідження САК, що реалізує нову, інтелектуально просунуту, функцію	Хобін В.А. д.т.н., проф. зав.каф. АТПіРС		

8. Дата видачі завдання 1 вересня 2022 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи магістра	Термін виконання етапів роботи	Примітка
1	Вступ та загальна характеристика роботи	01.02.23 – 15.02.23	
2	Технологічний процес та задачі підвищення його ефективності засобами автоматичного керування	16.02.23 – 1.03.23	
3	Удосконалення алгоритмів, що реалізують функції регулювання САК ТП	02.03.23 – 17.03.23	
4	Удосконалення алгоритмів, що реалізують функції регулювання САК ТП, на основі застосування штучних нейронних мереж	18.03.23 – 08.04.23	
5	Розробка та дослідження САК, що реалізує нову, інтелектуально просунуту, функцію	09.04.23 – 30.04.23	
6	Реалізація та підготовка до впровадження результатів роботи	01.05.23 – 15.05.23	
7	Оформлення додатків та роботи в цілому	16.05.23 – 23.05.23	
8	Здача роботи на перевірку керівникові, виправлення зауважень, підпис керівника	24.05.23 – 31.05.23	
9	Брошування роботи, представлення її завідувачу кафедри, підпис, направлення роботи на зовнішнє рецензування та захист у ЕК	01.06.23 – 05.06.23	

Студент

Саприкін М.О.

Керівник кваліфікаційної роботи

Мазур О.В.

## Анотація

Обсяг роботи – 107 стр., кількість розділів - 5, ілюстрацій – 104, таблиць – 4, джерел по переліку посилань – 16.

Об'єкт дослідження або розробки – підвищення ефективності автоматичного керування вакувної деалкоголізацією вина в потоці.

Мета роботи – підвищити ефективність автоматичного керування процесом вакувної деалкоголізації вина в потоці, яка б підтримувала регульовані змінні в регламентних зонах як в сталих, так і в перехідних режимах роботи.

Методи дослідження та інструментарій – при ідентифікації властивостей об'єкту керування використовувалися методи уявного активного та пасивного експерименту з подальшою обробкою їх результатів; моделі об'єкту керування та системи регулювання розроблялися в середовищі Simulink/Matlab; параметричний синтез системи керування проведений методом оптимізації показника якості її функціонування; розробка удосконаленої системи проводилася аналітично із застосуванням апарату передатних функцій.

Отримані результати – отримана система підвищеної динамічної точності, яка підтримує регульовані змінні в регламентних зонах як в сталих, так і в перехідних режимах роботи

Область застосування – система може бути використана при модернізації процесу вакувної деалкоголізації вина в потоці.

Значущість роботи і висновки – розроблена система має переваги у порівнянні із традиційними САК, які використовуються на практиці.

Ключові слова: технологічний процес, об'єкт керування, регулятор, система керування.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	8
ВСТУП.....	9
РОЗДІЛ 1. ТЕХНОЛОГІЧНИЙ ПРОЦЕСС ВАКУМНОЇ ДЕАЛКОГОЛІЗАЦІЇ ВИНА І ЗАДАЧІ ПІДВИЩЕННЯ ЙОГО ЕФЕКТИВНОСТІ ЗАСОБАМИ АВТОМАТИЧНОГО КЕРУВАННЯ.	12
1.1. Опис технологічного процесу і основного обладнання .....	12
1.2. Аналіз впливу режимів ведення технологічного процесу і роботи обладнання на його техніко-економічні показники, аналіз існуючих регламентів і наслідків їх порушень .....	12
1.3. Аналіз існуючих САК ТП вакумної деалкоголізації вина в потоці.....	14
1.4. Комплекс математичних моделей технологічного процесу вакумної деалкоголізації вина в потоці як об'єкта регулювання.....	15
1.5. Недоліки існуючих САК ТП вакумної деалкоголізації вина, обґрунтування напрямлень вдосконалення САК, як нових джерел підвищення економічної ефективності технологічного процесу.....	19
1.6. Висновки.....	19
РОЗДІЛ 2. ВДОСКОНАЛЕННЯ АЛГОРИТМІВ, ЩО РЕАЛІЗУЮТЬ ФУНКЦІЇ РЕГУЛЮВАННЯ САК ВАКУМНОЇ ДЕАКЛГОЛІЗАЦІЇ ВИНА.....	20
2.1. Специфічні особливості ОР та обґрунтування вибору методів корекції САР для підвищення їх динамічної точності.....	20
2.2. Структурний синтез САР підвищеної динамічної точності.....	20
2.3. Висновки.....	26
РОЗДІЛ 3. ВДОСКОНАЛЕННЯ АЛГОРИТМІВ, ЯКІ РЕАЛІЗУЮТЬ ФУНКЦІЇ РЕГУЛЮВАННЯ САК ТП ВАКУМНОЇ ДЕАЛКОГОЛІЗАЦІЇ ВИНА, НА ОСНОВІ ВИКОРИСТАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ.....	27

3.1. Обґрунтування необхідності використання штучних нейронних мереж для підвищення динамічної точності САР та розробка структури САР з нейрорегулятором.....	27
3.2. Реалізація в програмному середовищі MATLAB\Simulink моделей каналів регулювання розглянутого об'єкта керування, що характеризується нелінійними статичними характеристиками.....	28
3.3. Параметричний синтез САР з традиційним ПІД-регулятором для об'єкта з нелінійною статичною характеристикою каналу регулювання.....	31
3.4. Розробка моделі САР з нечітким регулятором (НР) для об'єкта з нелінійною статичною характеристикою каналу регулювання.....	34
3.5. Порівняльний аналіз функціонування САР з традиційним ПІД-регулятором і з САР з нечітким регулятором.....	44
3.6. Обґрунтування актуальності застосування нейромережових алгоритмів керування.....	46
3.7. Розробка моделі САР з нейрорегулятором.....	47
3.8. Параметрична оптимізація САР з нейрорегулятором.....	58
3.9. Аналіз функціонування різних САР за якістю регулювання за умов впливів, що збурюють, випадкового характеру.....	61
3.10. Висновки.....	64
<b>РОЗДІЛ 4. РОЗРОБКА СТРУКТУРИ МОДЕРНІЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ ПРОЦЕСОМ ВАКУМНОЇ ДЕАЛКОГОЛІЗАЦІЇ</b>	
<b>ВИНА.....</b>	<b>66</b>
4.1. Опис загальних принципів побудови системи керування.....	66
4.2. Розробка структурної схеми САК та опис принципу її дії.....	69
4.3. Обґрунтування налаштовувальних параметрів нових алгоритмів керування.....	70
4.4. Доопрацювання математичної моделі ОР для вирішення задачі синтезу та аналізу системи гарантуючого управління.....	70

4.5. Розробка імітаційної моделі системи гарантуючого управління температури конденсату та дослідження її роботи .....	71
4.6. Висновки.....	78
<b>РОЗДІЛ 5. РЕАЛІЗАЦІЯ І ПІДГОТОВКА ДО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ РОБОТИ НА ПІДПРИЄМСТВІ.....</b>	
5.1. Існуюча програмно-технічна структура підсистеми керування процесом вакуумної деалкоголізації вина в потоці і формулювання задач по її розвитку для впровадження розробок .....	80
5.2. Розробка, налаштування і тестування прикладного програмного забезпечення для реалізації алгоритмів керування.....	85
5.3. Оцінка економічної ефективності впровадження розробок.....	99
5.4. Висновки за розділом.....	99
<b>ВИСНОВКИ.....</b>	<b>100</b>
Додаток А.....	102
<b>СПИСОК ЛІТЕРАТУРИ.....</b>	<b>110</b>

## Перелік умовних позначень

- ПФ – імпульсна перехідна функція ;
- АЧХ – амплітуда-частотні характеристики ;
- ФЧХ – фаза-частотні характеристики ;
- СКП – середньо квадратичний період ;
- САР – система автоматизованого регулювання;
- ЗНВ – зона незначних відхилень
- ММ – математична модель
- ОК – об'єкт керування
- САК – система автоматичного керування
- САК ПДТ – САК підвищеної динамічної точності
- ТП – технологічний процес
- ПЗО - пристрій зв'язку з об'єктом;
- СКЗ - спрощений коригуючий зв'язок;
- ФЛС – функціональна логічна схема.

## Вступ

### *Актуальність теми*

Одним із етапів технологічного процесу деалкоголізації вина є процес підтримання різниці тисків, який дозволяє отримати кінцевий продукт з бажаними споживчими характеристиками. Існуючі системи керування процесом деалкоголізації вина не забезпечують необхідної динамічної точності відтворення заданого значення температури виноматеріалу та температуру конденсату, що спричиняє брак готової продукції.

В лабораторії кафедри АТПтаРС Одеської національної академії харчових технологій розроблено установку для деалкоголізації вина. Для максимально ефективного керування процесом деалкоголізації вина актуальною є розробка спеціалізованої вбудованої мікропроцесорної САР.

### *Зв'язок з науковими програмами, планами, темами.*

Робота виконувалась на кафедрі автоматизації виробничих процесів, Одеської національної академії харчових технологій (ОНАХТ) і відповідає основному науковому напрямку №3 ОНАПТ «Створення і розробка нового високоефективного обладнання, теорії, методів його розрахунку і проектування».

### *Мета і задачі дослідження.*

Мета дослідження – засобами автоматичного регулювання не допустити вихід за граничні значення технологічних параметрів, підвищити ефективність деалкоголізації вина в потоці, забезпечити стійкість, зменшити затрати енергії.

*Задачі дослідження* у зв'язку з поставленою метою можна сформулювати так:

- Детально дослідити процес деалкоголізації вина в потоці, створити його математичні моделі, визначити основні регламенти ведення технологічного процесу.

- Проаналізувати існуючі алгоритми ведення технологічного процесу, та синтезувати свій власний алгоритм, який покращить техніко-економічні показники даного процесу.

*Об'єктом дослідження* в даній роботі є керований технологічний процес вакуумна деалкоголізації вина.

*Предметом дослідження* є система автоматичного керування процесом вакуумної деалкоголізації вина в потоці, що містить моделі процесу як об'єкта управління.

*Методи дослідження*

- загальні теорії систем, в тому числі їх функціонально – структурного аналізу (задача 1);
- програмно – технічного синтезу комп'ютерних інформаційно – керуючих систем комп'ютерної обробки рисунків (задачі 2 і 5);
- планування і проведення експериментів, обробка їх результатів і ідентифікації моделей ОУ (задача 3);
- теорії автоматичного керування і її підрозділів – статистичної теорії автоматичних систем, теорії систем з запізненням, теорії систем гарантуючого управління, теорії цифрових систем управління (задача 4);
- імітаційного моделювання систем керування (задачі 3 і 4).

*Наукова новизна отриманих результатів.*

Наукова новизна отриманих результатів полягає в тому, що:

- Отримано математичні моделі процесу вакуумної деалкоголізації вина в потоці як об'єкту керування, які можна використовувати в подальшому, для синтезу і аналізу систем керування різного інтелектуального рівню.

- Створено нову систему автоматичного регулювання процесу вакуумної деалкоголізації вина в потоці, основу якого складають два контури регулювання: температура виноматеріалу, температура конденсату.

*Практична значимість отриманих результатів.*

Практична цінність роботи полягає в тому, що в результаті впровадження розроблених алгоритмів в систему керування вакуумної деалкоголізації вина забезпечується зниження енерговитрат. Результати дослідження можуть також застосовуватися для пошуку оптимальних режимів функціонування аналогічних систем деалкоголізації.

*Особистий внесок здобувача.*

У роботі був виконаний літературний пошук та систематизація отриманої інформації. Була розроблена структура і сформульовані основні функції створюваної системи автоматичного регулювання. Було проаналізовано чинники і процеси, які впливають на стан технологічного процесу вакуумної деалкоголізації вина, сформульована в загальному вигляді задача побудови математичної моделі цього процесу.

# РОЗДІЛ 1. ТЕХНОЛОГІЧНИЙ ПРОЦЕС ВАКУМНОЇ ДЕАЛКОГОЛІЗАЦІЇ ВИНА І ЗАДАЧІ ПІДВИЩЕННЯ ЙОГО ЕФЕКТИВНОСТІ ЗАСОБАМИ АВТОМАТИЧНОГО КЕРУВАННЯ

## 1.1 Опис технологічного процесу і основного обладнання.

Суть технологічного процесу – це цілеспрямоване перетворення продуктових і енергетичних потоків у спеціальному технологічному устаткуванні (ТУ) (машинах, апаратах, агрегатах). Суть технологічного процесу деалкоголізації вина з використанням парокомпресійного теплового насосу полягає в підтриманні різниці тисків для забезпечення випарювання спирту, а також підтримки заданої температури виноматеріалу.

Технологічна схема процесу наведена на рис.1.1.

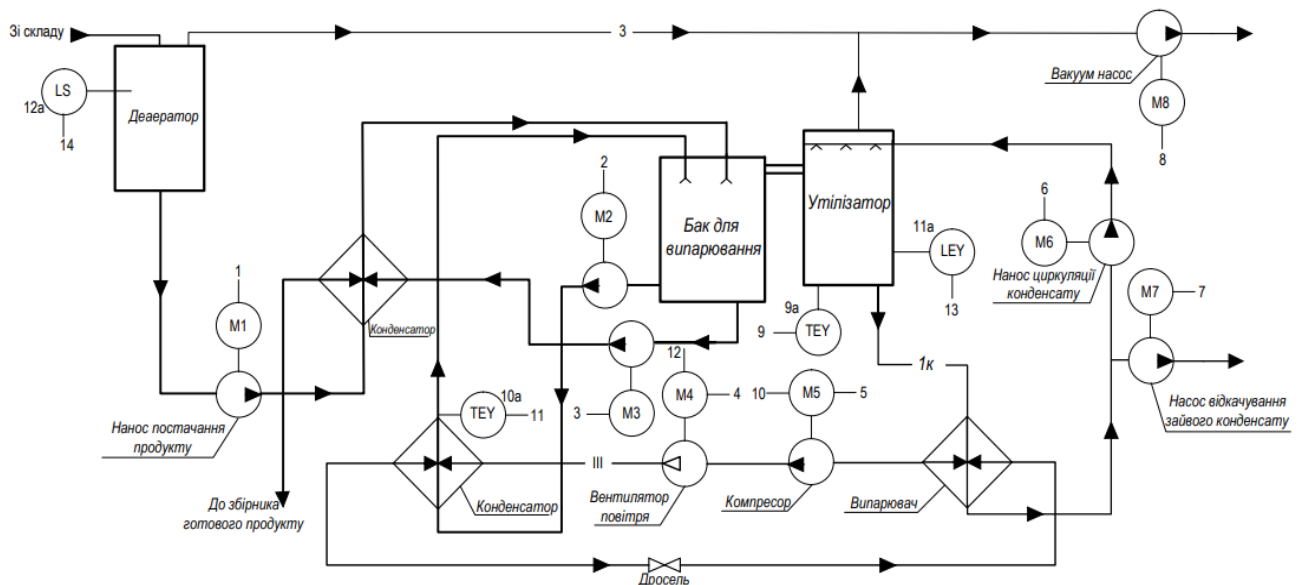


Рис. 1.1 – Технологічна схема процесу вакуумної деалкоголізації вина

У стаціонарному режимі роботи ТП виноматеріал потрапляє у бак для випарювання, де починає циркулювати проходячи через конденсатор який підігріває його до температури 28°C, при цьому в утилізаторі температура конденсату потрібна бути 10°C. За рахунок різниці тисків спирт випарюється, виноматеріал залишається не тронутим. Після цього виноматеріал відкачується нососом при цьому проходячи через пасивний конденсатор, який

прискорює підігрів виноматеріалу.

Апарат для деалкоголізації, який розроблен в лабораторії кафедри АТП і РС ОНАХТ являє собою вакуум випарну установку з парокompресійним насосом.

При цьому номінальне споживання установки становить 40 л/год. Також до складу обладнання входить вакуум насос, парокompресійний насос, електропривід компресору якого має потужність 5кВт.

1.2 Аналіз впливу режимів ведення технологічного процесу і роботи обладнання на його техніко-економічні показники, аналіз існуючих регламентів і наслідків їх порушень.

Для якісної і економічно доцільної реалізації технологічного процесу деалкоголізації вина необхідно регулювати температуру конденсату  $T_{\text{конд}}$ , та температуру вина на виході з конденсатора  $T_{\text{вина}}$ .

Відповідно до нормативів ведення технологічних процесів температура конденсату  $T_{\text{конд}}$  має підтримуватися на рівні  $10^{\circ}\text{C}$  з точністю  $\pm 1^{\circ}\text{C}$ . За цим параметром припустимі короточасні відхилення до  $\pm 5^{\circ}\text{C}$  протягом часу не більше 200 секунд. Регламентна зона за цим параметром наведена на рис.1.2.

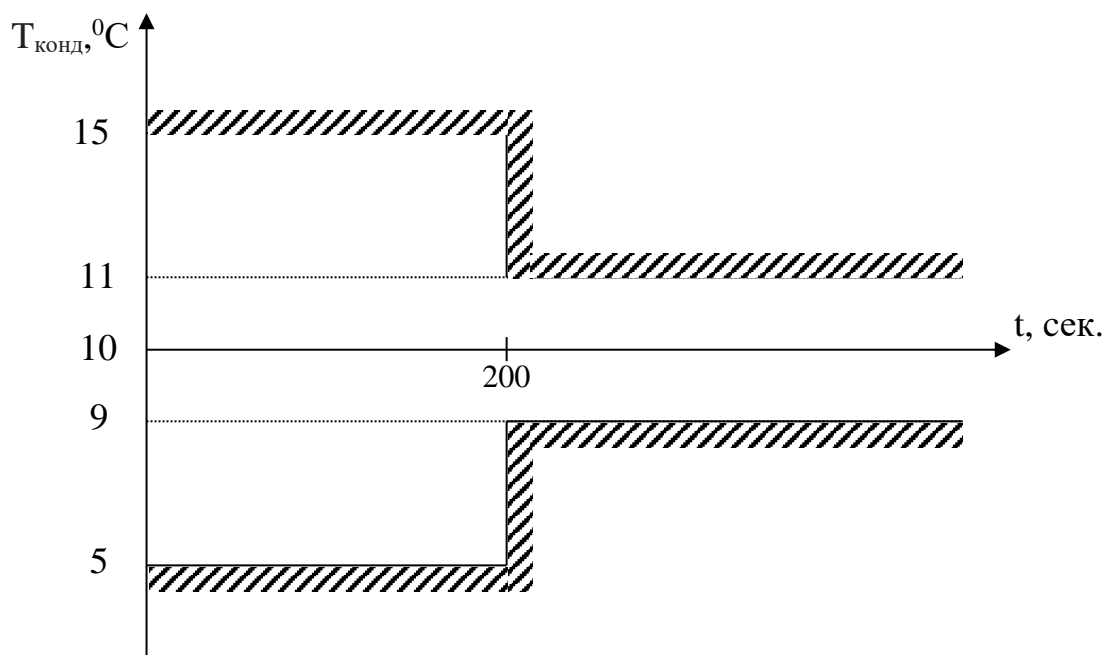


Рис. 1.2 – Регламентна зона для температури конденсату  $T_{\text{конд}}$

Температура виноматеріалу  $T_{\text{вина}}$  має підтримуватися на рівні  $28^{\circ}\text{C}$  з точністю  $\pm 1^{\circ}\text{C}$ . За цим параметром припустимі короточасні відхилення до  $\pm 5^{\circ}\text{C}$  протягом часу не більше 200 секунд. Регламентна зона за цим параметром наведена на рис.1.2.. Регламентна зона за цим параметром наведена на рис. 1.3.

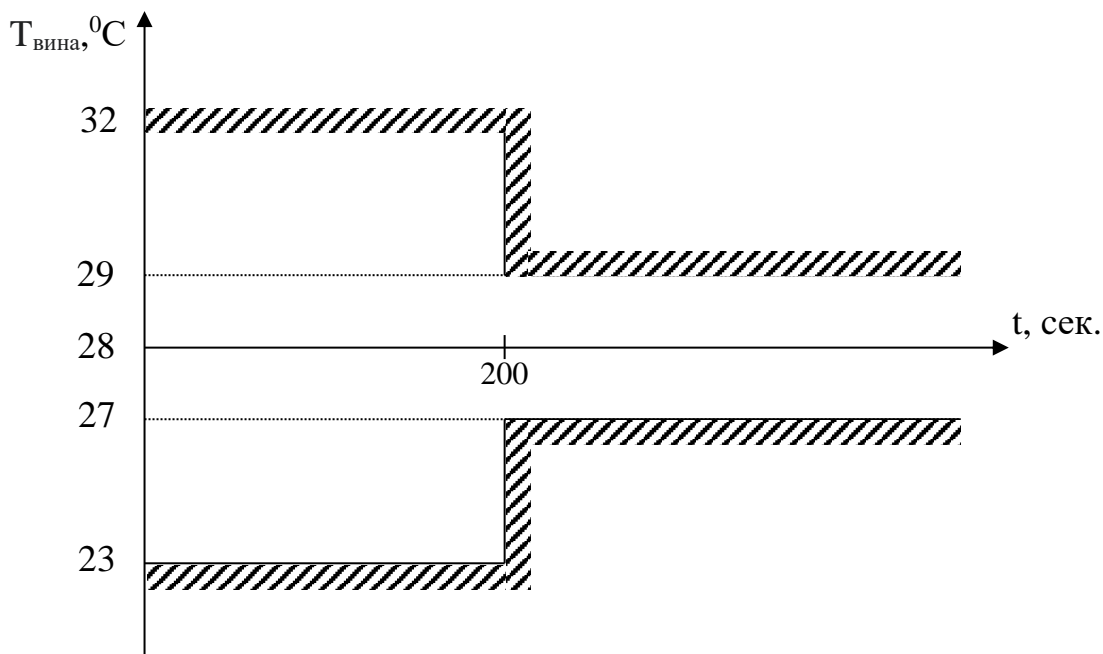


Рис. 1.3 – Регламентна зона для температури виноматеріалу  $T_{\text{вина}}$

Як видно з регламентів на САР для процесу деалкоголізації вина особливо жорсткі вимоги пред'являються до короточасних відхилень, так як саме недотримання цих регламентів може призвести до браку готової продукції і суттєвих економічних збитків.

### 1.3 Аналіз існуючих САК ТП вакуумної деалкоголізації вина в потоці

Найбільш близьким до пропонованого, який вибраний як прототип, є спосіб управління портативною установкою для видалення спирту із алкогольних напоїв, який включає керування температурою пластини для плівкового випаровування, шляхом зміни потужності термоелектричних елементів, та керування тиском в системі за допомогою зміни продуктивності вакуумної системи. Пат. US20130243922 A1 США. Removal of alcohol from potable liquid using vacuum extraction / Judd B. Lynn, Kelli Lynn Fuller. Опубл. 19.09.2013.

1.4 Комплекс математичних моделей технологічного процесу вакуумної деалкоголізації вина в потоці як об'єкта регулювання.

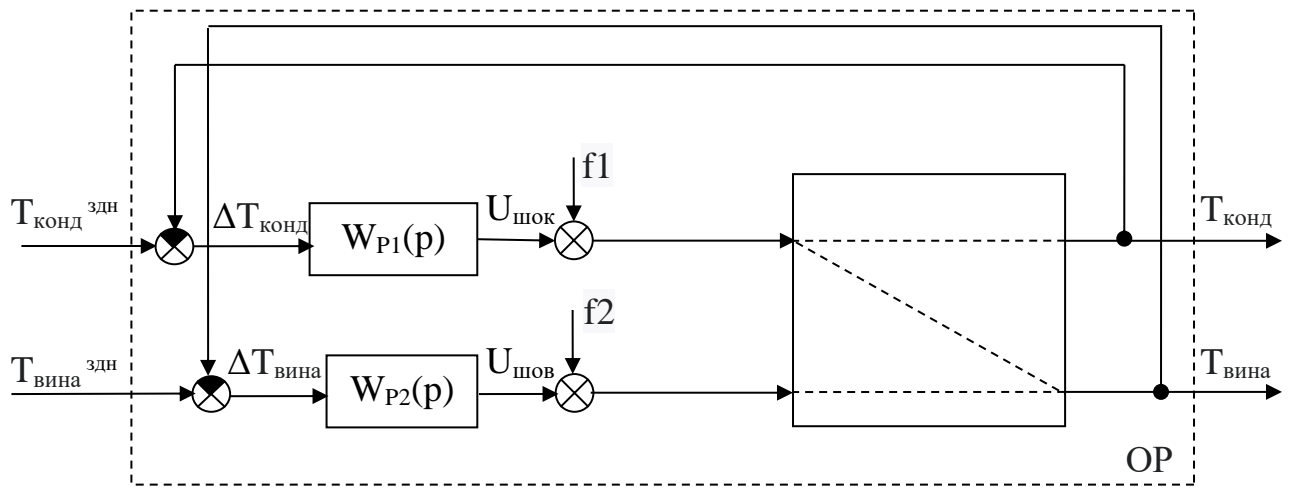


Рис. 1.4– Структурна схема процесу деалкоголізації вина як об'єкту регулювання

$U_{шок}$  – положення регулюючого органу температури конденсату;

$U_{шов}$  – положення регулюючого органу температури виноматеріалу;

$T_{конд}$  – температура конденсату  $^{\circ}C$ ;

$T_{вина}$  – температура охолодженої води  $^{\circ}C$ ;

$f1, f2$  – вектори неконтрольованих збурень;

$W_{P1}(p), W_{P2}(p)$  – передаточні функції регуляторів;

$T_{конд}^{здн}$  – задане значення температури конденсату  $^{\circ}C$ ;

$T_{вина}^{здн}$  – задане значення температури вина  $^{\circ}C$ ;

$\Delta T_{конд}$  – помилка регулювання температури конденсату;

$\Delta T_{вина}$  – помилка регулювання температури вина;

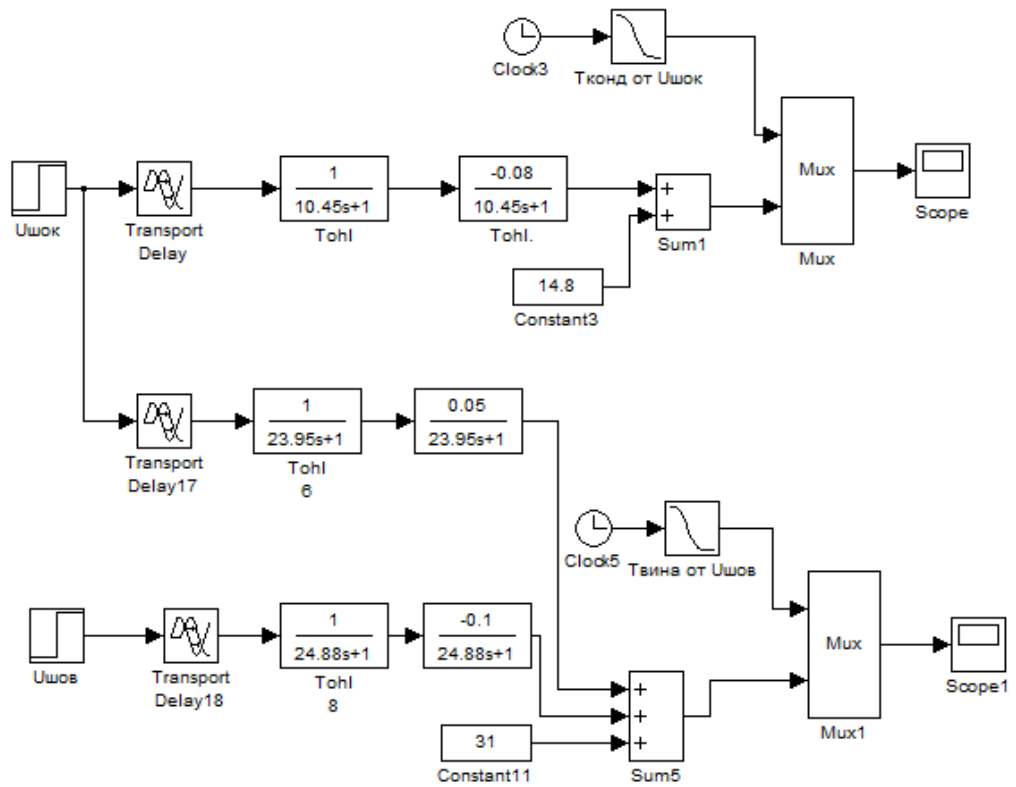


Рис. 1.5 – Схема моделювання повної моделі об'єкта

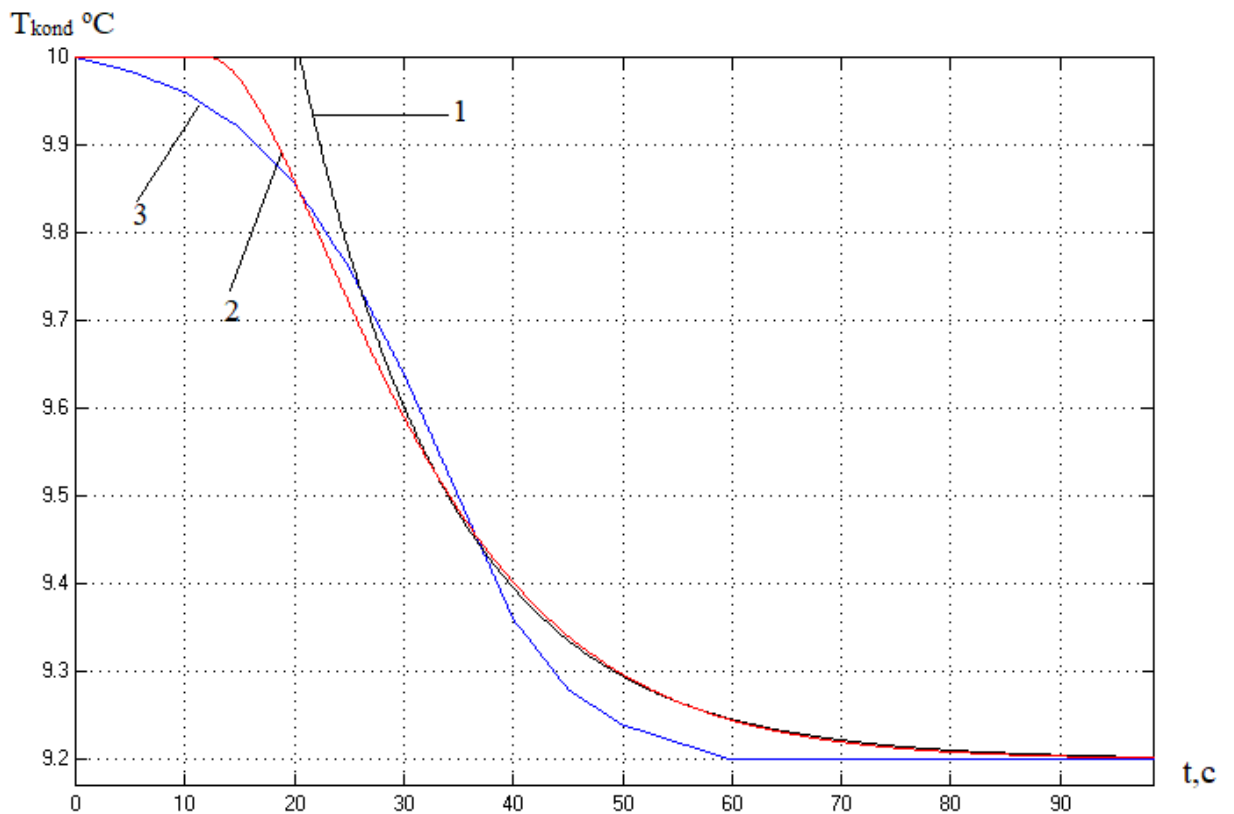


Рис. 1.6 – Результат моделювання каналу « $U_{шок} - T_{конд}$ », де 1- модель 1-го порядку; 2- модель 2-го порядку ; 3- експериментальні данні.

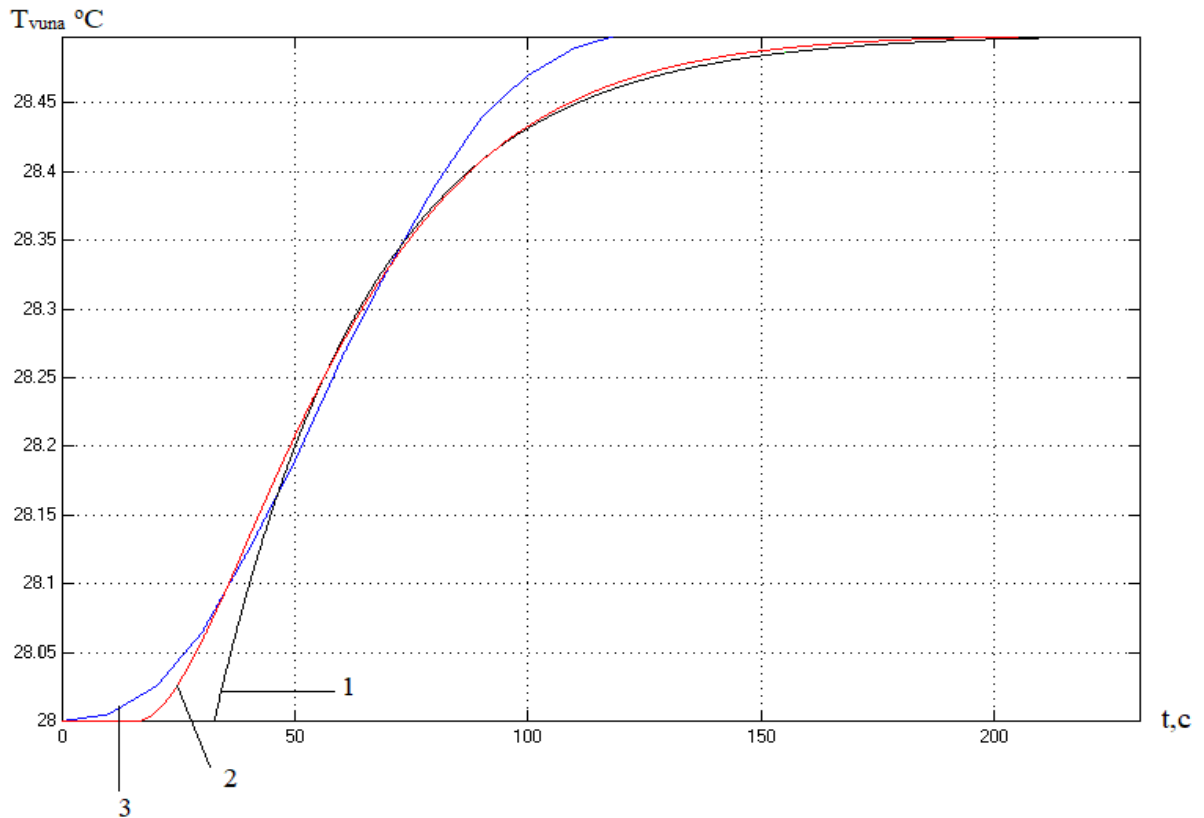


Рис. 1.7 – Результат моделювання каналу « $U_{\text{шок}} - T_{\text{вина}}$ », де 1- модель 1-го порядку; 2- модель 2-го порядку ; 3- експериментальні данні.

Типова структура і алгоритми САУ ТП сушіння плодоовочевої сировини, які реалізують функцію регулювання.

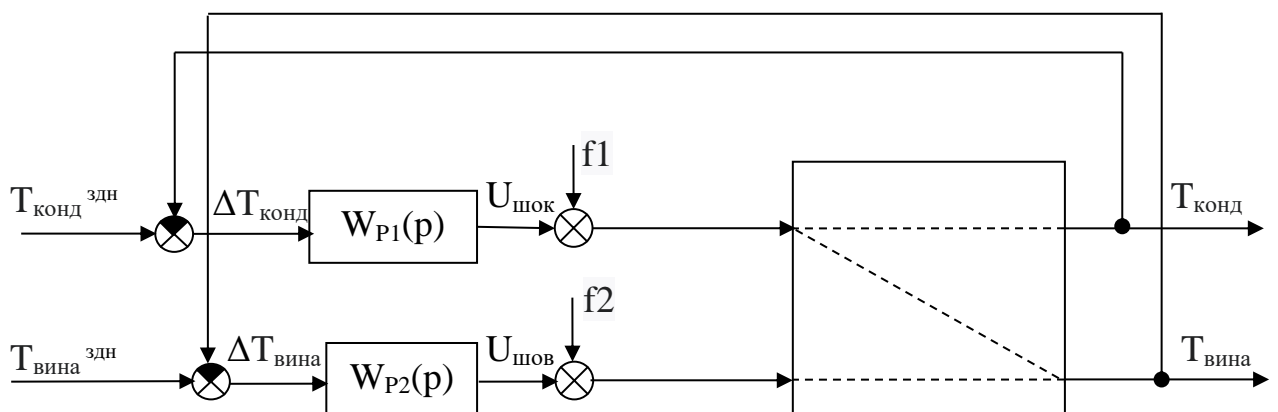


Рис. 1.8 – Структурна схема САУ процесу вакуумної деалкоголізації вина в потоці базової структури

В якості алгоритму регулювання обрано пропорційно-інтегрально-диференційний алгоритм регулювання (ПІД-регулятор).

Диференційне рівняння:

$$u(t) = k_p(\Delta y(t) + \frac{1}{T_{із}} \int_0^t \Delta y(t) dt + T_{пр} \frac{d\Delta y}{dt}) + u_0;$$

Передатна функція ПІД регулятора:

$$W^P(p) = K_p \cdot \left( 1 + \frac{1}{T_{із}p} + \frac{T_{пр}p}{0,2 \cdot T_{пр}p + 1} \right)$$

$k_p$  – коефіцієнт передачі регулятора;

$T_{із}$  – час ізодрому;

$T_{пр}$  – час попередження.

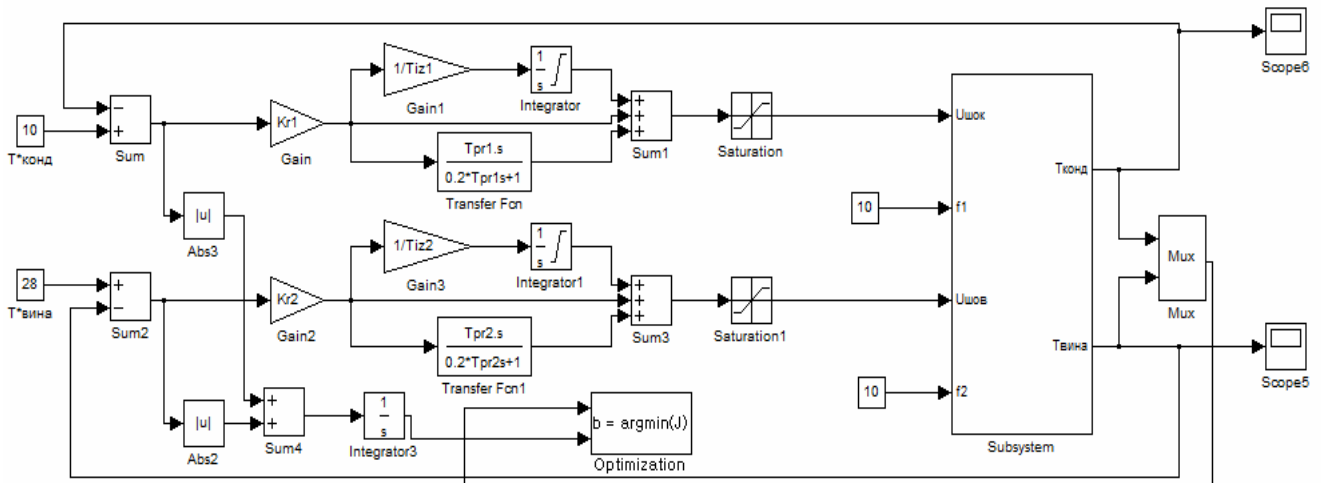


Рис. 1.9 Структурна схема моделювання САР для оптимізації налаштувань ПІД-регулятора

Таблиця 1 - Прямі показники якості

Канали	$\Delta y^{\max}$	$t_{пер}$
ПІД-регулятор		
« $U_{шок} - T_{конд}$ »	0,45	50
« $U_{шов} - T_{вина}$ »	0,75	150

Перевірка на грубість САК з оптимальними настройками ПІД-регуляторів.

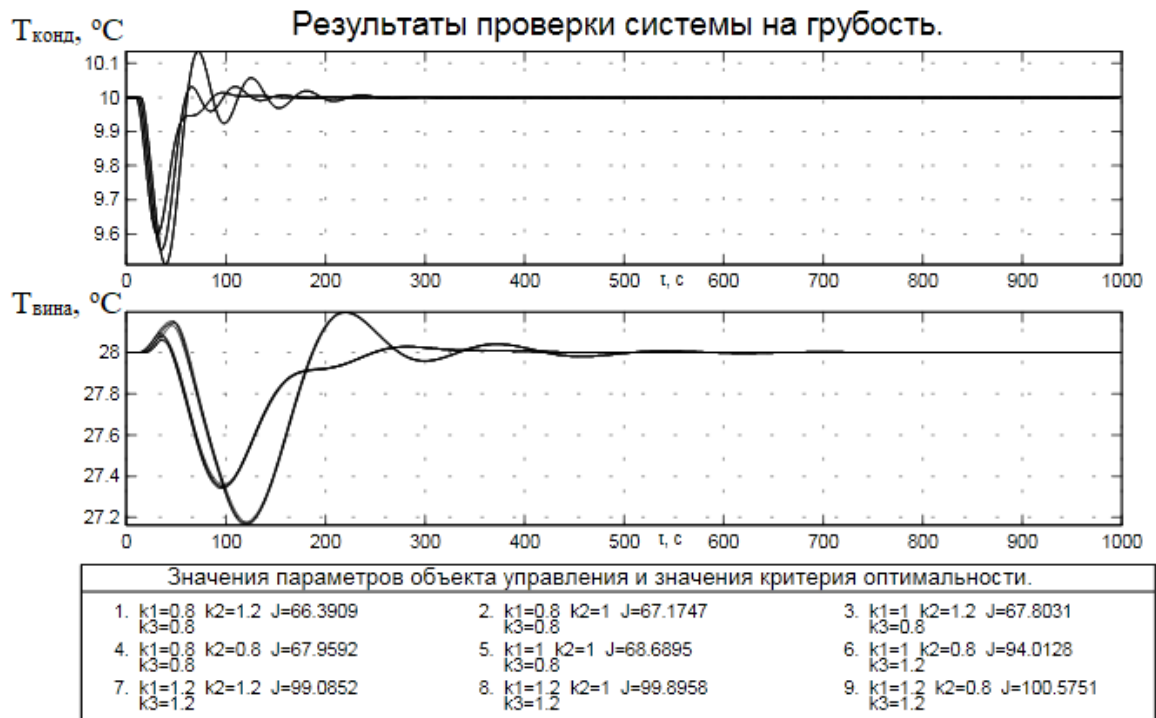


Рис. 1.10 - Перевірка на грубість САК

Система груба.

1.5 Недоліки існуючих САК ТП вакуумної деалкоголізації вина, обґрунтування напрямлень вдосконалення САК, як нових джерел підвищення економічної ефективності технологічного процесу.

Недоліком є низька точність стабілізації температури конденсату, що призводить до браку готової продукції.

### 1.6 Висновки

Виходячи з вищевказаних обґрунтувань, доцільно розробити для даного процесу абсолютно нову систему, яка забезпечить високу ефективність. Тому задача синтезу САК підвищеної динамічної точності повинна бути вирішена.

## РОЗДІЛ 2

### ВДОСКОНАЛЕННЯ АЛГОРИТМІВ, ЩО РЕАЛІЗУЮТЬ ФУНКЦІЇ РЕГУЛЮВАННЯ САК ВАКУМНОЇ ДЕАЛКОГОЛІЗАЦІЇ ВИНА

2.1 Специфічні особливості ОР та обґрунтування вибору методів корекції САР для підвищення їх динамічної точності.

Для САР деалкоголізації вина основною причиною недостатньої динамічної точності САР є дія перекрестного зв'язку. Основним шляхом підвищення динамічної точності саме розглянутої САР є побудова САР, інваріантної до дії перекрестного зв'язку.

2.2 Структурний синтез САР підвищеної динамічної точності.

В основу підвищення динамічної точності САР буде покладено принцип двоканальний Петрова, суть якого у введенні додаткового каналу впливу " $U_{\text{шов}} - T_{\text{вина}}$ " через коригуючий зв'язок. Відповідно до принципу інваріантності Петрова структурна схема САР, інваріантної до дії контрольованого збурення матиме вигляд, наведений на рис. 2.1.

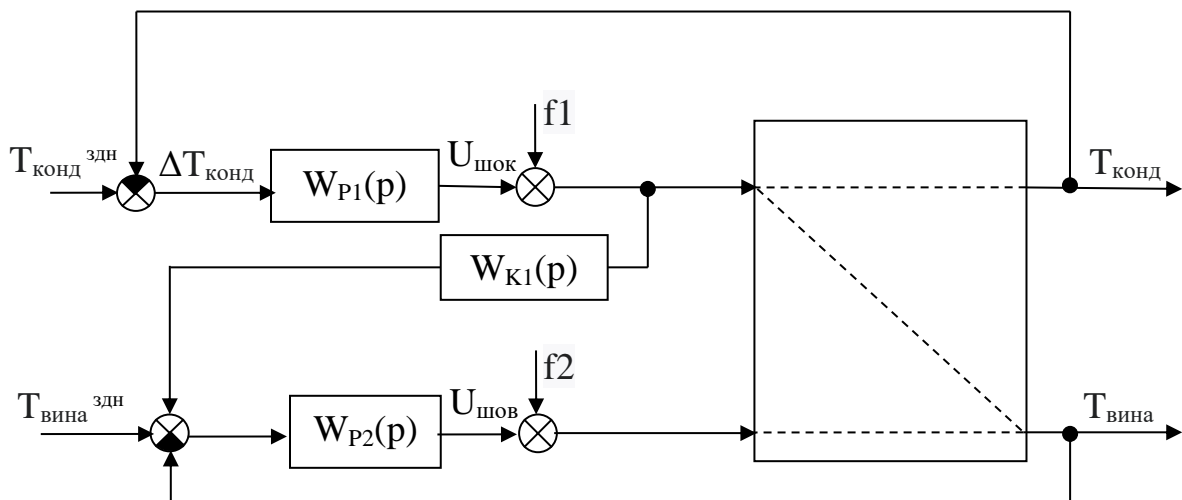


Рис. 2.1 – Структурна схема САР підвищеної динамічної точності  
На схемі  $W_{K1}(p)$  – передаточна функція корегуючого зв'язку.  
Необхідною умовою реалізації САР є фізична можливість організації додаткового каналу впливу за перекрестного зв'язку, а для розглянутої САР така можливість є. Достатньою умовою реалізації САР є фізична реалізація передаточної функції коригуючого зв'язку.

Для забезпечення інваріантності до дії контрольованого збурення САР каналу « $U_{\text{шов}} - T_{\text{вина}}$ » повинна забезпечити помилку стабілізації, що має дорівнювати нулю:

$$\Delta T_{\text{вина}} = 0$$

Відомо, що

$$\Delta T_{\text{вина}} = W_{k1}^C(p) \cdot T_{\text{конд}} = 0$$

Відомо, що  $T_{\text{конд}} \neq 0$ , тоді

$$W_{k1}^C(p) = 0$$

$$W_{k1}^C(p) = W_{T_{\text{вина}}-T_{\text{конд}}}^o(p) + W_K 1(p) * W_{p2}^p(p) * W_{T_{\text{вина}}}^o(p) = 0$$

Звідси

$$W_K 1(p) = - \frac{W_{T_{\text{вина}}-T_{\text{конд}}}^o(p)}{W_{p2}^p(p) * W_{T_{\text{вина}}}^o(p)}$$

Якщо

$$W_{U_{\text{шок}}-T_{\text{вина}}}^o(p) = \frac{0,05 \cdot e^{-32,56p}}{33,97p + 1}, W_{U_{\text{шов}}-T_{\text{вина}}}^o(p) = \frac{0,1 \cdot e^{-54,635p}}{36,61p + 1}$$

$$W_p(p) = -6,7 \cdot \left( 1 + \frac{1}{109,27p} + 27,31p \right)$$

Тоді

$$\begin{aligned} W_K 1(p) &= \frac{\frac{0,05 \cdot e^{-32,56p}}{33,97p + 1}}{-6,7 \cdot \left( 1 + \frac{1}{109,27p} + 27,31p \right) \cdot \frac{0,1 \cdot e^{-54,635p}}{36,61p + 1}} = \\ &= e^{(-32,56 - (-54,635))p} * \frac{0,05}{0,1 * (-6,7)} * \\ &\quad * \frac{1}{33,97p + 1} = \\ &= \frac{2984,2p^2 + 109,27p + 1}{109,27p} * \frac{1}{36,61p + 1} = \\ &= -0,007 \cdot e^{22p} \cdot \frac{109,27p * (36,61p + 1)}{(2984,2p^2 + 109,27p + 1) * (33,97p + 1)} \end{aligned}$$

Отримана передаточна функція коригуючого зв'язку є фізично нереалізовуваною, оскільки містить ланку чистого випередження. Для

забезпечення фізичної реалізованості передаточної функції проведемо заміну ланки чистого випередження форсуючою ланкою:

$$e^{22p} = 1 + \frac{22p}{0,22p+1}$$

Тоді передаточна функція фізично реалізованого коригуючого зв'язку матиме вигляд:

$$W_k(p) = -0,007 * \left(1 + \frac{22p}{0,22p+1}\right) * \frac{4000p^2+109,27p}{101373p^3+3712p^2+143p+1}$$

Структурна схема моделювання передаточної функції коригуючого зв'язку наведена на рис. 3.29, а її перехідна характеристика – на рис. 2.2.

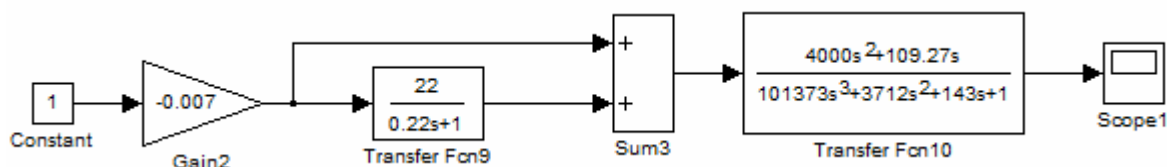


Рис. 2.2 – Структурна схема моделювання передаточної функції коригуючого зв'язку

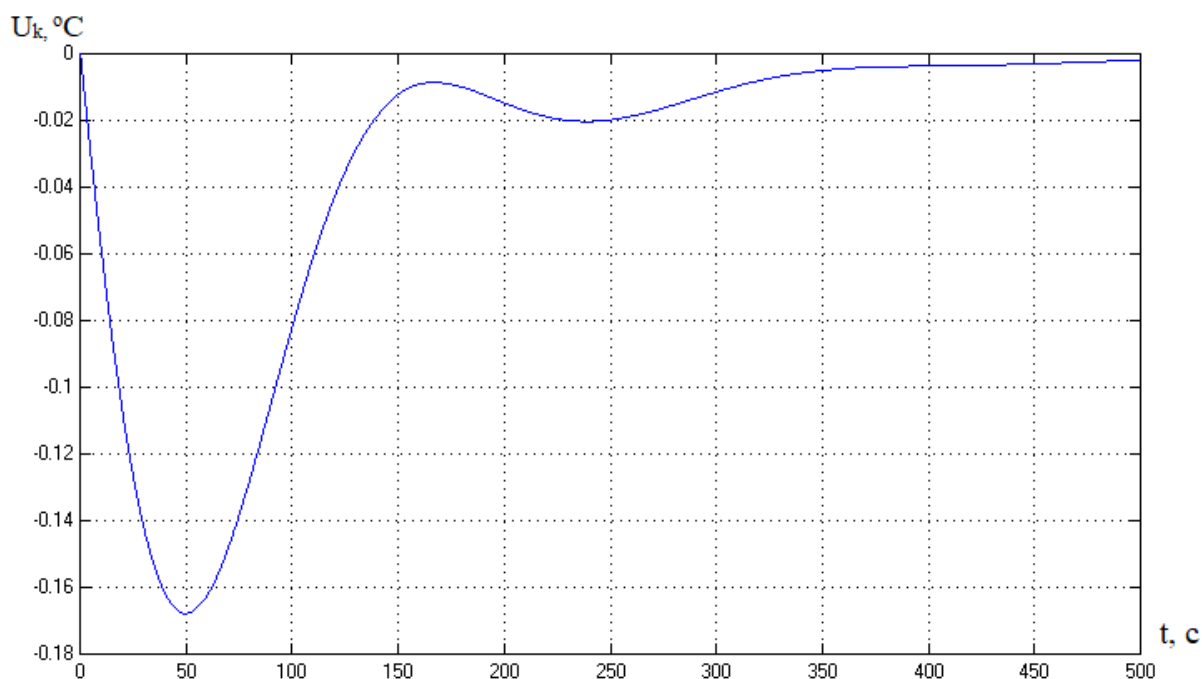


Рис. 2.3 – Перехідна характеристика коригуючого зв'язку

Отримана передаточна функція коригуючого зв'язку представлена занадто складною передаточною функцією, тому її доцільно спростити при

збереженні її диференціюючих властивостей. Спрощена передаточна функція коригуючого зв'язку матиме вигляд:

$$W_K(p) = K_K \cdot e^{-T_{K1}p} \cdot \frac{T_{K2}p}{(T_{K2}p+1)^2}$$

В якості початкових наближень параметрів коригуючого зв'язку приймемо такі значення параметрів:  $K_K = -0,007$ ;  $T_{K1} = 22$ ;  $T_{K2} = 109,27$ . Для уточнення початкових наближень параметрів коригуючого зв'язку скористаємося схемою моделювання, наведеної на рис. 3.31. Результати розрахунку початкових наближень параметрів коригуючого зв'язку наведені на рис. 3.32, а результати моделювання – на рис. 3.33.

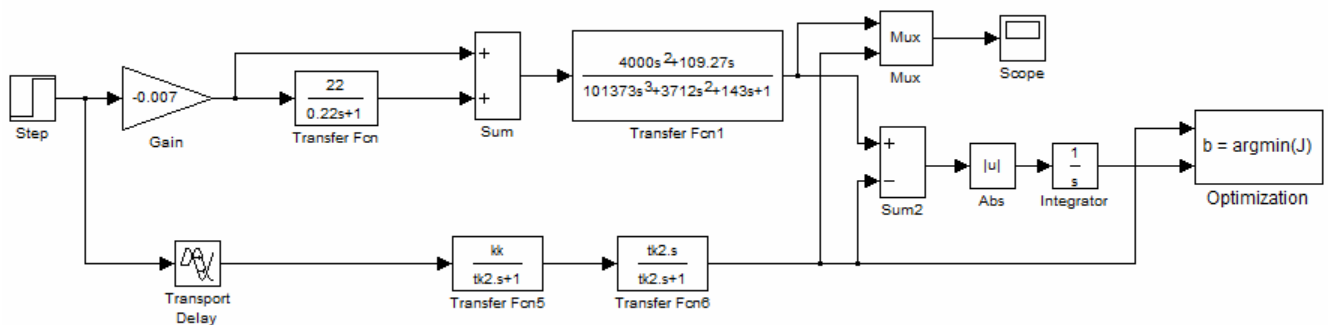
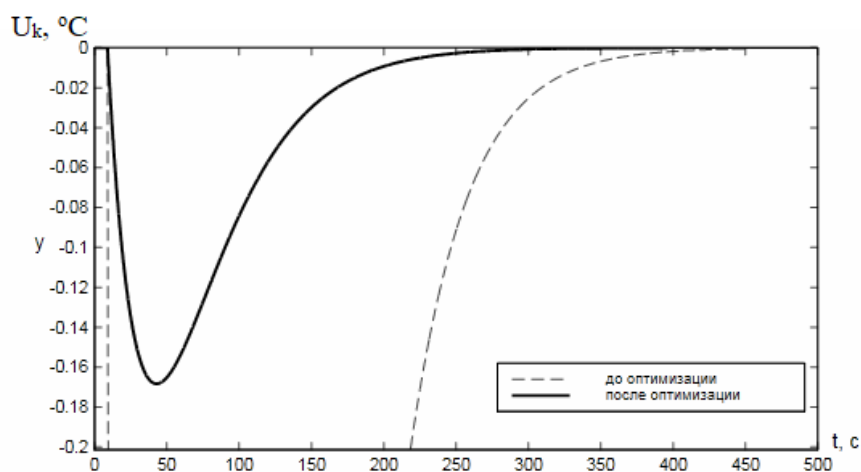


Рис. 2.4 – Схема моделювання для уточнення початкових наближень параметрів коригуючого зв'язку



Імя перемінної	Оптимальне значення	Нижній предел	Начальне приближення	Верхній предел	Значення критерія оптимальності:
kk	-0.45671	-100	-15.5352	100	до - 511.8905
tk1	7.9481	0	7.9495	100	після - 4.3806
tk2	34.0182	-10	34.011	300	оптимізації.
					Кількість шагів процедури оптимізації:
					максимальное - 1000
					фактическое - 214

Рис. 2.5 – Результати розрахунку початкових наближень параметрів коригуючого зв'язку

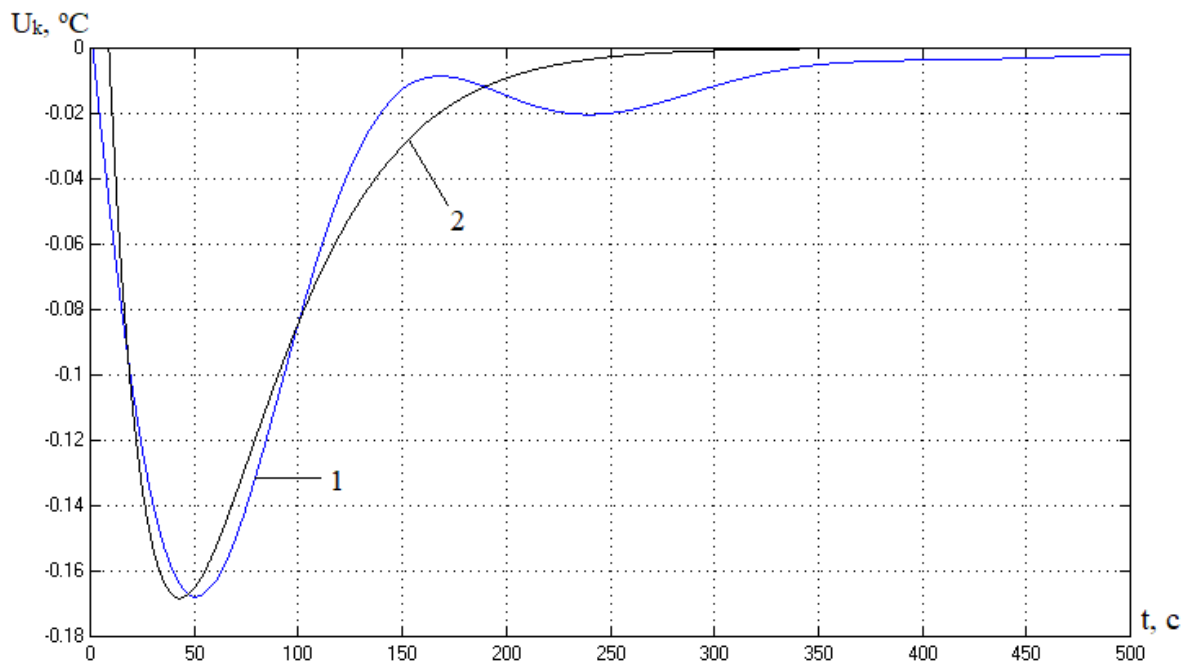


Рис. 2.6 – Перехідні характеристики розрахункового 1 та спрощеного 2 коригуючого зв'язку

Схема моделювання САР підвищеної динамічної точності для оптимального параметричного синтезу коригуючого зв'язку приведена на рис. 3.34, а результати оптимізації - на рис. 3.35.

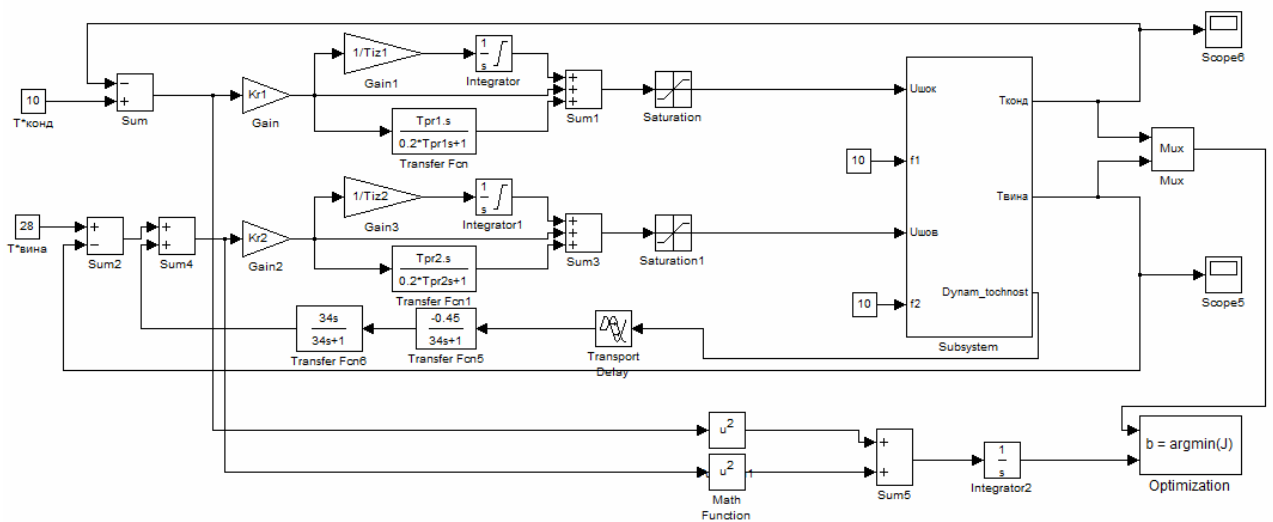


Рис. 2.7 – Схема моделювання САР підвищеної динамічної точності для оптимального параметричного синтезу коригуючого зв'язку

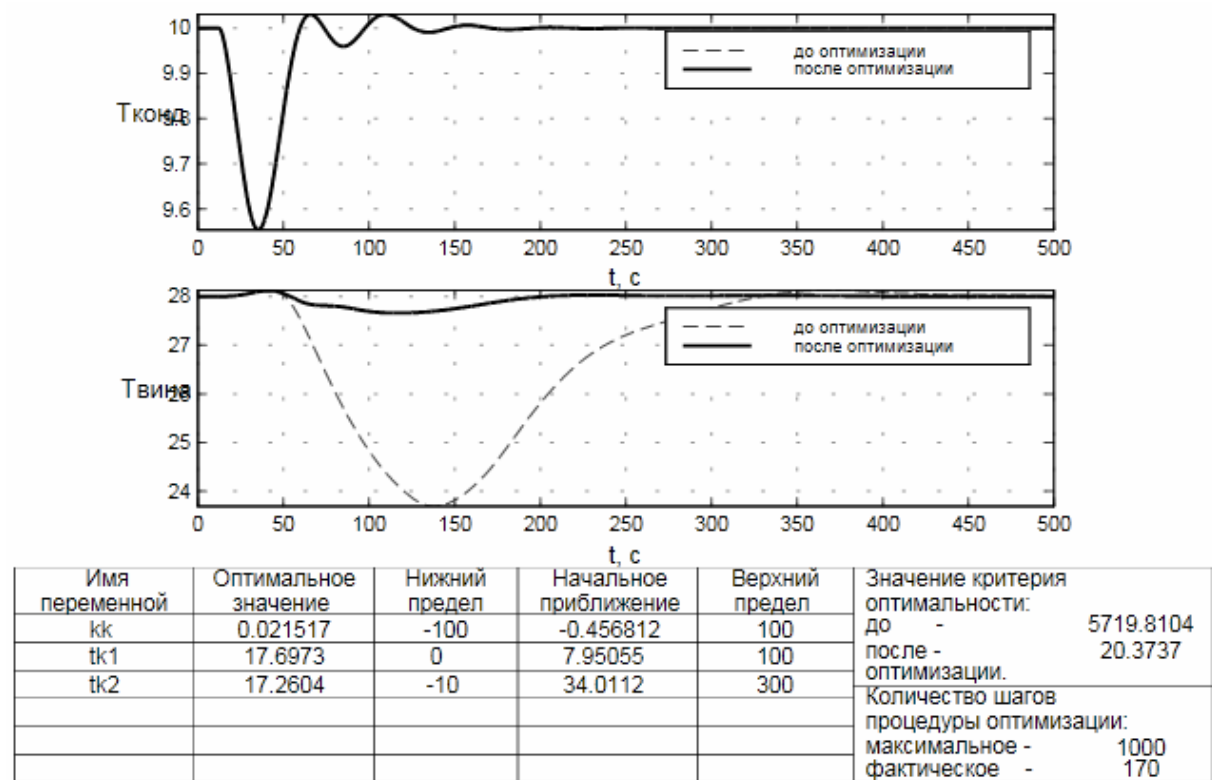


Рис. 2.8 – Результати оптимізації параметрів коригуючого зв'язку

Проведемо аналіз грубості САР підвищеної динамічної точності до варіацій параметрів об'єкта регулювання

Як і в випадку з САР базової структури, САР підвищеної динамічної точності досліджуємо на грубість в умовах варіації параметрів ОК, а саме, часу запізнення в каналах ОК  $\pm 20\%$ .

Результати перевірки САР підвищеної динамічної точності на грубість в умовах варіації параметрів ОК наведено на рис. 3.36. Як видно з результатів оцінки САР на грубість, в умовах варіації параметрів ОК САР дає перехідні процеси, що сходяться, а значить, САР є грубою.

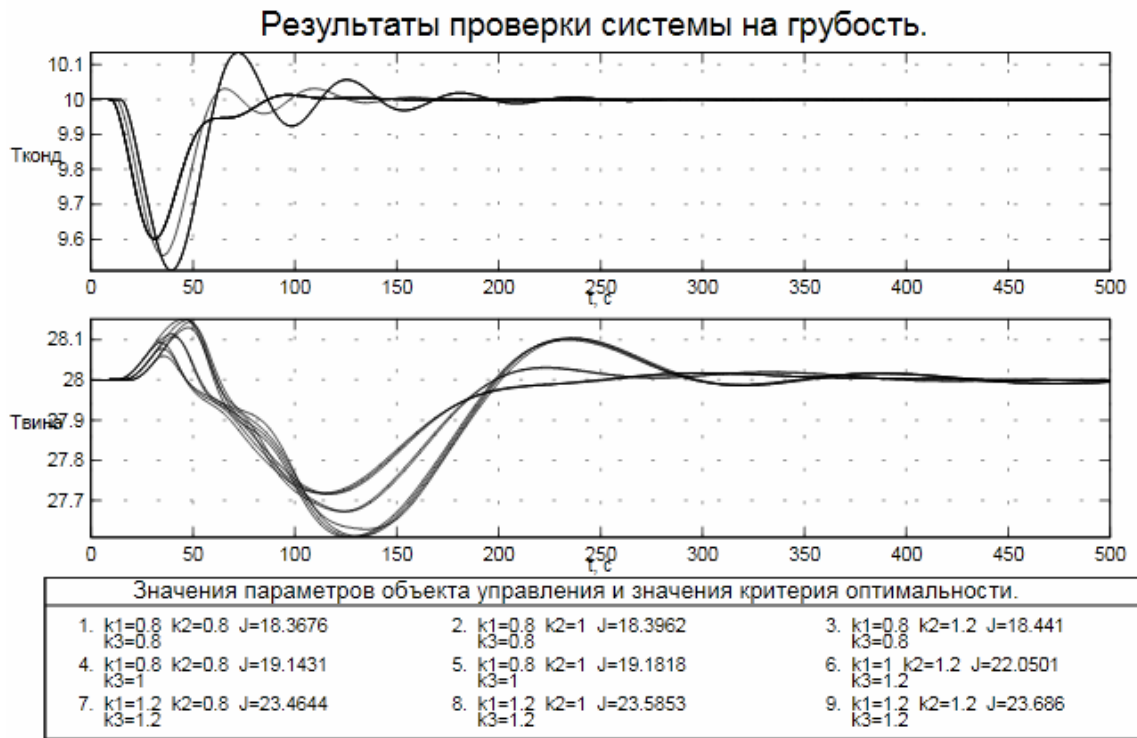


Рис. 2.9 – Аналіз САР підвищеної динамічної точності на грубість в умовах варіації параметрів ОК

Як видно з результатів оцінки САР на грубість, найбільш "несприятливим" для керування є зменшення часу запізнення на 10% за каналом « $U_{\text{шוק}} - T_{\text{конд}}$ » збільшення часу запізнення на 20% за каналами « $U_{\text{шок}} - T_{\text{вина}}$ », « $U_{\text{шов}} - T_{\text{вина}}$ ». Найбільш "сприятливим" для керування є більший на 10% час запізнення каналу « $U_{\text{шок}} - T_{\text{конд}}$ » та зменшення на 20% часу запізнення за каналами « $U_{\text{шок}} - T_{\text{вина}}$ », « $U_{\text{шов}} - T_{\text{вина}}$ ».

### 2.3 Висновки

Як видно з результатів аналізу САР на грубість, САР підвищеної динамічної точності є грубою, тому що при будь-яких поєднаннях параметрів ОУ дає перехідні процеси, що сходяться і які, до того ж, не виходять за рамки зони незначних відхилень.

## РОЗДІЛ 3

### ВДОСКОНАЛЕННЯ АЛГОРИТМІВ, ЯКІ РЕАЛІЗУЮТЬ ФУНКЦІЇ РЕГУЛЮВАННЯ САК ТП СУШННЯ ПЛОДООВОЧЕВОЇ СИРОВИНИ, НА ОСНОВІ ВИКОРИСТАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

3.1. Обґрунтування актуальності застосування систем керування, побудованих на базі апарату нечіткої логіки, для розглянутого об'єкта керування

Особливою проблемою в сфері автоматизації є керування об'єктами, функції яких описуються нелінійними залежностями. При проектуванні систем керування для нелінійних об'єктів зазвичай використовуються такі методи як лінійна апроксимація або кусочно-лінійна апроксимація. Однак застосування цих методів у реальних технічних пристроях не завжди дозволяє досягнути бажаного ефекту. У цьому випадку для забезпечення збільшення ефективності можна використовувати систему керування з нечіткою логікою («fuzzy logic»).

Нечітке керування ефективне в умовах невизначеності інформації про динамічну поведінку складних об'єктів керування. Доцільність використання нечіткого підходу для побудови грубих регуляторів впливає з розгляду особливостей нечіткого керування. Основна перевага нечіткого підходу – можливість формування числа правил керування залежно від комбінації значень вхідних змінних регулятора і, отже, від зміни режиму роботи, рівня збурювань.

Для технічних систем з випадковим характером впливу, що збурює, складністю розробки динамічної моделі, її високим порядком, та нелінійним характером можна говорити про проблему керування в умовах невизначеності. Використання нечітких регуляторів забезпечує грубість і стабільну збіжність процесів, тому такий підхід слід вважати доцільним.

Таким чином, в умовах наявності істотних невизначеностей у статичних і динамічних характеристиках плівкового дистилятора, як об'єкта керування, представляються труднощі в формалізації завдань синтезу регуляторів системи керування. У зв'язку з цим доцільним є застосування регуляторів, заснованих на нечіткій логіці.

У якості переваги нечіткого регулювання можна також відзначити наявність сучасних систем програмування контролерів з вбудованими бібліотеками нечіткого управління, що мають добрий графічний інтерфейс, у якому дуже легко і наочно представляється і коректується вид функцій приналежності і нечіткого висновку. Отже, спрощується і настроювання системи автоматичного регулювання.

3.2. Реалізація в програмному середовищі MATLAB\Simulink моделей каналів регулювання розглянутого об'єкта керування, що характеризується нелінійними статичними характеристиками

Для підтвердження актуальності застосування системи керування, побудованої на базі апарата нечіткої логіки, для розглянутого у даній роботі об'єкта керування, необхідно перейти до його нелінійної моделі. Таким чином, моделі каналів регулювання розглянутого об'єкта керування повинні характеризуватися несуттєво нелінійною статичною характеристикою.

Спочатку представимо несуттєво нелінійну статичну характеристику об'єкта керування для переходу від лінійної моделі об'єкта керування до первісної нелінійної. На рисунку 3.1 представлена екстремальна статична характеристика об'єкта керування за каналом « $U_{\text{шов}} - T_{\text{вина}}$ » для того, щоб переконатися у присутності нелінійних властивостей об'єкта.

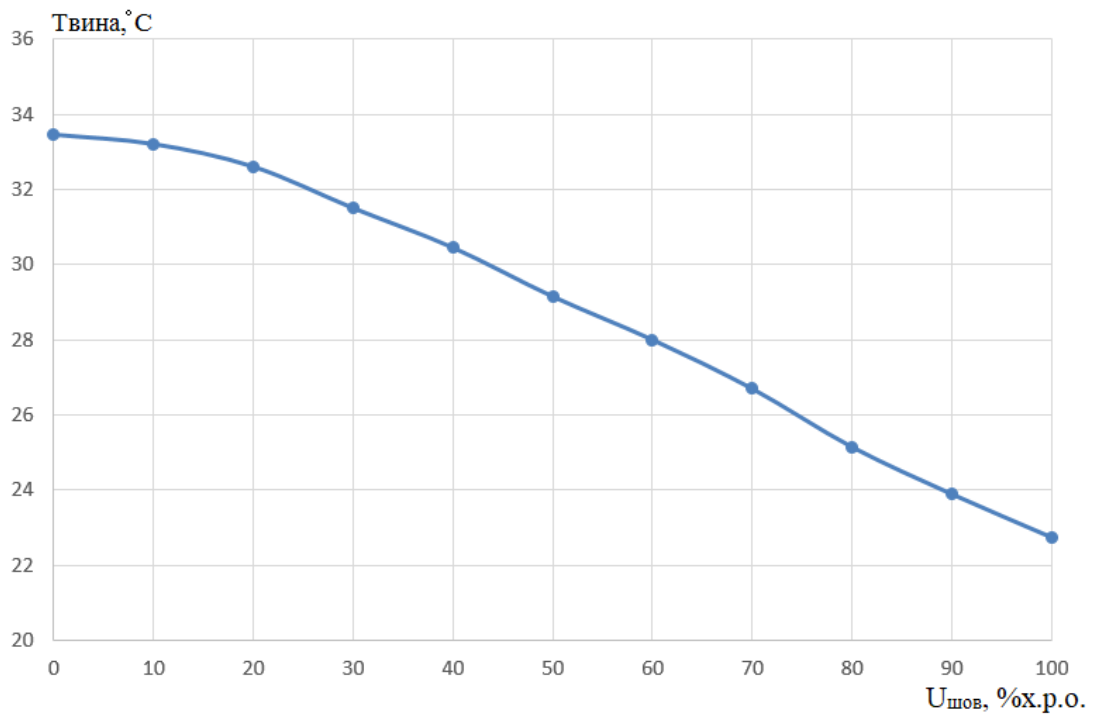


Рис. 3.1 – Статична характеристика об'єкта керування за каналом «керуюча дія  $U_{\text{шов}}$  – температура вина  $T_{\text{вина}}$ , °C»

Нелінійність реалізовано за допомогою включення блоку Look up Table у модель каналу регулювання « $U_{\text{шов}}$  –  $T_{\text{вина}}$ ». Структурну схему повної моделі об'єкта керування з додатковими блоком Look up Table, який визначає нелінійність статичної характеристики, представлено на рисунку 3.2.

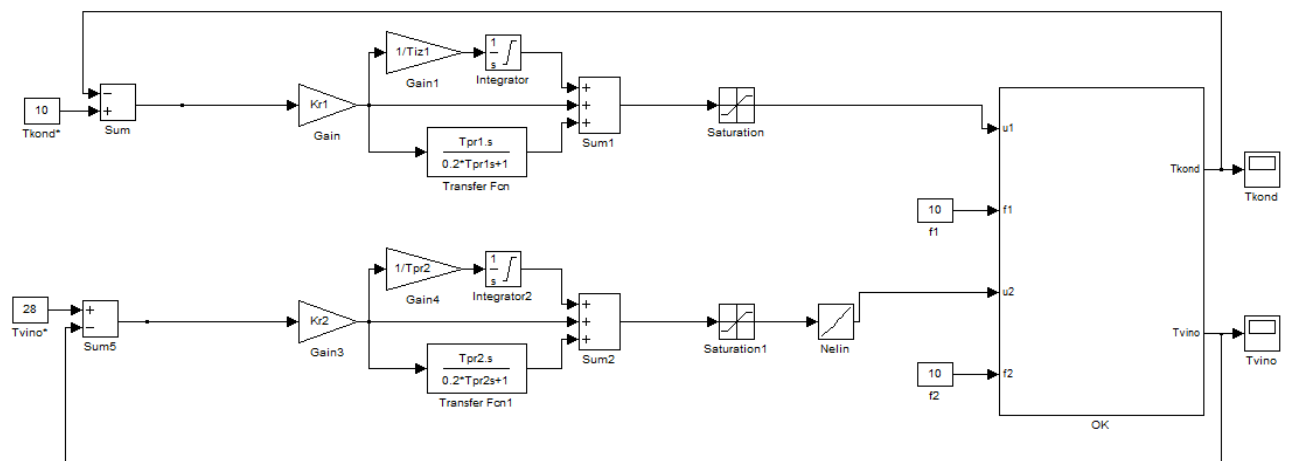


Рис. 3.2 – Структурна схема моделі об'єкта керування, що характеризується нелінійною статичною характеристикою за каналом регулювання « $U_{\text{шов}}$  –  $T_{\text{вина}}$ »

У блоці Look up Table задано нелінійний зв'язок (залежність) між вхідними змінними, який тягне «викривлення» статичної характеристики моделі каналу регулювання « $U_{\text{шов}} - T_{\text{вина}}$ ».

В результаті моделювання була отримана статична характеристика моделі каналу регулювання « $U_{\text{шов}} - T_{\text{вина}}$ », яка має вигляд, представлений на рисунку 3.3.

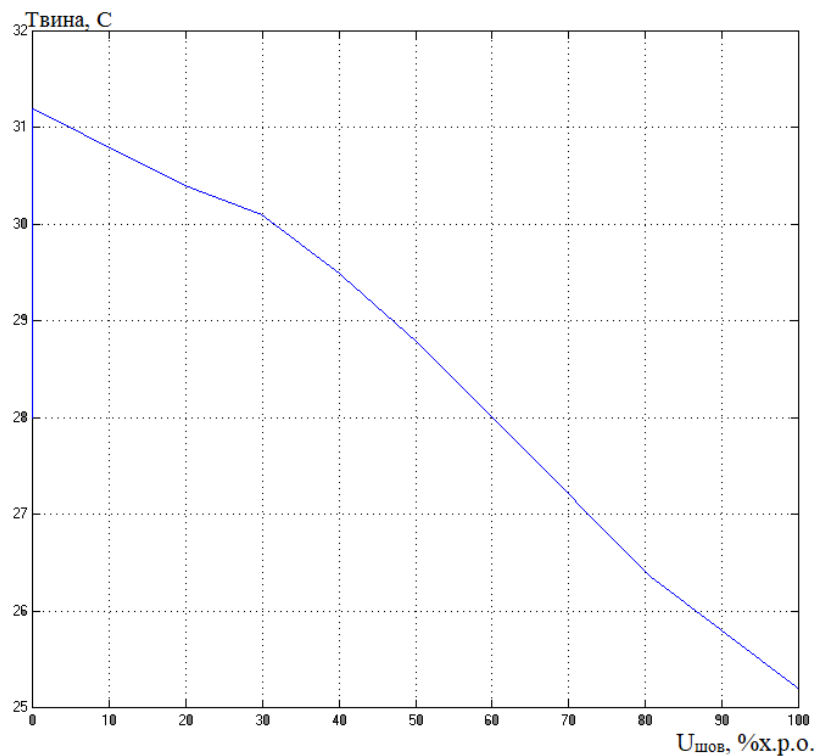


Рис 3.3 – Нелінійна статична характеристика моделі каналу регулювання « $U_{\text{шов}} - T_{\text{вина}}$ »

З рисунку 3.3 видно, що статична характеристика несуттєво нелінійна і схожа зі статичною характеристикою, представленою на рисунку 3.1. Це дає можливість виконати синтез САР з традиційними ПД-регуляторами і одержати необхідні показники якості регулювання.

### 3.3 Параметричний синтез САР з традиційним ПІД-регулятором для об'єкта з нелінійною статичною характеристикою каналу регулювання

У даній роботі був виконаний параметричний синтез САР з традиційним лінійним ПІД-регулятором для об'єкта з нелінійною статичною характеристикою за каналом регулювання « $U_{\text{шов}} - T_{\text{вина}}$ ». Слід зазначити, що нелінійність статичної характеристики була реалізована в моделі, «гладкої» і несуттєвої для одержання стійких перехідних процесів при застосуванні традиційних лінійних (П, ПІ, ПІД) регуляторів.

Параметри настроювання регуляторів були визначені на основі статичних і динамічних параметрів об'єкта управління в рамках лінійної моделі.

Для реалізації параметричної оптимізації регуляторів системи керування в програмному середовищі MATLAB\Simulink 5.2 була розроблена схема моделювання системи керування з автоматичним оптимізатором, що забезпечує пошук оптимальних параметрів регуляторів за наступним інтегральним модульним критерієм якості роботи системи:

$$J = \int_0^{t_M} [ |(\Delta T_{\text{конд}}(t))| + |(\Delta T_{\text{вина}}(t))| ] \cdot dt$$

де  $t_M$  – час моделювання;

$\Delta T_{\text{конд}}(t)$  – відхилення температури конденсату від заданого значення.

$\Delta T_{\text{вина}}(t)$  – відхилення температури вина від заданого значення.

Розроблена схема моделювання системи керування з автоматичним оптимізатором у середовищі MATLAB\Simulink 5.2 представлена на рисунку 3.1.

У результаті параметричної оптимізації були отримані графіки перехідних процесів до і після оптимізації і відповідні параметри настроювання регуляторів. Первісні значення параметрів настроювання регуляторів і відповідні оптимальні значення представлені на рисунку 3.2 у вікні результатів оптимізації. Також на рисунку 3.2 представлені відповідні

графіки перехідних процесів, отримані до і після оптимізації системи. Також на рисунку 3.4 показано аналіз САР на грубість в умовах варіації  $\pm 20\%$  часу запізень ОК.

Із графіків перехідних процесів видно, що якість регулювання, за каналом « $U_{\text{шов}} - T_{\text{вина}}$ », помітно покращилася після оптимізації САР. З аналізу САР на грубість можна помітити, що САР є грубою. Однак слід зазначити, що при нелінійній характеристиці каналу регулювання такі покращення можуть мати місце лише при відповідних режимах роботи чи початкових умовах системи керування. Таким чином, оптимальні параметри настроювання в даному випадку можуть бути неоптимальні в іншому випадку, це визначає необхідність застосування нелінійних регуляторів.

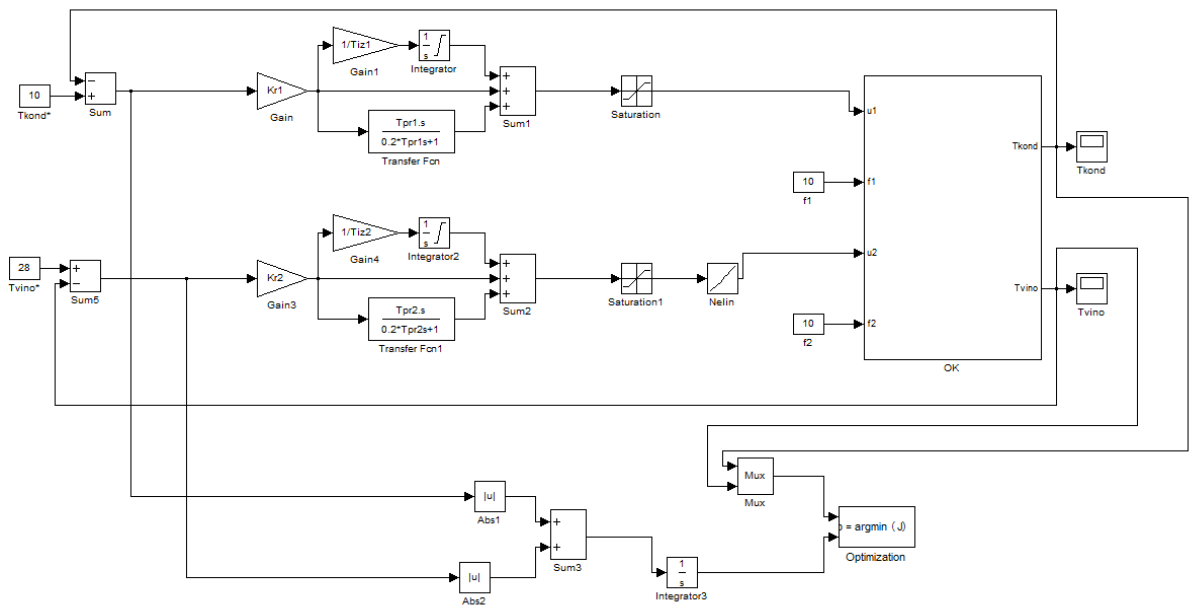
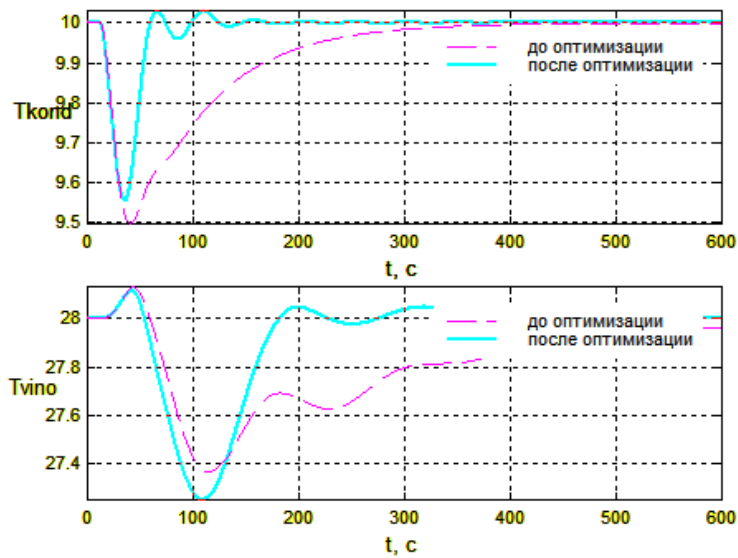


Рис. 3.4 – Структурна схема моделі САР з традиційним ПД-регулятором і параметричним оптимізатором, що реалізована засобами MATLAB\Simulink для визначення оптимальних параметрів системи керування за умови нелінійності статичної характеристики каналу регулювання « $U_{\text{шов}} - T_{\text{вина}}$ »



Имя переменной	Оптимальное значение	Нижний предел	Начальное приближение	Верхний предел	Значение критерия оптимальности:
Kr1	-18.6856	-50	-8.5	50	до - 173.0407
Tiz1	20.8812	1	40.86	200	после - 73.9275
Trp1	7.5032	1	10.215	200	оптимизации.
Kr2	-8.0745	-50	-6.7	50	Количество шагов
Tiz2	50.3517	1	109.27	500	процедуры оптимизации:
Trp2	21.9935	1	27.31	200	максимальное - 1000
					фактическое - 473

Рис. 3.5 – Результати параметричної оптимізації значень параметрів регуляторів САР за умови нелінійності статичної характеристики каналу регулювання « $U_{\text{шов}} - T_{\text{вина}}$ »

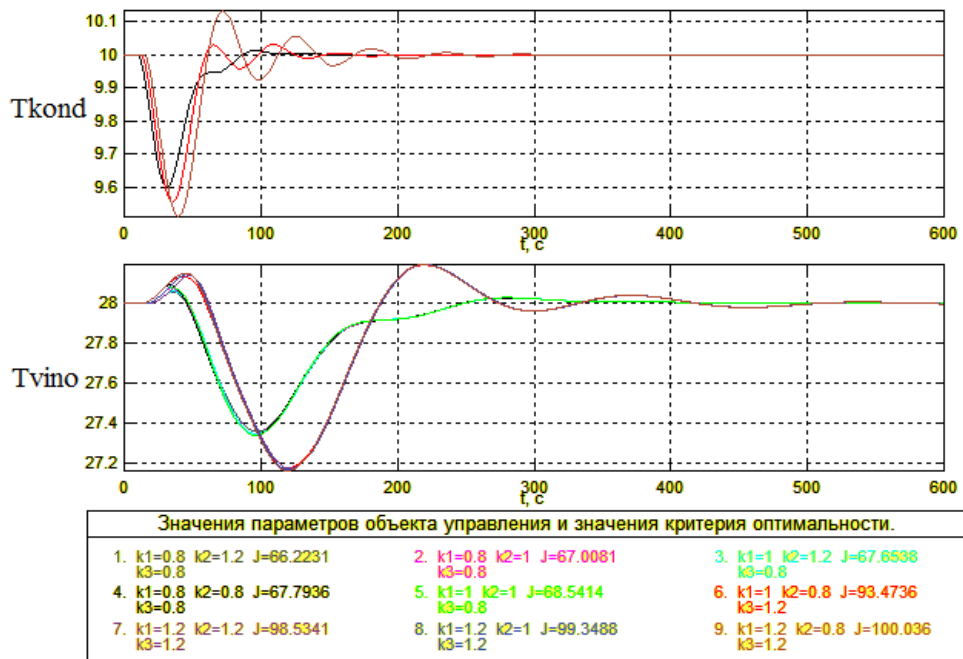


Рис. 3.6 – Результати перевірки САР на грубість за умови нелінійності статичної характеристики каналу регулювання « $U_{\text{шов}} - T_{\text{вина}}$ »

### 3.4. Розробка моделі САР з нечітким регулятором (НР) для об'єкта з нелінійною статичною характеристикою каналу регулювання

З урахуванням властивостей плівкового дистилятора, як нелінійного об'єкта керування, доцільно застосувати в контурі регулювання « $U_{\text{шов}} - T_{\text{вина}}$ » нечіткий логічний регулятор (НЛР), який має властивості нелінійного регулятора. Застосування нечіткого-логічного регулятора (Fuzzy-регулятора) обумовлено наявністю нелінійного об'єкта керування, а також складного опису його статичних режимів роботи.

Складність опису статичних режимів роботи плівкового дистилятора приводить до того, що режими функціонування можуть бути описані з недостатньою точністю, пов'язаною з нечіткими (розмитими) різними експериментальними даними. Таким чином, у справжній роботі була реалізована модель САР із Fuzzy – нечітким регулятором.

Нечіткий регулятор, що розробляється у даній роботі, являє собою об'єднання на деякій елементній базі трьох блоків фазікерування: фазифікація, логічного висновку і дефазифікація. Спрощена структурна схема розробляємої системи керування на базі нечіткого регулятора представлена на рисунку 4.1. З погляду структури дана схема регулятора нічим не відрізняється від схем класичних нечітких регуляторів і будується за класичними зразками.

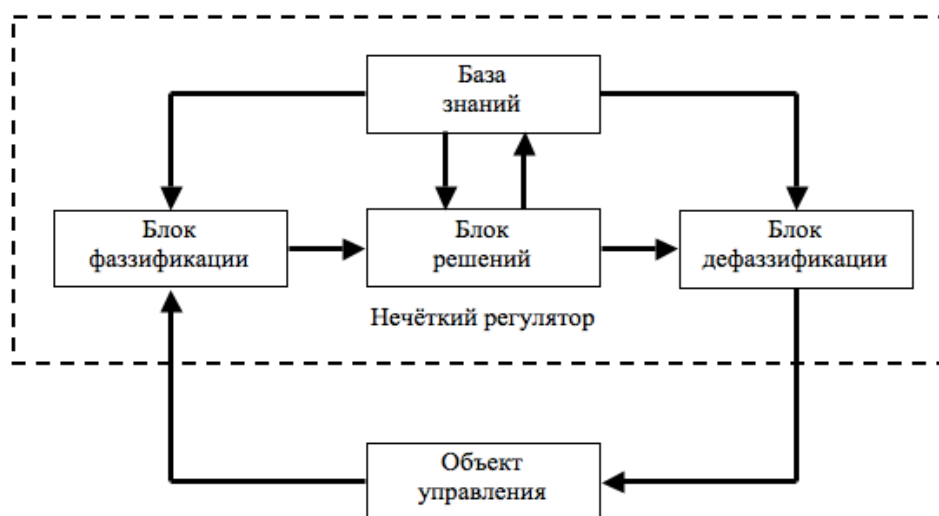


Рис. 3.7 – Спрощена структурна схема нечіткого логічного регулятора

Всі визначені блоки (фазифікація, блок логічного розв'язку, дефазифікація), що представлені на рисунку 4.1, редагуються у відповідному редакторі FIS Editor середовища MATLAB.

На рисунку 3.8 зображено вікно налаштування fis-файлу для нечіткого регулятора температури в розстойній шафі з відповідними вхідними та вихідними сигналами.

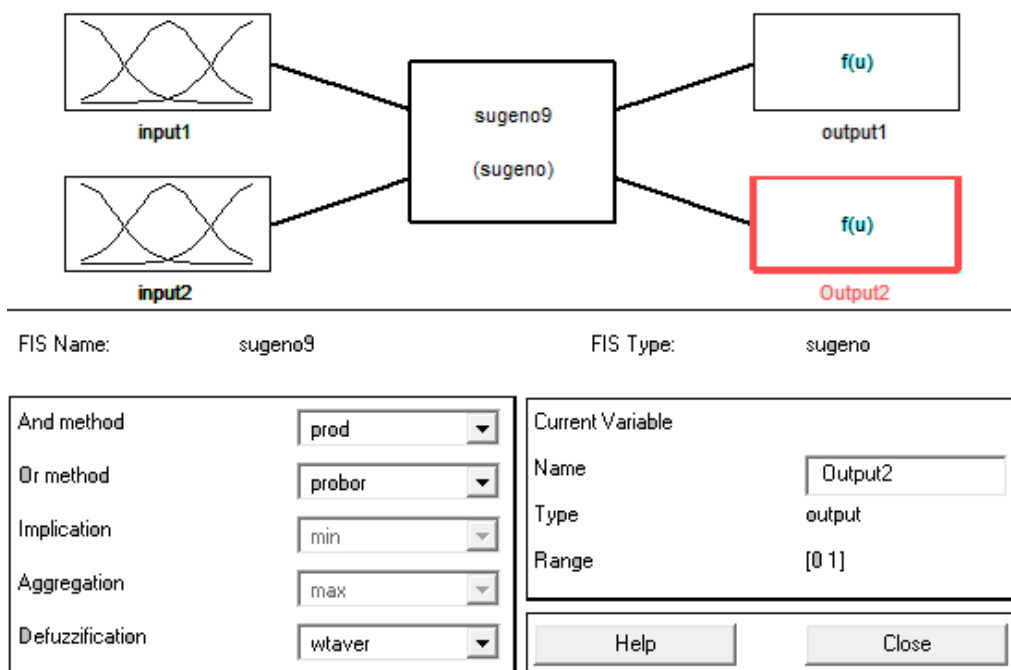


Рис. 3.8 – Вікно редактора нечіткого регулятора з двома входами і двома виходами сигналів

До теперішнього часу запропоновано кілька алгоритмів нечіткого висновку (алгоритм Мамдані, Цукамото, Ларсена, Сугено та ін.). У даній роботі представляється алгоритм Сугено 0-порядку, так як він являється найпростішим алгоритмом нечіткого виводу, також було розглянуто алгоритм Мамдані.

Формально алгоритм Сугено, запропонований Сугено та Такагі, може бути визначений таким чином.

У базі правил використовуються тільки правила нечітких продукцій в формі:

ПРАВИЛО <1>: ЯКЩО  $(x \in A_1 \mid y \in B_1)$ , ТО,  $z_1 = f(x_1, \dots, x_n)$ ,

ПРАВИЛО <2>: ЯКЩО  $(x \in A_2 \mid y \in B_2)$ , ТО,  $z_2 = f(x_1, \dots, x_n)$ ,

де  $x, y$  – вхідні змінні,  $A_i, B_i$  – нечіткі множини  $z_1=f(x_1, \dots, x_n)$  – довільна чітка функція.

Якщо у якості  $f$  використовується поліном  $f(x)=C_i$ , то говорять про алгоритм Сугено 0-порядку. Тоді правила будуть мати наступний вигляд:

ПРАВИЛО <1>: ЯКЩО  $(x \in A_1 \mid y \in B_1)$ , ТО,  $z_1 = C_1$ ,

ПРАВИЛО <2>: ЯКЩО  $(x \in A_2 \mid y \in B_2)$ , ТО,  $z_2 = C_2$ ,

(1)

де  $C_1, C_2$  – звичайні (чіткі) числа.

Розглядаючи структуру керування, можна визначити наступні п'ять правил функціонування НЛР у форматі «if – then», як показано вище у форматі (1):

П1: Якщо похибка регулювання  $e_1$  належить нечіткій множині «велика позитивна PE» і приріст похибки регулювання  $\Delta e_1$  належить множині «приблизно нульова Z», то керуюча дія  $u_1 = w_1$ , а приріст керуючої дії  $\Delta u_1 = w_2$ ;

П2: Якщо похибка регулювання  $e_1$  належить нечіткій множині «велика негативна NE» і приріст похибки регулювання  $\Delta e_1$  належить множині «приблизно нульова Z», то керуюча дія  $u_1 = -w_1$ , а приріст керуючої дії  $\Delta u_1 = -w_2$ ;

П3: Якщо похибка регулювання  $e_1$  належить нечіткій множині «приблизно нульова Z» і приріст похибки регулювання  $\Delta e_1$  належить множині «приблизно нульова Z», то керуюча дія  $u_1 = 0$ , а приріст керуючої дії  $\Delta u_1 = 0$ ;

П4: Якщо похибка регулювання  $e_1$  належить нечіткій множині «приблизно нульова Z» і приріст похибки регулювання  $\Delta e_1$  належить множині «велика позитивна PE», то керуюча дія  $u_1 = w_1$ , а приріст керуючої дії  $\Delta u_1 = w_2$ ;

П5: Якщо похибка регулювання  $e_1$  належить нечіткій множині «приблизно нульова Z» і приріст похибки регулювання  $\Delta e_1$  належить множині

«велика негативна NE», то керуюча дія  $u_1 = -w_1$ , а приріст керуючої дії  $\Delta u_1 = -w_2$ .

Відповідно вище представленим правилам складемо таблиці баз знань для НЛР температури вина.

Як видно з рисунку 4.2, на вхід даного нечіткого регулятора будуть подаватись дві змінні – помилка регулювання (input1) та приріст помилки регулювання (input2). Для змінної input1 задаємо три трикутні функції приналежності «trimf». Це проста функція приналежності і найбільш часто застосовується. Трикутна функція приналежності задається наступною аналітичною формулою:

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases}$$

[a, c] – діапазон зміни змінної, для справжнього випадку  $a = -35$ ,  $c = 35$ ;  $b$  - найбільш можливе значення змінної.

Таблиця 3.1 – Таблиця бази знань для нечіткого логічного регулятора температури вина

$e_1$ \ $\Delta e_1$	NE (Велика негативна $\Delta e_1$ )	Z (приблизно нульова $\Delta e_1$ )	PE (Велика позитивна $\Delta e_1$ )
NE	–	$u_1 = -200$ ; $\Delta u_1 = -0,5$ ;	–
Z	$u_1 = -200$ ; $\Delta u_1 = -0,5$ ;	$u_1 = 0$ ; $\Delta u_1 = 0$ ;	$u_1 = 200$ ; $\Delta u_1 = 0,5$ ;
PE	–	$u_1 = 200$ ; $\Delta u_1 = 0,5$ ;	–

Всі функції приналежності були визначені у відповідному редакторі на базі експертних даних. На рисунку 4.3 зображено вікно налаштування функцій приналежності  $mf1$ ,  $mf2$ ,  $mf3$  до нечітких множин похибки регулювання «велика негативна», «приблизно нульова», «велика позитивна» відповідно. Також були визначені функції приналежності  $mf1$ ,  $mf2$ ,  $mf3$  до нечітких множин приросту похибки регулювання «великий негативний», «приблизно нульовий» та «великий позитивний», які представлені на рисунку 4.4.

Згідно з таблицею 4.1 бази правил функціонування регулятора, були прописані відповідні правила у вікні редактора правил функціонування нечіткого регулятора середовища MATLAB\Simulink. Це вікно з відповідними правилами зображено на рисунку 4.5.

При дефазифікації вихідних змінних використовується модифікований варіант у формі методу центру тяжіння для одноточкових множин згідно з наступною формулою:

$$y_{out} = \frac{\sum_{i=1}^n (\alpha_i \cdot C_i)}{\sum_{i=1}^n (\alpha_i)}$$

де  $n$  – загальна кількість активних правил нечітких продукцій, у нашому випадку – 5 правил;

$\alpha_1 = \min(\mu_{A_1}(x_0), \mu_{B_1}(x_0))$ ,  $\alpha_2 = \min(\mu_{A_1}(x_0), \mu_{B_2}(x_0))$  – значення ступенів істинності всіх висновків правил нечітких продукцій з використанням методу  $\min$ -активізації.

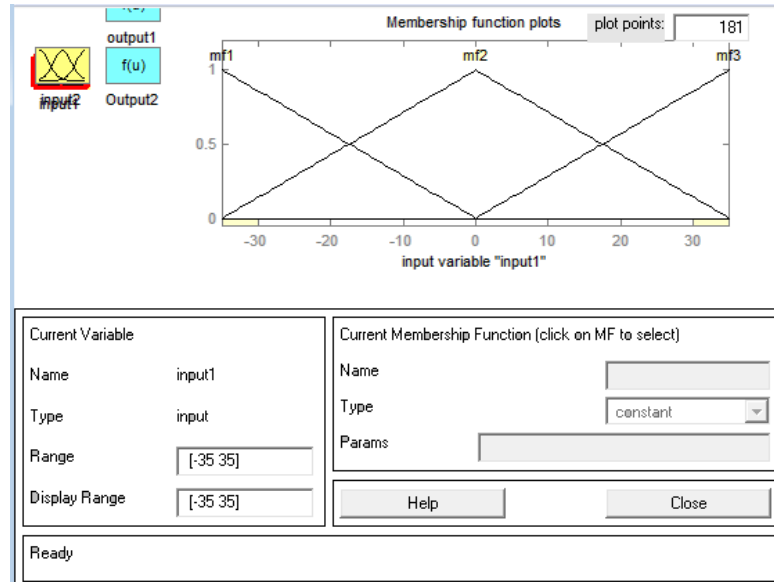


Рис. 3.9 – Вікно формування функцій приналежності до нечітких множин похибки регулювання PE, Z, NE редактора нечіткого регулятора середовища MATLAB\Simulink

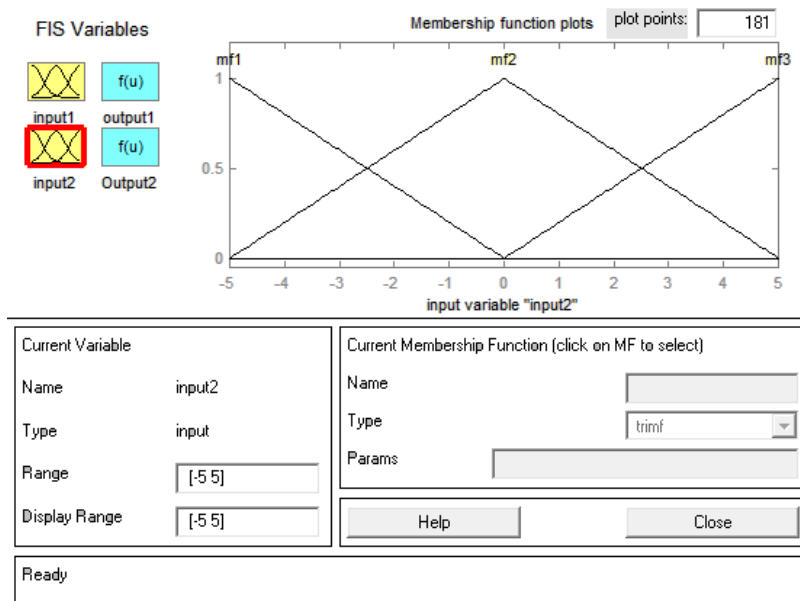


Рис. 3.10 – Вікно формування функцій приналежності до нечітких множин приросту похибки регулювання PE, Z, NE редактора нечіткого регулятора середовища MATLAB\Simulink

У даному випадку вихідні значення сигналів  $C_i$  згідно з алгоритмом Сугено нульового порядку були визначені у вікні реактора, представленого на рисунку 3.11. Виходами синтезованого нечіткого регулятора визначені зміни

«output1» та «output2». Вихід «output1» буде відпрацьовувати інтегральну П-складову регулятора, а «output2» – відповідно І-складову.

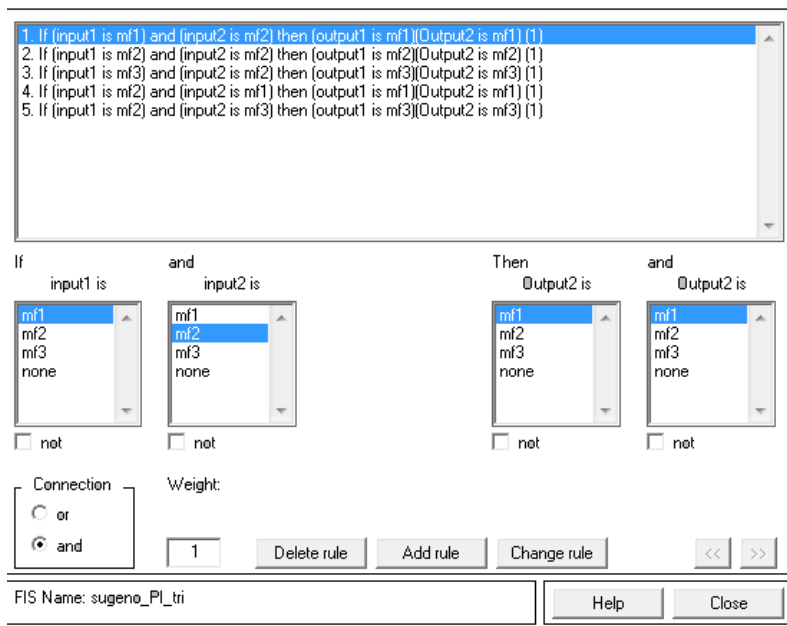


Рис. 3.11 – Вікно редактора правил функціонування нечіткого регулятора середовища MATLAB\Simulink

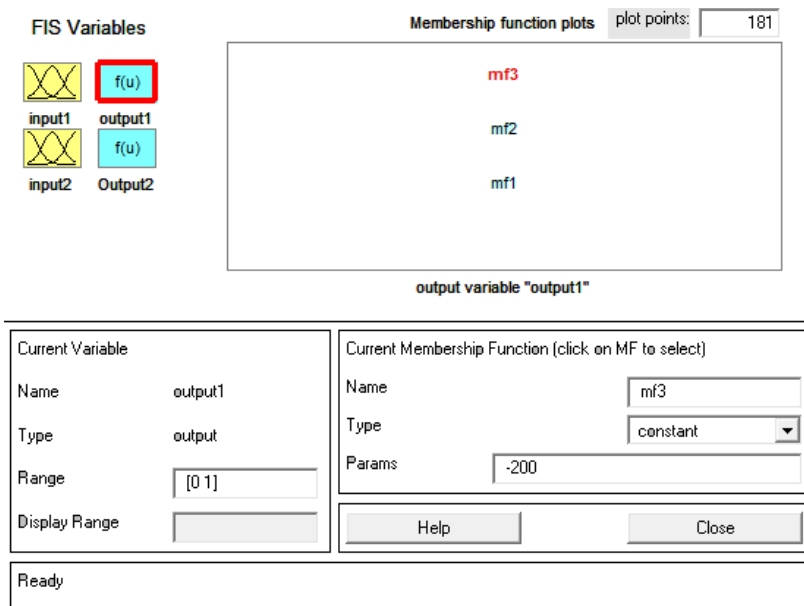


Рис. 3.12 – Вікно формування вихідних значень сигналів  $C_i$  згідно з алгоритмом Сугено нульового порядку

Як була сказано раніше, в роботі також розглядався варіант нечіткого регулятора з алгоритмом Мандані.

Алгоритм використовується переважно в задачах нечіткого моделювання, де дозволяє значно зменшити обсяги обчислень.

Формується в предметній області у вигляді нечітких предикатних правил виду:

П1: якщо  $x \in A_1$ , тоді  $z \in B_1$ ,

П2: якщо  $x \in A_2$ , тоді  $z \in B_2$ ,

.....

Пn: якщо  $x \in A_n$ , тоді  $z \in B_n$ ,

де  $x$  — вхідна змінна (ім'я для відомих значень даних),  $z$  — змінна виводу (ім'я для значення даних, яке буде обчислене).  $A_i$  та  $B_i$  — нечіткі множини, визначені на  $X$  та  $Z$  відповідно за допомогою функції приналежності та ( $z$ ).

У представлений ситуації даних вивід у формі алгоритму Мамдані математично можна представити наступним чином.

1. Введення нечіткості (fuzzification): для заданого (чіткого) значення аргументу  $x = x_0$  знаходяться степені істинності для передумов кожного правила  $\alpha_i = A_i(x_0)$ .

2. Нечіткий вивід: знаходиться рівні відсічення для передумов кожного з правил (з використання правила мінімуму):

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0)$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0)$$

де через  $\wedge$  позначено операцію логічного мінімуму.

Потім знаходяться усічені функції належності:

$$C'_1(z) = (\alpha_1 \wedge C_1(z))$$

$$C'_2(z) = (\alpha_2 \wedge C_2(z))$$

3. Композиція: з використанням операції max (позначеної як  $\vee$ ) виконується об'єднання знайдених усічених функцій, що приводить до отримання підсумкової нечіткої підмножини для змінної виходу з функцією належності.

$$M(z) = C(z) = C'_1(z) \vee C'_2(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z))$$

Приведення до чіткості (дефазифікації). Дана процедура використовується, коли необхідно перетворити нечітку вихідну множину в чітке число. Найчастіше для моделі Мамдані використовують дефазифікація центроїдним методом, коли чітке значення вихідної змінної визначається як центр ваги для кривої  $\mu_{\Sigma}(z)$

Центр ваги (англ. Centre of Gravity, CoG) або центроїд площі розраховується за формулою:

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

де  $y$  – результат дефазифікації;  $x$  – змінна, що відповідає вихідній лінгвістичній змінній  $w_i$ ;  $\mu(x)$  – функція приналежності нечіткої множини, що відповідає вихідній змінній  $w$  після етапу акумуляції;  $min, max$  – ліва та права точки інтервалу носія нечіткої множини вихідної змінної  $w$ .

У даному випадку вихідні значення сигналів  $C_i$  згідно з алгоритмом Мамдані були визначені у вікні реактора, представленого на рисунку 3.13. Виходами синтезованого нечіткого регулятора визначені зміни «output1» та «output2». Вихід «output1» буде відпрацьовувати інтегральну П-складову регулятора, а «output2» – відповідно І-складову.

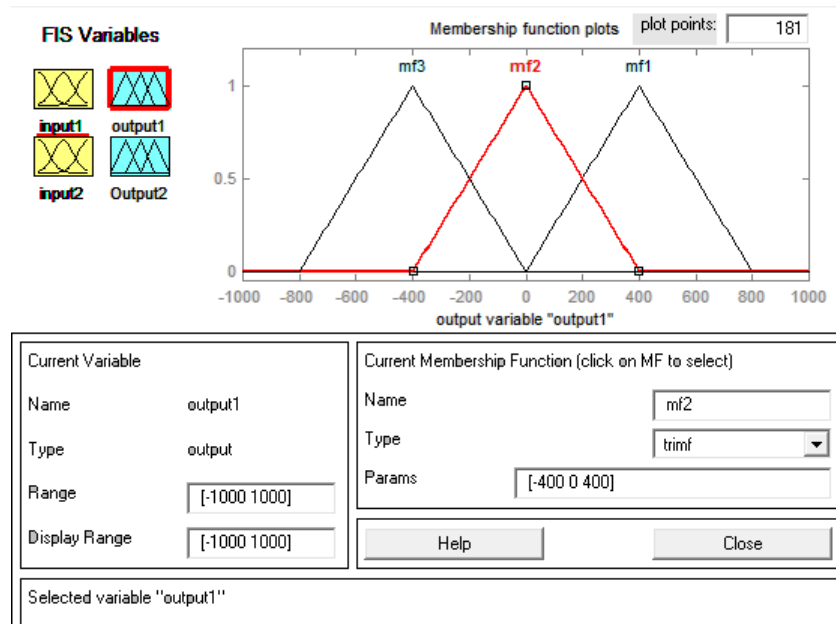


Рис. 3.13 – Вікно формування вихідних значень сигналів  $C_i$  згідно з алгоритмом Мамдані

Для запуску синтезованого алгоритму було перетягнуто на робоче поле створення моделі SIMULINK блок «Fuzzy Logic Controller» зі стандартної бібліотеки. У відкритому вікні «Fuzzy Logic Controller» було записано в запропонованому полі ім'я fis-файлу, а в головному вікні налаштування fis-файлу було вибрано File / Save to Workspace для завантаження у робочій простір середовища MATLAB нечіткого алгоритму управління.

У кінцевому випадку була визначена схема моделювання САР з нечітким регулятором, яка представлена на рисунку 4.8. З рисунку 4.8 видно, що управляюча дія нечіткого регулятора формується з декілька складових за формулою:

$$u(t) = u_1(t) + (u_2(t - \tau_z) + \Delta u_2),$$

де  $u_1(t)$  – управляюча дія П-складової регулятора;

$\Delta u_2 = u_2(t) - u_2(t - \tau_z)$  – приріст управляючої дії від нечіткого регулятора;

$u_2(t - \tau_z)$  – управляюча дія з затримкою за часом  $\tau_z$ .

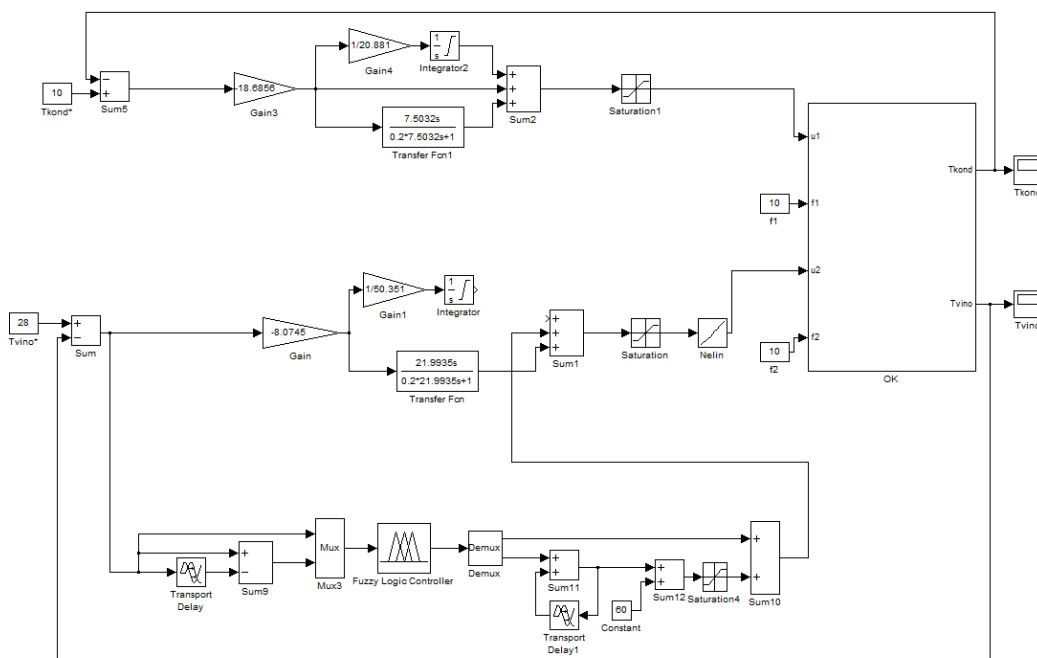


Рис. 3.14 – Схеми моделювання САР з нечітким логічним регулятором

Слід зазначити, що при синтезі нечіткого логічного регулятора «Fuzzy Logic Controller» всі функції приналежності к нечітким множинам та логічні правила функціонування визначені на базі експертних даних. А потім скореговані при моделюванні методом «проб і помилок». Також логічні правила функціонування були визначені з урахуванням правил функціонування нечітких регуляторів, представлених у відомих наукових роботах, одних із перших з Fuzzy керуванням.

### 3.5. Порівняльний аналіз функціонування САР з традиційним ПІД-регулятором і з САР з нечітким регулятором

У результаті моделювання САР з нечітким регулятором були отримані графіки перехідних процесів при детермінованому і випадковому впливі, що збурює. Для порівняльного аналізу також були представлені відповідні графіки перехідних процесів у САР з традиційним ПІД-регулятором.

У якості нечітких регуляторів було обрано регулятор з алгоритмом Сугено 0-порядку з функцією приналежності до нечітких безлічей похибки регулювання «trimf», а також регулятор з алгоритмом Мамдані з функцією приналежності до нечітких безлічей похибки регулювання «trimf».

Таким чином, на рисунку 5.1 представлені графіки перехідних процесів при детермінованому впливі, що збурює. А на рисунку 5.2 представлені графіки сталих процесів при випадковому впливі, що збурює.

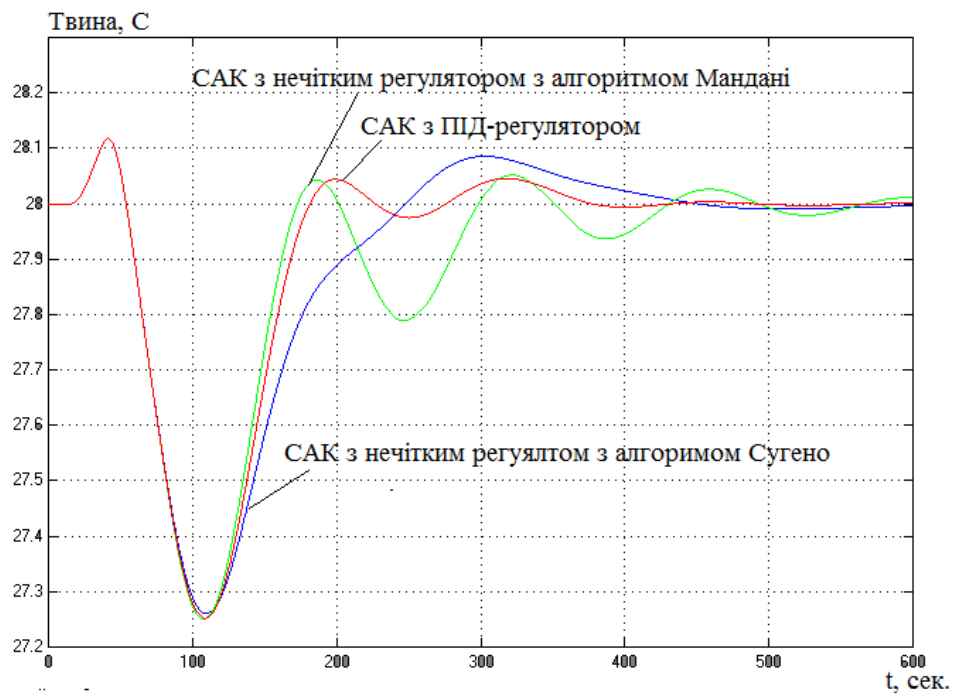


Рис. 3.15 – Графіки перехідних процесів при детермінованому впливі, що збурює, за каналом регулювання « $U_{\text{шов}} - T_{\text{вина}}$ », що отримані в результаті моделювання різних САР

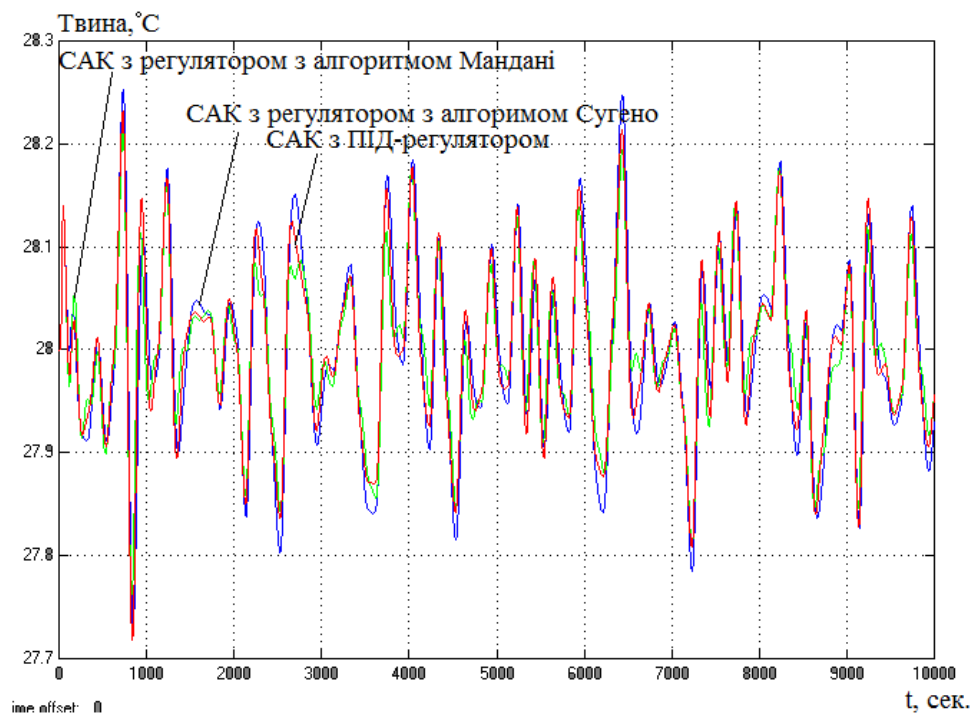


Рис. 3.16 – Графіки сталих процесів при випадковому впливі, що збурює, за каналом регулювання « $U_{\text{н}} - \theta$ », що отримані в результаті моделювання різних САР

З рисунків вище можна побачити, що при детермінованому збуренні нечіткий регулятор з алгоритмом Мамдані працює значно гірше, ніж ПД-регулятор та нечіткий регулятор з алгоритмом Сугено 0-порядку. Перехідні процеси в САР з нечітким регулятором з алгоритмом Сугено 0-порядку з функцією приналежності «trimf» трохи гірші за перехідні процеси в САР з ПД-регулятором.

Також можна побачити, що в сталих режимах при випадковому збуренні усі САР працюють задовільно.

Отже надалі в якості нечіткого регулятора будемо використовувати нечіткий регулятор з алгоритмом Сугено 0-порядку з функцією приналежності «trimf».

### 3.6. Обґрунтування актуальності застосування нейромережових алгоритмів керування

Потенційними областями застосування штучних нейронних мереж є ті, де людський інтелект малоефективний, а традиційні обчислення трудомісткі або фізично неадекватні. Актуальність застосування нейронних мереж багаторазово зростає, коли з'являється необхідність розв'язку погано формалізованих задач. Основні області застосування нейронних мереж: автоматизація процесу класифікації, автоматизація прогнозування, автоматизація процесу розпізнавання, автоматизація процесу прийняття розв'язків; керування, кодування і декодування інформації; апроксимація залежностей і ін.

Штучні нейронні мережі (ШНМ) у цей час знаходять широке застосування в самих різних предметних областях. Одним з важливих напрямків використання штучних нейронних мереж є нейроуправління в системах автоматичного керування різних типів. Нейроуправління — це окремий випадок інтелектуального керування, при якому у якості інструмента для розв'язку завдань керування застосовуються штучні нейронні мережі.

Штучна нейронна мережа як нейрорегулятор здійснює нелінійне перетворення вхідного сигналу і формування управляючого впливу. При цьому регулятор може мати велику кількість оптимізованих параметрів (коефіцієнтів міжнейронних зв'язків), що дає можливість оптимізувати його практично під будь-який об'єкт з нелінійною статичною характеристикою. У даній роботі нейронна мережа – це математична модель з параметрами, що оптимізуються, яка буде виконувати функції розробленого раніше нечіткого логічного регулятора.

### 3.7. Розробка моделі САР з нейрорегулятором

Нейрорегулятор у нашому випадку – це штучна нейронна мережа, що характеризується нелінійними характеристиками. Завдяки цьому застосування нейрорегулятора на базі нейронної мережі дозволить досягти більш високих результатів в керуванні. Нейронна мережа – це послідовність нейронів (нейронелементів), з'єднаних між собою синапсами. Структура нейронної мережі прийшла в світ програмування прямо з біології. Найбільш поширена структура – багат шарова, застосовується в якості регуляторів нейромережевих алгоритмів керування.

Математична модель, що застосовується у якості нейронелементу у нейромереживому алгоритмі керування, аналогічна відомим найбільш розповсюдженим моделям, які схожі з формальною моделлю нейрона Маккаллоха–Питса. У даній моделі (рисунок 7.1) сигнали, що поступили на входи нейроелементу, множаться на свої ваги. Сигнал першого входу  $x_1$  множиться на відповідну вагу  $w_1$ . У підсумку отримуємо  $x_1w_1$ . І так до n-ого входу. У підсумку на останньому вході отримуємо  $x_nw_n$ . Потім всі множення передаються в суматор. Уже виходячи з його назви можна зрозуміти, що він робить. Він просто підсумовує всі вхідні сигнали, помножені на відповідні ваги:

$$x_1w_1 + x_2w_2 + \dots + x_nw_n = \sum_{i=1}^n x_iw_i$$

Просто так подавати виважену суму на вихід досить безглуздо. Нейроелемент повинен якимось обробити її і сформувати адекватний вихідний сигнал. Саме для цих цілей і використовують функцію активації. Вона перетворює зважену суму в якесь число, яке і є виходом нейрона.

У зв'язку з тим, що нечіткій регулятора визначає приріст управляючої дії як позитивне значення, так і негативне значення, то при синтезі нейрорегулятора необхідно вибрати функції активації нейронів, які приймають значення від -1 до 1. В нашому випадку буде використано гіперболічну тангенціальну функцію (tansig). Така функція дозволяє отримати на виході значення різних знаків (наприклад, від -1 до 1), що може бути необхідним в системі керування. Графічне представлення функції відображено на рисунку 7.2.

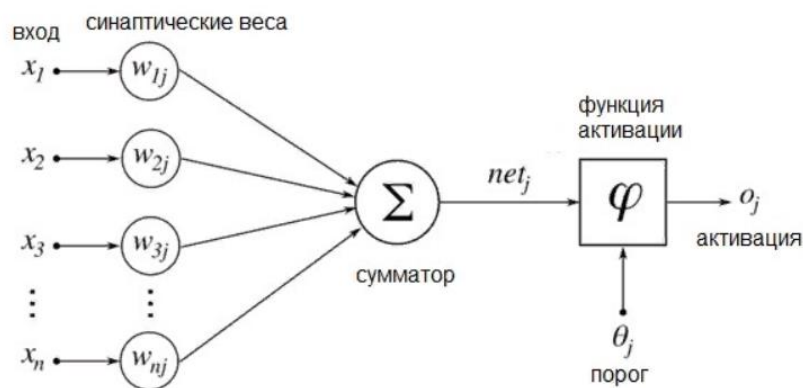


Рис. 3.17 – Структурна схема моделі нейроелемента нейронного регулятора

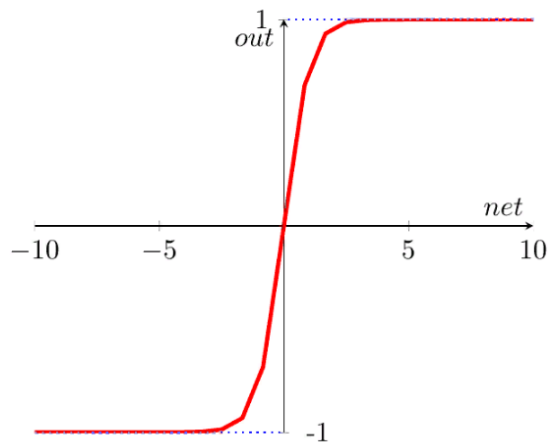


Рис. 3.18 – Графічне представлення функції активації штучного нейрона, що застосовується у нейрорегуляторі

Аналітично функція записується у наступному вигляді:

$$f(net) = \frac{2}{1 + \exp(-\alpha \cdot net)} - 1, \text{ де } net \text{ – вхідний аргумент.}$$

Відомо, що одношарова нейронна мережа має обмежені можливості. Багатошарова чотирьох-шарова вже надлишкова для рішення різних задач. В даному випадку нейрорегулятор буде представлений нейронною мережею з двома шарами, як показано на рисунку 7.3. На рисунку 7.3 також зображені похибки на кожному нейроні для коректування ваг за методом зворотного поширення. Якщо така структура буде недостатня, то необхідно буде збільшити кількість нейронів у вхідному шарі мережі та збільшити кількість шарів до трьох.

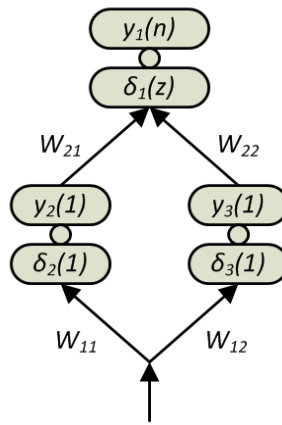


Рис. 3.19 – Структурна схема нейрорегулятора інтегральної або пропорційної складової

Для визначення ваг між нейронними з'єднаннями за методом зворотного поширення було використано програмне середовище MATLAB, що дозволяє згенерувати задану нейрону мережу на базі вхідних і вихідних значень сигналів. Таким чином, для тренування нейронної мережі необхідно указати вхідні і вихідні значення сигналів нейрорегулятора, а точніше, значення похибок регулювання та приріст керуючого впливу.

Штучна нейрона мережа, що виконує роль регулятора, синтезується на базі характеристик нечіткого логічного регулятора. Ці характеристики представляються у табличній формі у вигляді вхідних і вихідних значень нечіткого регулятора (таблиця 3.2). Дані, представлені у таблиці 3.2, визначені на базі моделювання нечіткого регулятора.

Таблиця 3.2 – Вхідні та вихідні значення сигналів регулятора

№	$\Delta T$	$u$	$\Delta u$
1	2	-11,43	-0,02857
2	1,6	-9,143	-0,0278
3	1,2	-6,857	-0,01714
4	0,8	-4,571	-0,0114
5	0,4	-2,286	-0,0057
6	0	0	0
7	-0,4	2,286	0,0057
8	-0,8	4,571	0,0114
9	-1,2	6,857	0,01714
10	-1,6	9,143	0,0278
11	-2	11,43	0,02857

Далі створюємо М-файл в середовищі Matlab і в цьому файлі прописуємо команди для створення і навчання нашого нейрорегулятора. Вхідні і вихідні значення нейрорегулятора були отримані на основі характеристик нечіткого логічного регулятора, отриманих на основі схеми моделювання, представлені на рисунку 3.21.

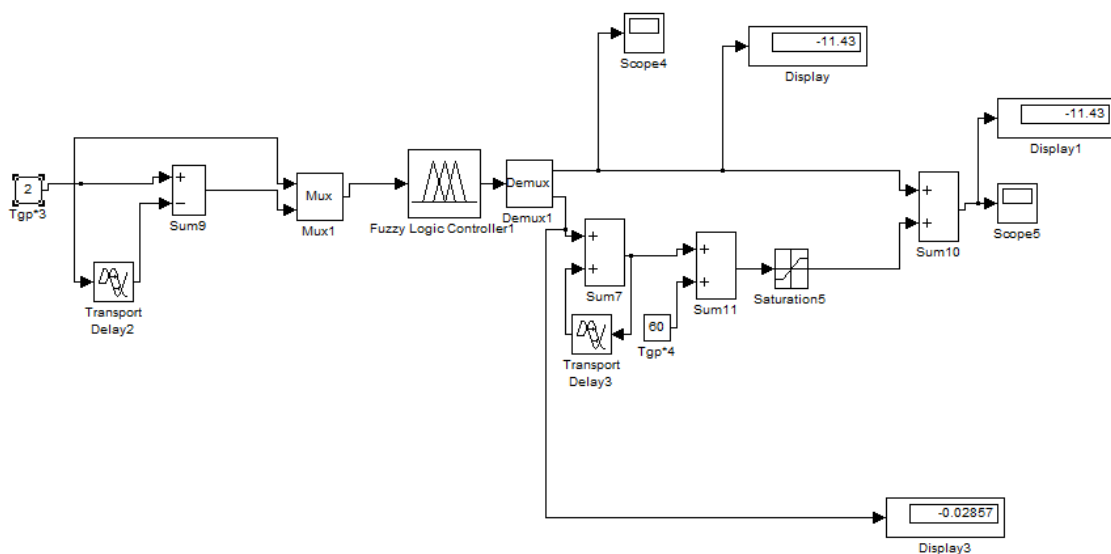


Рис. 3.20 – Схема для отримання даних для навчання нейронної мережі

В М-файлі було записано виконавчий код, що представлений на рисунках 3.22 і 3.23 відповідно для пропорційної та інтегральної складової регулятора. Представлений виконавчий код також можливо записати у командній строчці Matlab – command window, в результаті чого згенерується відповідна нейронна мережа.

В даному виконавчому коді p – вхідні значення нейромережі (похибка регулювання), t – відповідне значення виходу нейромережі. На третій строчці задаємо діапазон, кількість шарів нейромережі та функції активації. Також задаємо кількість кроків тренування, команду тренування нейромережі та генерації.

```

» p=[-2 -1.6 -1.2 -0.8 -0.4 0 0.4 0.8 1.2 1.6 2];
» t=[1.143 0.9143 0.6857 0.4571 0.2286 0 -0.2286 -0.4571 -0.6857 -0.9143 -1.143];
» net=newff([-2 2], [2 1], {'tansig' 'tansig'});
» net.trainParam.epochs=100;
» net=train(net, p, t);
TRAINLM, Epoch 0/100, MSE 1.25302/0, Gradient 5.51364/1e-010
TRAINLM, Epoch 25/100, MSE 0.00455284/0, Gradient 0.00131383/1e-010
TRAINLM, Epoch 50/100, MSE 0.00446654/0, Gradient 0.000389193/1e-010
TRAINLM, Epoch 75/100, MSE 0.00444209/0, Gradient 0.000465183/1e-010
TRAINLM, Epoch 100/100, MSE 0.00443357/0, Gradient 0.00185139/1e-010
TRAINLM, Maximum epoch reached, performance goal was not met.

» a=sim(net,p)

a =

Columns 1 through 7

    0.9926    0.9220    0.7164    0.4450    0.2038    -0.0000    -0.2038

Columns 8 through 11

   -0.4450   -0.7164   -0.9221   -0.9926

» gensim(net)

```

Рис. 3.21 – Виконавчий код для генерації нейромережі, що відтворює пропорційну складову нейрорегулятора

```

» p=[-2 -1.6 -1.2 -0.8 -0.4 0 0.4 0.8 1.2 1.6 2];
» t=[0.02857 0.0228 0.01714 0.0114 0.0057 0 -0.0057 -0.0114 -0.01714 -0.0228 -0.02857];
» net=newff([-2 2], [2 1], {'tansig' 'tansig'});
» net.trainParam.epochs=100;
» net=train(net, p, t);
TRAINLM, Epoch 0/100, MSE 0.821887/0, Gradient 2.78806/1e-010
TRAINLM, Epoch 25/100, MSE 9.64408e-008/0, Gradient 0.00109749/1e-010
TRAINLM, Epoch 50/100, MSE 2.98157e-008/0, Gradient 0.000214184/1e-010
TRAINLM, Epoch 75/100, MSE 1.71491e-008/0, Gradient 9.02932e-005/1e-010
TRAINLM, Epoch 100/100, MSE 1.19508e-008/0, Gradient 4.98513e-005/1e-010
TRAINLM, Maximum epoch reached, performance goal was not met.

» a=sim(net,p)

a =

Columns 1 through 7

    0.0285    0.0230    0.0171    0.0113    0.0056   -0.0000   -0.0056

Columns 8 through 11

   -0.0113   -0.0170   -0.0228   -0.0287

» gensim(net)

```

Рис. 3.22 – Виконавчий код для генерації нейромережі, що відтворює інтегральну складову нейрорегулятора

Характеристики точності навчання показані на рисунку 7.7, з яких можливо побачити, що десь на 70-м кроці навчання коректування ваг між нейронними з'єднаннями було закінчено.

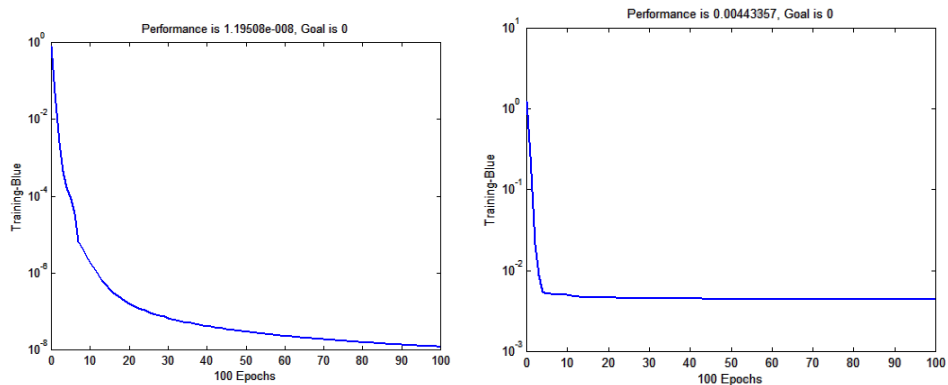


Рис. 3.23 – Характеристика точності навчання пропорційної (справа) та інтегральної (зліва) складової нейрорегулятора

Водночас з генерацією НМ в головному вікні Matlab відображаються фактичні значення вихідного сигналу і-того вихідного нейрону мережі. На рисунках 3.25 та 3.26 зображено бажані та фактичні значення вихідного сигналу і-того вихідного нейрону НМ, що відтворює II – складову та I – складову нейрорегулятора температури вина.

**Фактичні значення:**

0.9926 0.9220 0.7164 0.4450 0.2038 0 -0.2038 -0.4450 -0.7164 -0.9221 -0.9926

**Бажані значення:**

1.143 0.9143 0.6857 0.4571 0.2286 0 -0.2286 -0.4571 -0.6857 -0.9143 -1.143

Рис. 3.24 – Бажані та фактичні значення вихідного сигналу  $i$ -того вихідного нейрону НМ, що відтворює  $\Pi$  – складову нейрорегулятора

**Фактичні значення:**

0.0285 0.0230 0.0171 0.0113 0.0056 0 -0.0056 -0.0113 -0.0170 -0.0228 -0.0287

**Бажані значення:**

0.02857 0.0228 0.01714 0.0114 0.0057 0 -0.0057 -0.0114 -0.01714 -0.0228 -0.02857

Рис. 3.25 – Бажані та фактичні значення вихідного сигналу  $i$ -того вихідного нейрону НМ, що відтворює  $I$  – складову нейрорегулятора

Виходячи з цих даних, представлених на рисунках 7.8 та 7.9, розрахуємо похибки для вихідного шару та похибки для  $i$ -тих нейронів прихованого шару НМ за методом зворотного поширення для оцінки якості тренування нейронної мережі.

Згідно навчанню штучної нейронної мережі за методом зворотного розповсюдження похибки, похибка для вихідного шару нейронної мережі визначається за наступною формулою (1):

$$\delta_i(n) = (y_i(n) - d_i) \cdot y_i(n) \cdot (1 - y_i(n)) \quad (1)$$

де  $d_i$  – бажане значення вихідного сигналу  $i$ -того вихідного нейрону мережі.

$y_i$  – фактичне значення вихідного сигналу  $i$ -того вихідного нейрону мережі.

Похибка для  $i$ -того нейрону прихованого шару розраховується за наступною формулою (2):

$$\delta_i(n) = y_i(n) \cdot (y_i(n) - d_i) \cdot \sum w_{in+1} \cdot \delta_i(n+1) \quad (2)$$

де  $w_{in+1}$  – вага між-нейронного з'єднання n+1 шару;  $\delta_1(n+1)$  – похибка і-того нейрону n+1 шару.

Таким чином, розрахунок похибок для II – складової нейрорегулятора наступний:

Розрахунок помилки вихідного шару згідно з рисунку 7.3:

$$\begin{aligned}\delta_1(z) &= (y_1(n) - d_1) * y_1(n)(1 - y_1(n)) = \\ &= (-0,2038 - (-0,2286)) * (-0,2038) * (1 - (-0,2038)) = \\ &= -0,006\end{aligned}$$

Розрахунок помилки для і-тих нейронів прихованого шару:

$$\begin{aligned}\delta_2(1) &= y_2(1) * (1 - y_2(1)) * W_{21} * \delta_1(z) = \\ &= 0,9967 * (1 - 0,9967) * 92,32 * (-0,006) = \\ &= -0,001\end{aligned}$$

$$\begin{aligned}\delta_3(1) &= y_3(1) * (1 - y_3(1)) * W_{22} * \delta_1(z) = \\ &= 0,9989 * (1 - 0,9989) * (-92,31) * (-0,006) = \\ &= 0,0006\end{aligned}$$

Розрахунок помилок для I – складової нейрорегулятора:

Розрахунок помилки вихідного шару:

$$\begin{aligned}\delta_1(z) &= (y_1(n) - d_1) * y_1(n)(1 - y_1(n)) = \\ &= (-0,0056 - (-0,0057)) * (-0,0056) * (1 - (-0,0056)) = \\ &= -5,63 * 10^{-7}\end{aligned}$$

Розрахунок помилки для і-тих нейронів прихованого шару:

$$\begin{aligned}\delta_2(1) &= y_2(1) * (1 - y_2(1)) * W_{21} * \delta_1(z) = \\ &= -0,8235 * (1 - (-0,8235)) * (-1,45) * (-5,63 * 10^{-7}) = -12,25 * 10^{-7}\end{aligned}$$

$$\begin{aligned}\delta_3(1) &= y_3(1) * (1 - y_3(1)) * W_{22} * \delta_1(z) = \\ &= -0,9537 * (1 - (-0,9537)) * 0,0017 * (-5,63 * 10^{-7}) = 0,017 * 10^{-7}\end{aligned}$$

Результати розрахунків показали, що помилки практично нульові, отже ними можна знехтувати, а розрахунок значень наведеної погрішності не проводити.

Сформовані нейронні мережі були визначені у схемі моделювання замість нечіткого регулятора, як показано на рисунку 3.27. У такий спосіб була синтезована модель САУ з нейрорегулятором засобами середовища MATLAB\Simulink, структурна схема якої представлена на рисунку 3.27.

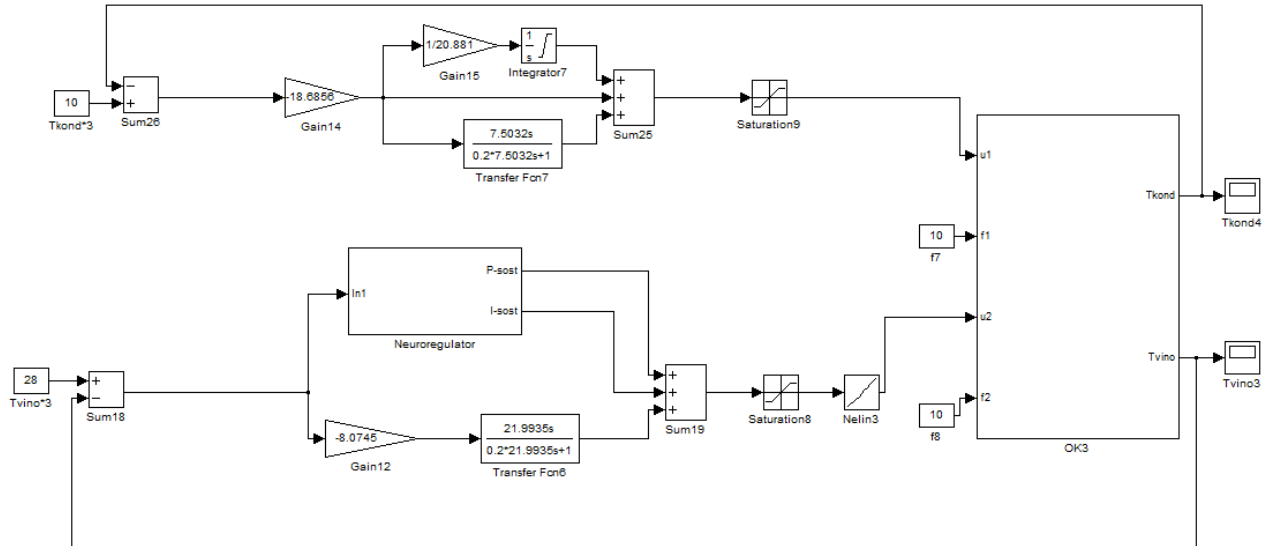


Рис. 3.26 – Структурна схема моделі САУ з нейрорегулятором, яка представлена засобами середовища MATLAB\Simulink .

На основі даних параметрів сформованої нейронної мережі була визначена схема нейрорегулятора у середовищі MATLAB\Simulink. Схема нейрорегулятора, що представлена на рисунку 3.28, є розгорнутою схемою, що представлена на попередньому рисунку 3.27.

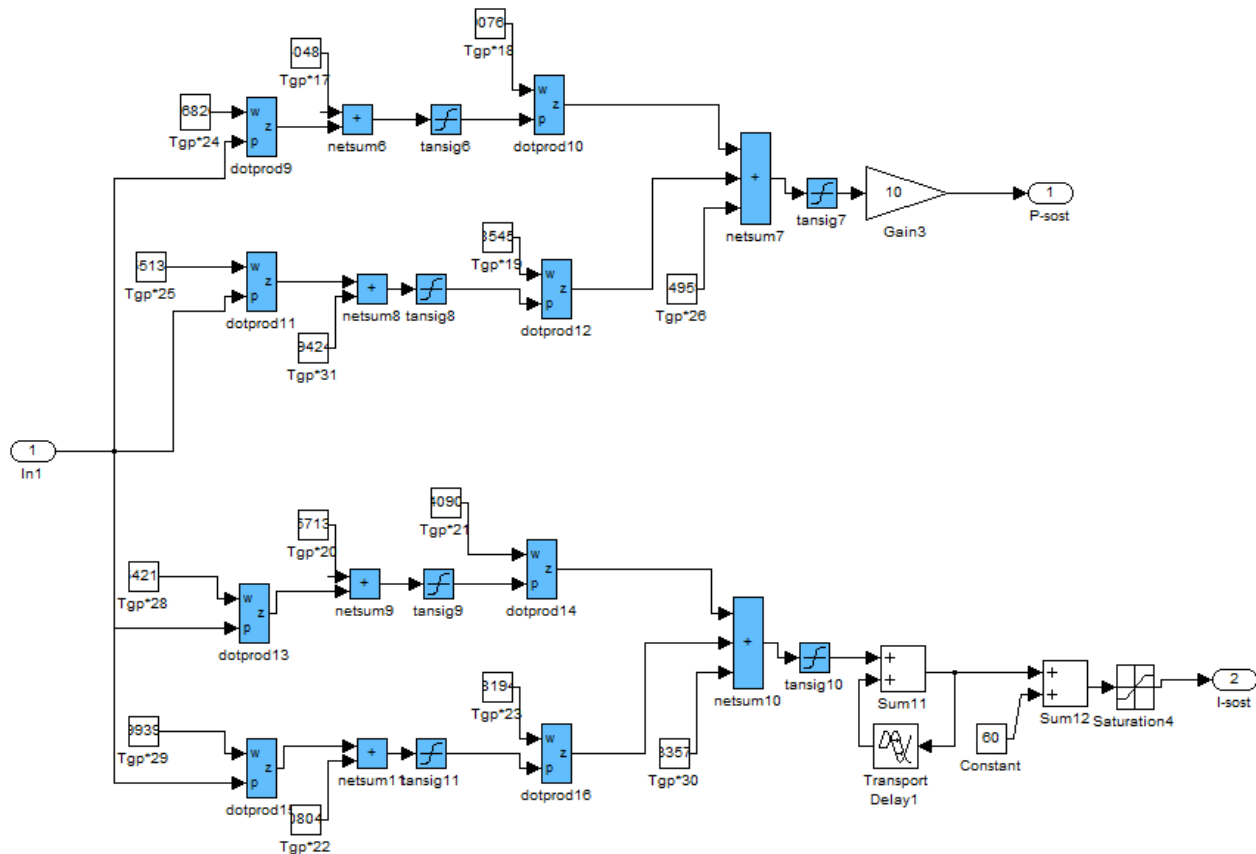


Рис. 3.27 – Схема моделювання нейронного регулятора

У результаті моделювання САР з нейрорегулятором були отримані графіки перехідних процесів при тих же умовах, які були при моделюванні САР з нечітким регулятором та традиційним ПД-регулятором. Ці графіки представлені на рисунках 7.12. З графіків перехідних процесів можливо побачити, що за якістю регулювання САР з нейрорегулятором працює нічим не гірше, ніж САР з нечітким регулятором та ПД-регулятором. Але за допомогою оптимізації можливо домогтись кращого показника.

Як видно з рис. 7.11 регулятор, що синтезований на базі штучних нейронних мереж, має 8 ваг міжнейронних з'єднань, а також є 6 зміщень. Методом «проб і помилок» було встановлено, що доцільно оптимізувати тільки 4 ваги вхідного слою, так як зміна значень інших коефіцієнтів призводить до помилки статизму в системі.

Можливо зробити висновок, що виходячи з якості регулювання різних САР необхідно нейрорегулятор оптимізувати при тих же умовах, як і

попередній традиційний ПІД-регулятор для визначення остаточного висновку про доцільність застосування різних САР.

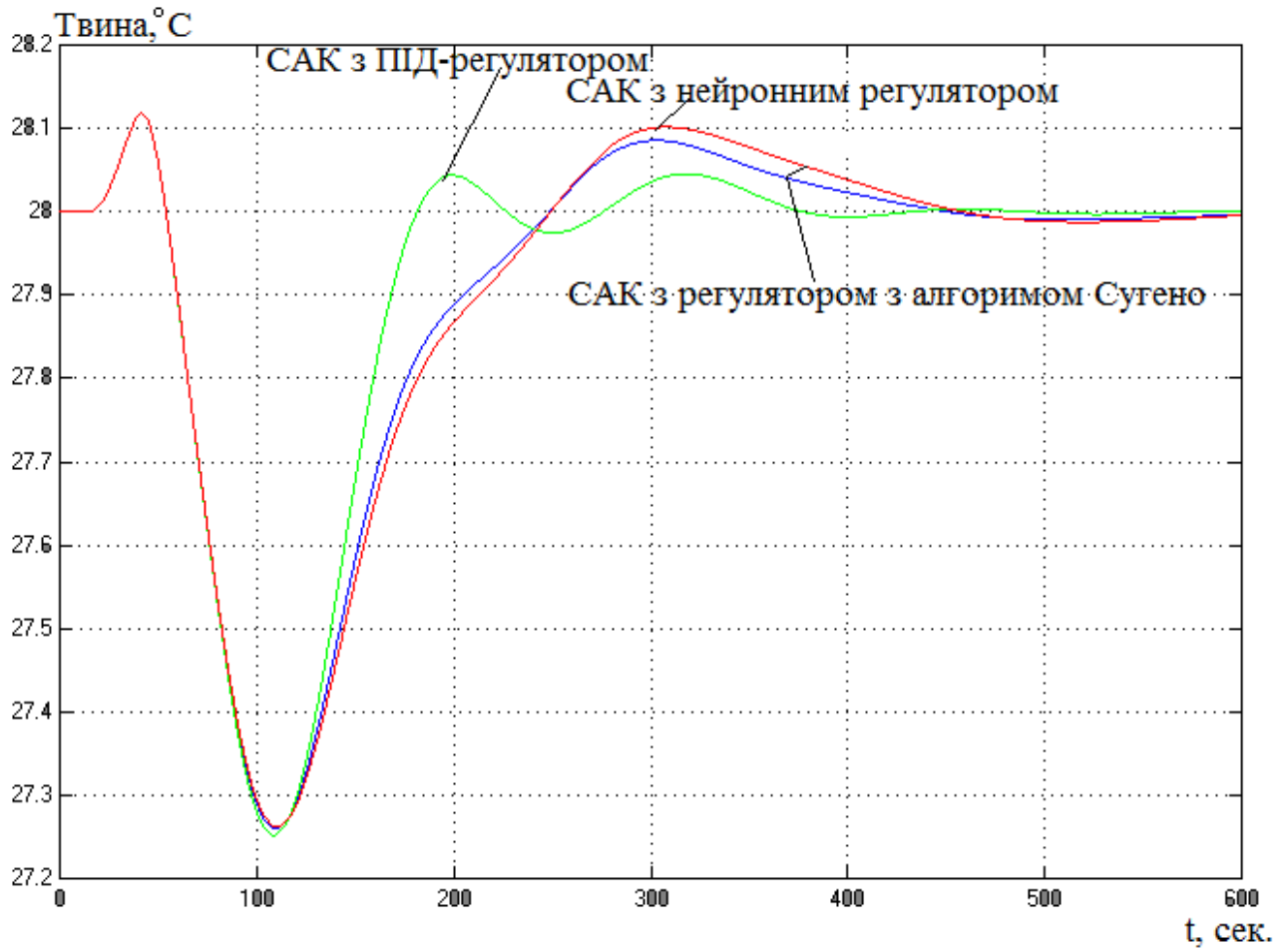


Рис. 3.28 – Графіки переходних процесів за каналом регулювання « $U_{\text{шов}} - T_{\text{вина}}$ », що отримані в результаті моделювання різних САР

### 3.8. Параметрична оптимізація САР з нейрорегулятором

Для покращення якості регулювання у роботі була проведена параметрична оптимізація САР з нейронним регулятором. Оптимізувались чотири параметри настроювання нейрорегуляторів – це вагові коефіцієнти нейронних з'єднань першого-вхідного шару штучної нейронної мережі. Оптимізація нейронного регулятора проводилась при таких же умовах, які були при оптимізації САР з традиційним ПІД-регулятором. Таким чином, інтегральний показник якості функціонування при оптимізації був обраний відповідно таким же, як при оптимізації САР з традиційним ПІД-регулятором.

У програмному середовищі MATLAB\Simulink 5.2 була визначена модель САР з нейрорегулятором і параметричним оптимізатором, структурна схема якої відображено на рисунку 3.30.

У результаті параметричної оптимізації були отримані графіки перехідних процесів до і після оптимізації, а також оптимальні визначені ваги міжнейронних з'єднань першого шару нейронної мережі. Графіки відповідних перехідних процесів та значення вагових коефіцієнтів представлені у відповідному вікні результатів оптимізації, представленою рисунку 3.31. На рисунку 3.31 представлений результат оптимізації, який був отриманий при декількох попередніх процесах оптимізації при визначеному інтегральному критерії якості роботи системи.

Після оптимізації доцільно перевірити систему на грубість з результатів аналізу перевірки системи на грубість (рис. 8.3) можна побачити, що система груба, а значить працездатна.

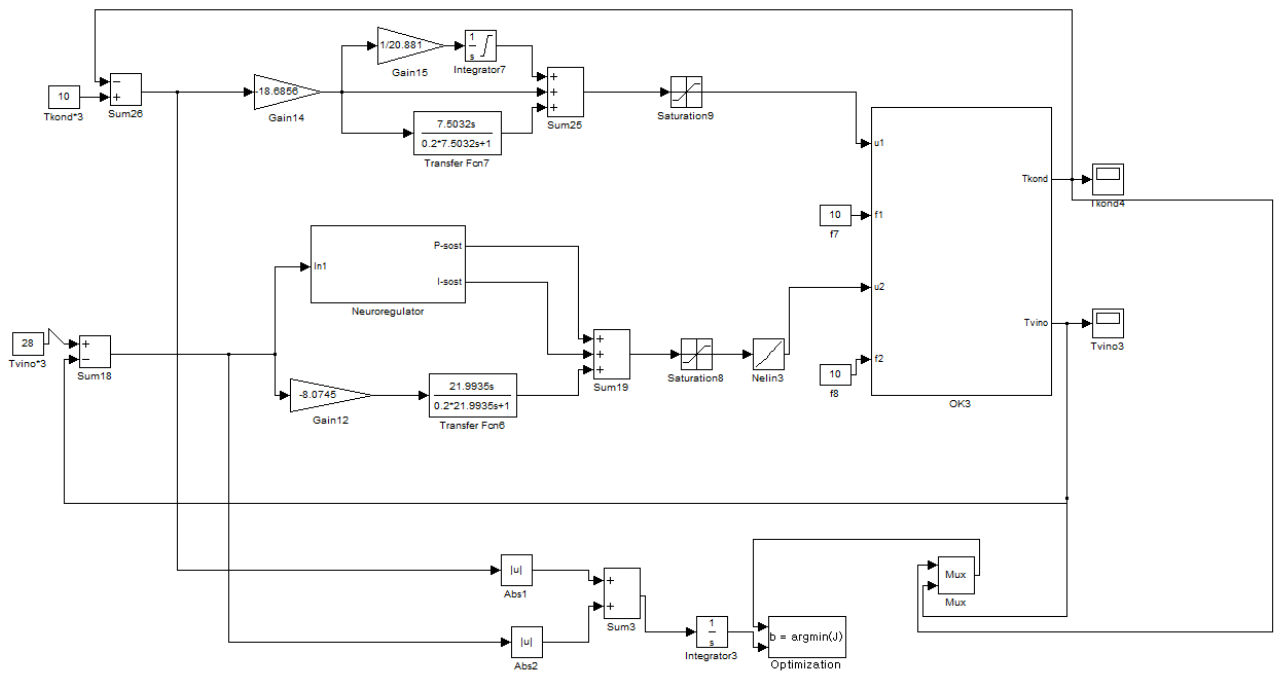
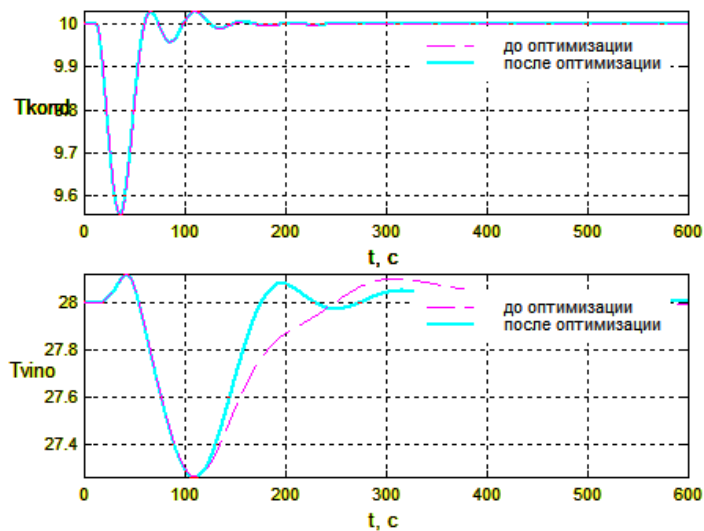


Рис. 3.29 – Структурна схема моделі САР з нейронним регулятором і параметричним оптимізатором, що реалізована засобами MATLAB\Simulink для визначення оптимальних параметрів системи керування за умови нелінійності статичної характеристики каналу регулювання « $U_{шов} - T_{вина}$ »



Имя переменной	Оптимальное значение	Нижний предел	Начальное приближение	Верхний предел	Значение критерия оптимальности:
K1	-1.0449	-50	-0.6951	50	до - 94.7994
K2	0.80976	-50	0.6947	50	после - 74.5064
K3	0.035204	-50	0.0295	50	оптимизации.
K4	-0.68709	-50	-1.1067	50	Количество шагов процедуры оптимизации:
					максимальное - 1000
					фактическое - 79

Рис. 3.30 – Вікно результатів оптимізації параметрів регулятора, представленого штучною нейронною мережею

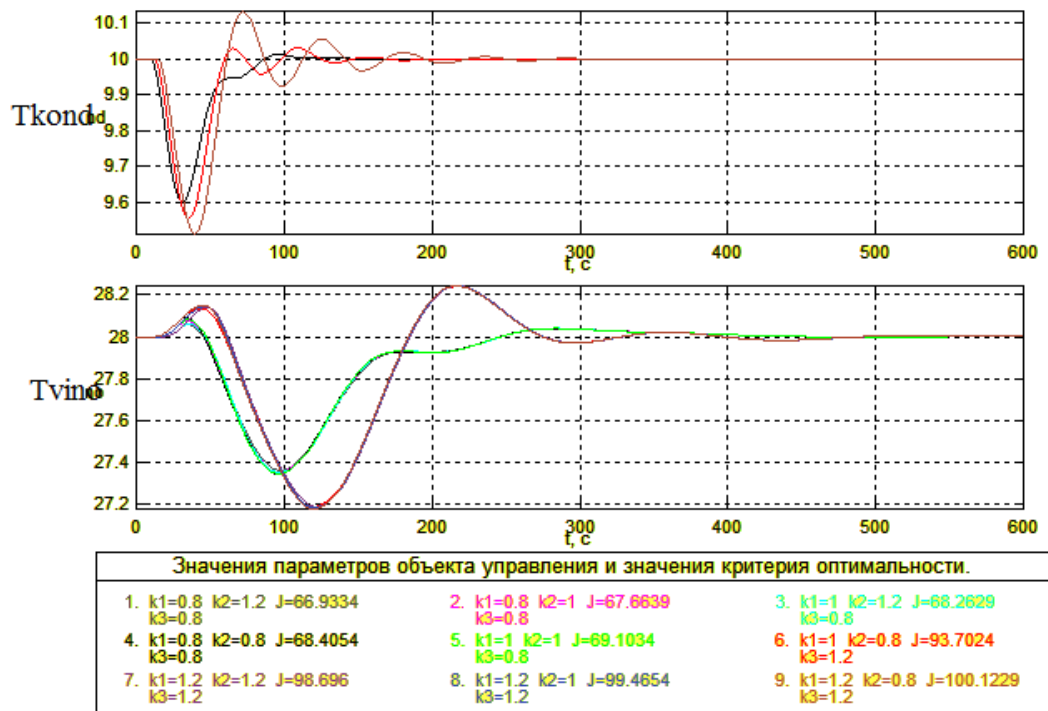


Рис. 3.31 – Вікно результатів перевірки САР з нейрорегулятором на грубість

### 3.9. Аналіз функціонування різних САР за якістю регулювання за умов впливів, що збурюють, випадкового характеру

На заключному етапі розробки САР був виконаний порівняльний аналіз функціонування різних САР за якістю регулювання. У результаті моделювання різних САР були отримані графіки перехідних процесів при детермінованому і випадковому впливі, що збурює.

Таким чином, на рисунку 3.33 представлені графіки перехідних процесів при детермінованому впливі, що збурює. А на рисунку 3.34 представлені графіки сталих процесів при випадковому впливі, що збурює.

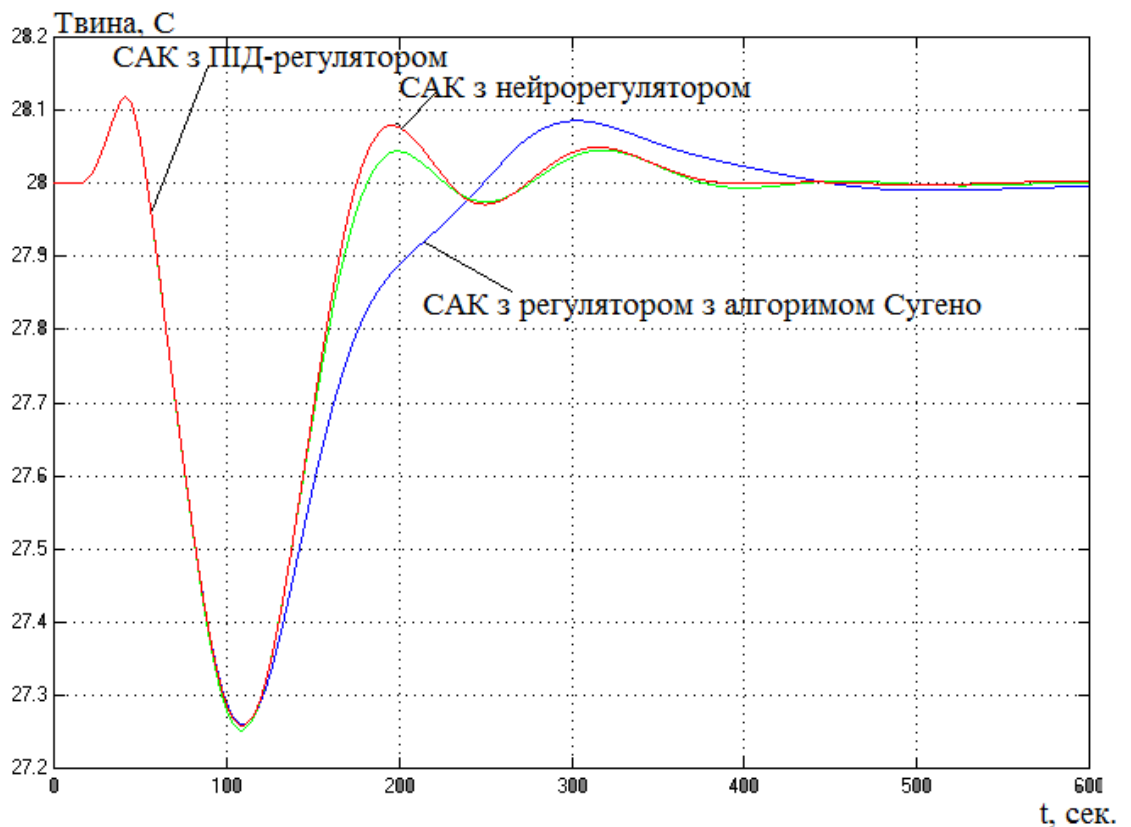


Рис. 3.32 – Графіки перехідних процесів при детермінованому впливі, що збурює, за каналом регулювання « $U_{\text{шов}} - T_{\text{вина}}$ », що отримані в результаті моделювання різних САР

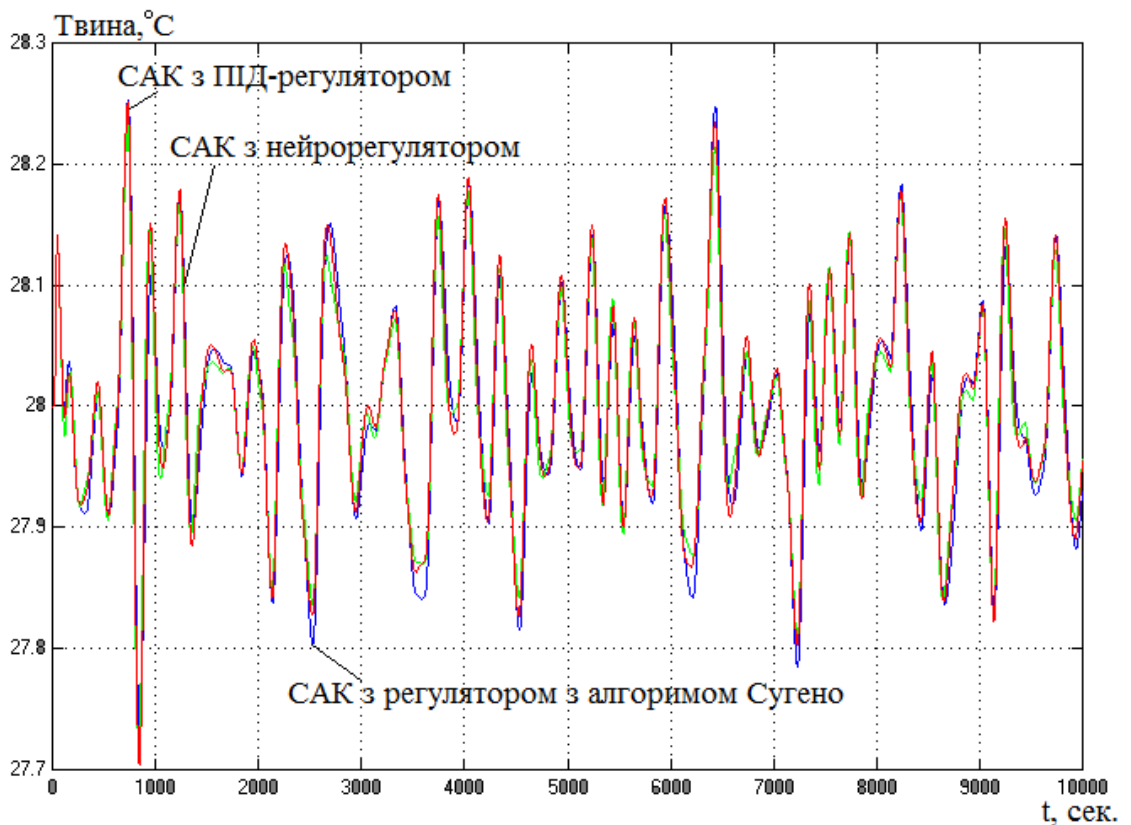


Рис. 3.33 – Графіки сталих процесів при випадковому впливі, що збурює, за каналом регулювання « $U_{\text{шов}} - T_{\text{вина}}$ », що отримані в результаті моделювання різних САР

Також на основі аналізатора імовірнісних характеристик, що був вбудований у середовище MATLAB\Simulink, була визначена якість регулювання різних САР при випадкових впливах, що збурюють. У даному випадку порівнюються середні квадратичні відхилення значень похибки регулювання різних САР при випадкових впливах. Результати представлені на рисунку 3.35, у якому зображене вікно результатів аналізу, де показані середні квадратичні відхилення і математичні очікування значень похибки регулювання при різних САР.

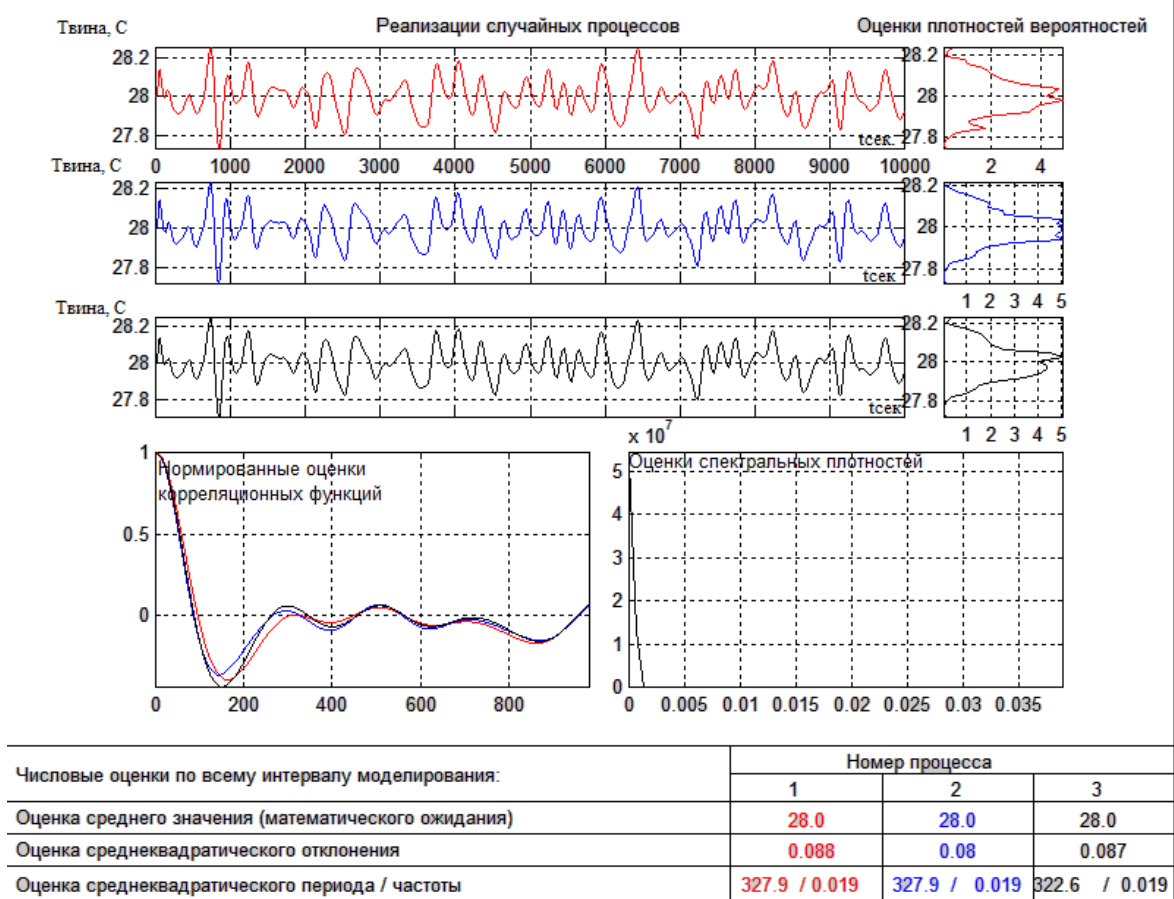


Рис. 3.34 – Вікно результатів аналізу аналізатора імовірнісних характеристик

З рисунків вище можна побачити, що САР з нейронним регулятором працює майже ідентично з САР з ПІД-регулятором. В кінцевому випадку показники функціонування різних САР за якістю регулювання були зведені до таблиці 3.2.

З таблиці 3.2 можливо зробити остаточний висновок, що за якістю регулювання САР з нейронним регулятором та САР з традиційним ПІД-регулятором функціонують за якістю регулювання практично ідентично при об'єкті з нелінійною статичною характеристикою за каналом регулювання « $U_{\text{шов}} - T_{\text{вина}}$ ».

Таблиця 3.2 – Показники якості регулювання різних САР

САР	САР з традиційним ПД-регулятором	САУ з нечітким регулятором	САР з регулятором, представленим штучною нейронною мережею
Час регулювання, с	≈160	≈180	≈160
Максимальне динамічне відхилення, °С	≈0,74	≈0,74	≈0,74
Інтегральний показник	61,44	76,44	60,86
Середнє квадратичне відхилення, °С	0,087	0,088	0,08

### 3.10 Висновки

В процесі виконання роботи була розроблена модель САР з нечітким регулятором. Нечіткий регулятор розроблений у відповідному редакторі середовища MATLAB\Simulink. Були визначені функції приналежності до нечітких множин, сформована база правил функціонування регулятора у табличному вигляді та у вікні відповідного редактора, визначений алгоритм

нечіткого виводу – Сугено 0-порядку. Також було розглянуто алгоритм Мамдані.

Було встановлено, що за якістю регулювання така САР функціонує трохи гірше за САР з традиційним ПД-регулятором при об'єкті керування з нелінійною статичною характеристикою за каналом регулювання « $U_{\text{шов}} - T_{\text{вина}}$ ».

Для подальшої реалізації параметричної оптимізації САР, за характеристиками нечіткого регулятора було, виконано тренування визначеної штучної нейронної мережі, яка може виконувати роль нейрорегулятора. Тренування штучної нейронної мережі було виконано також засобами середовища MATLAB\Simulink. Була розроблена модель САР з регулятором, що представляється штучною нейронною мережею. Параметрична оптимізація САР з нейрорегулятором була проведена при тих же умовах, що і САР з традиційним ПД-регулятором.

У результаті моделювання різних САР було встановлено, що за якістю регулювання САР з нейронним регулятором та САР з традиційним ПД-регулятором функціонують за якістю регулювання практично ідентично, це за інтегральним показником якості регулювання відповідно 61,44, при САР з нечітким регулятором – 76,44, і 60,86 при САР з традиційним ПД-регулятором. Середнє квадратичне відхилення регульованої зміни від заданого значення, при випадкових впливах, що збурюють, практично однаково при різних САР: з традиційним ПД-регулятором – 0,08, з нечітким регулятором – 0,088, з нейрорегулятором – 0,087.

Таким чином, на основі виконаного синтезу і аналізу різних САР можна зробити заключний висновок, що нейрорегулятор має принципову придатність для практичного використання на базі промислового контролера при об'єктах керування з нелінійними характеристиками за каналам регулювання.

## РОЗДІЛ 4

### РОЗРОБКА СТРУКТУРИ МОДЕРНІЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ ПРОЦЕСОМ ВАКУМНОЇ ДЕАЛКОГОЛІЗАЦІЇ ВИНА

#### 4.1. Опис загальних принципів побудови системи керування.

Найчастіше на змінні об'єкта управління (ОУ) регламентами накладаються обмеження, порушення яких може мати дуже несприятливі наслідки. Вони можуть призводити до аварійних зупинок обладнання, втрат сировини та енергоносіїв, зниження якості продукції, необхідність мати великий штаб персоналу та ремонтних служб. Ситуація посилюється, якщо найбільш ефективні режими роботи ОУ близькі до гранично-допустимих, ОУ є суттєво нестаціонарним, схильний до інтенсивних неконтрольованих збурень, має запізнення в каналах управління, а наслідки порушень тяжкі.

САР такими об'єктами включають:

- а) системи (контур) автоматичного регулювання (САР) регламентованих змінних  $y(t)$ , що забезпечують їх стабілізацію на околицях заданих значень (уставок)  $y^{зд}(t)$ ;
- б) системи захисту, що запобігають розвитку несприятливих наслідків, обумовлених порушенням регульованої змінної  $y(t)$  встановлених регламентом гранично допустимих значень  $y^{гп}$  (подія S).

При побудові САУ такими об'єктами є дві альтернативи:

1. Реалізувати звичайну САР, задане значення для якої встановити відповідним найбільш несприятливій ситуації, яка може виникнути під час роботи системи та таким чином запобігти порушенню  $y(t) y^{гп+}$ .

Недоліки:

- низька ефективність роботи ОУ завжди;
- найнесприятливіша ситуація ап'орі невідома, тому гарантувати відсутність події S неможливо навіть з урахуванням прийнятих несприятливих умов.

2. Реалізувати систему гарантуючого управління (СГУ), тобто, таку САУ, яка поряд із функціями регулювання та захисту, реалізувала б

функцію гарантованого дотримання встановлених обмежень з заданою ймовірністю.

Концепція побудови таких систем полягає у безперервному оцінюванні (на ковзному інтервалі часу) поточного значення ймовірності відсутності порушення та такого коригування режиму роботи об'єкта, щоб ця ймовірність відповідала наперед заданій ймовірності. Безпосередньо вимірювати значення цієї ймовірності неможливо, тому необхідна розробка спеціальної моделі, яка дозволить за оцінками доступних для вимірювання змінних отримати необхідну оцінку. Таку модель можна назвати моделлю порушення регламенту (МНР).

Альтернативні варіанти МНР для оцінки поточної ймовірності виникнення події S представлені у формі виразів для розрахунку поточних значень оцінок середньої інтенсивності подій S. Вихідною інформацією цих розрахунків є результати оцінювання ймовірнісних характеристик процесу  $y(t)$  і значення  $y^{rp\pm}$ . Вигляд МНР залежить від прийнятої моделі зміни  $y(t)$  і конкретно від виду прийнятої моделі зміни  $m_y(t)$ . У нашому випадку приймаємо, що  $m_y(t)$  є апіорі невідомою функцією часу. Тоді розрахунок заданого допустимого значення  $y^{здд\pm}$  вироблений безпосередньо за моделлю порушення регламенту буде здійснюватися за такою формулою:

$$y^{здд\pm}(t) = y^{rp\pm} \mp \hat{\sigma}_y \left| \sqrt{2 \ln \left| \frac{T \hat{\sigma}_y}{2\pi \hat{\sigma}_y \ln P_S^{\Gamma}(y^{rp\pm}, T)} \exp \left\{ -\frac{\hat{m}_y^2(t)}{2\hat{\sigma}_y^2} \mp \frac{\sqrt{2\pi} \hat{m}_y(t)}{\hat{\sigma}_y} \Phi \left( \frac{\hat{m}_y(t)}{\hat{\sigma}_y} \right) \right\} \right| \right|,$$

$$\text{где } \Phi(\dots) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z \exp \left\{ -\frac{1}{2} z^2 \right\} dz, \quad \Phi'(\dots) = \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{1}{2} z^2 \right) - \text{інтеграл}$$

вероятности и его производная;

$\hat{m}_y$ ,  $\hat{\sigma}_y$ ,  $\hat{\sigma}_y$ ,  $\hat{A}_y$ ,  $\hat{\omega}_y$  – оценки соответствующих характеристик случайных процессов, вычисляемые на скользящем интервале времени  $T_{кст}$ ;

$\hat{m}_y(t)$ ,  $\hat{m}_y(t)$  – оценки изменяющегося математического ожидания и его первой производной, вычисляемые на скользящем интервале времени  $T_m \ll T_{кст}$ .

Оценки характеристик случайных процессов, вычисляемые в реальном времени на скользящих интервалах, определяются усреднением на  $T_{\text{кст}}$  (или на  $T_m \ll T_{\text{кст}}$  для  $m(t)$ ). При усреднении с равномерным взвешиванием реализуются следующие выражения:

$$\hat{m}_y = \frac{1}{T} \int_t^{t+T} y(t) dt, \quad \hat{m}_{\dot{y}} = \frac{1}{T} \int_t^{t+T} \dot{y}(t) dt, \quad \hat{m}_y(t) = \frac{1}{T_m} \int_t^{t+T_m} y(t) dt, \quad \hat{m}_{\dot{y}}(t) = \frac{1}{T_m} \int_t^{t+T_m} \dot{y}(t) dt;$$

$$\hat{\sigma}_y = \frac{1}{T} \int_t^{t+T} (y(t) - \hat{m}_y)^2 dt, \quad \hat{\sigma}_{\dot{y}} = \frac{1}{T} \int_t^{t+T} (\dot{y}(t) - \hat{m}_{\dot{y}})^2 dt, \quad (3.6)$$

$$\hat{\mu}_y^4 = \frac{1}{T} \int_t^{t+T} (y(t) - \hat{m}_y)^4 dt, \quad \hat{\mu}_{\dot{y}}^4 = \frac{1}{T} \int_t^{t+T} (\dot{y}(t) - \hat{m}_{\dot{y}})^4 dt.$$

Структурна схема СГУ з розрахунком допустимого заданого значення контуру регулювання безпосередньо за моделлю порушення регламенту представлена на рис. 4.1.

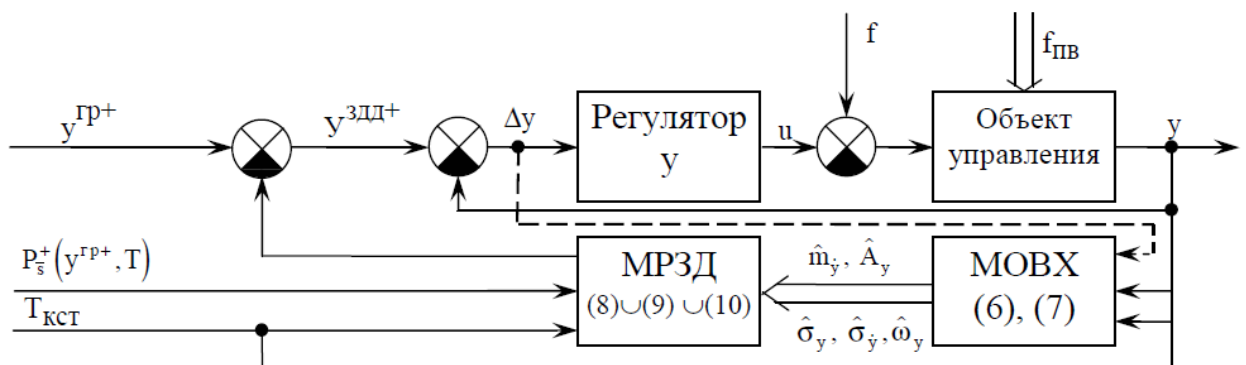


Рис. 4.1 – Структурна схема СГУ з розрахунком допустимого заданого значення контуру регулювання безпосередньо за моделлю порушення регламенту (МРЗД – модуль розрахунку допустимого заданого значення; МОВХ – модуль оцінки ймовірнісних характеристик).

Сигнал від помилки регулювання та поточний стан змінної надходить на вхід МОВХ, куди також надходить  $T_{\text{кст}}$  - мінімальний інтервал часу, де виконуються умови квазістаціонарних  $y^0(t)$ . У МОВХ йде оцінка складових центрованого випадкового процесу (середньоквадратичне відхилення поточної змінної, середньоквадратичне відхилення її похідної, частота та амплітуда випадкової складової, а також математичне очікування похідної від поточної змінної). Далі вище зазначені змінні надходять на вхід МРЗД куди також надходить  $T_{\text{кст}}$ , і  $P_s^+(y^{р+}, T)$ -імовірність відсутності події  $S$ , де

формується таке значення, що після відрахування з граничним значенням в  $u^{гр}$  було таке значення  $u^{здд+}$ , яке разом із контуром регулювання забезпечувало роботу нашої САУ з наперед заданою ймовірністю безаварійності.

#### 4.2. Розробка структурної схеми САК та опис принципу її дії.

На основі САР підвищеної динамічної точності розроблена структура модернізованої САР, що передбачає керування температурою конденсату. Структурна схема САУ представлена на рис. 4.2.

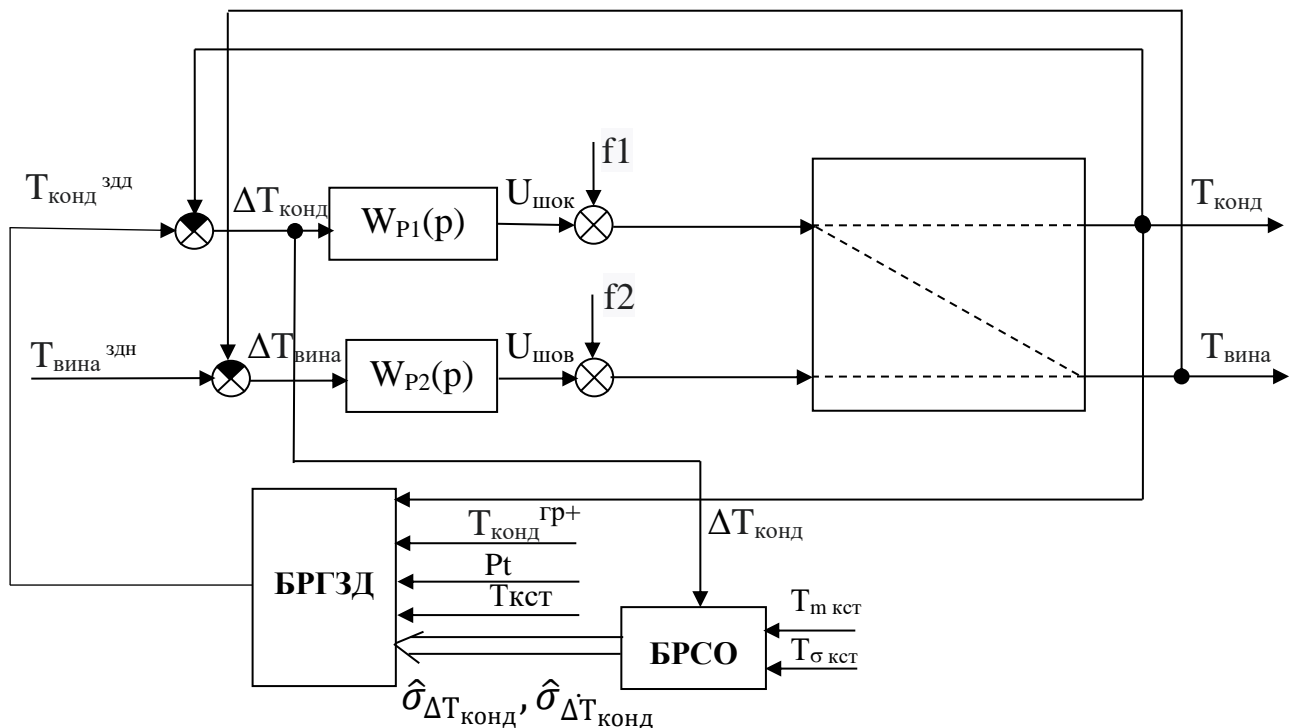


Рис. 4.2 – Структурна схема модернізованої системи керування яка додатково реалізує функцію гарантуючого керування температурою конденсату (БРГЗД – блок розрахунку гарантуючого заданого значення; БРСО – блок розрахунку статистичних оцінок).

У представленій системі передбачено регулювання температурою конденсату. Система стабілізує температуру конденсату на заданому значенні. Регулятор температури конденсату отримує задне значення від блоку БРГЗД алгоритма гарантуючого керування. Алгоритм гарантуючого управління реалізований двома блоками: БРСО та БРГЗД. У блоці БРСО на ковшних

інтервалах часу розраховуються оцінка середньоквадратичного відхилення помилки регулювання температури конденсату та оцінка середньоквадратичного відхилення похідної помилки регулювання температури конденсату відповідно до формул, наведених у пункті 2.1. Ковзаючи інтервали часу задаються як параметри налаштування системи. У блоці БРГЗД безпосередньо з урахуванням моделі порушення регламенту виконується розрахунок гранично допустимого заданого значення температури конденсату. Розрахунок виконується на основі: граничного значення температури конденсату –  $T_{\text{конд}}^{\text{гр}+}$ ; заданого значення ймовірності відсутності порушення обмеження –  $P_t$ ; заданого значення інтервалу часу, на якому гарантується відсутність порушення обмеження; поточних значень оцінок ймовірнісних характеристик розрахованих у блоці БРСО та поточного значення температури конденсату.

4.3. Обґрунтування настроювальних параметрів нових алгоритмів керування.

Настроювальні параметри регулятора температури конденсату залишимо отриманими раніше при проведенні оптимального параметричного синтезу САР.

Налаштовуванні параметри системи гарантуючого управління задаватимемо на основі вимог технологічного регламенту та динамічних властивостей помилки регулювання як випадкового процесу. Зокрема, для розрахунку інтервалів зосередження використовуватимемо оцінку середньоквадратичного періоду помилки регулювання.

4.4 Доопрацювання математичної моделі ОР для вирішення задачі синтезу та аналізу системи гарантуючого управління.

Для виконання завдання синтезу та аналізу роботи системи гарантуючого управління доповнимо модель об'єкта керування моделями параметричних збурень (зміна запізнення в каналі регулювання температури конденсату) та моделлю зміни дисперсії координатних збурень.

Повна схема моделювання об'єкта регулювання представлена на рис. 4.3.

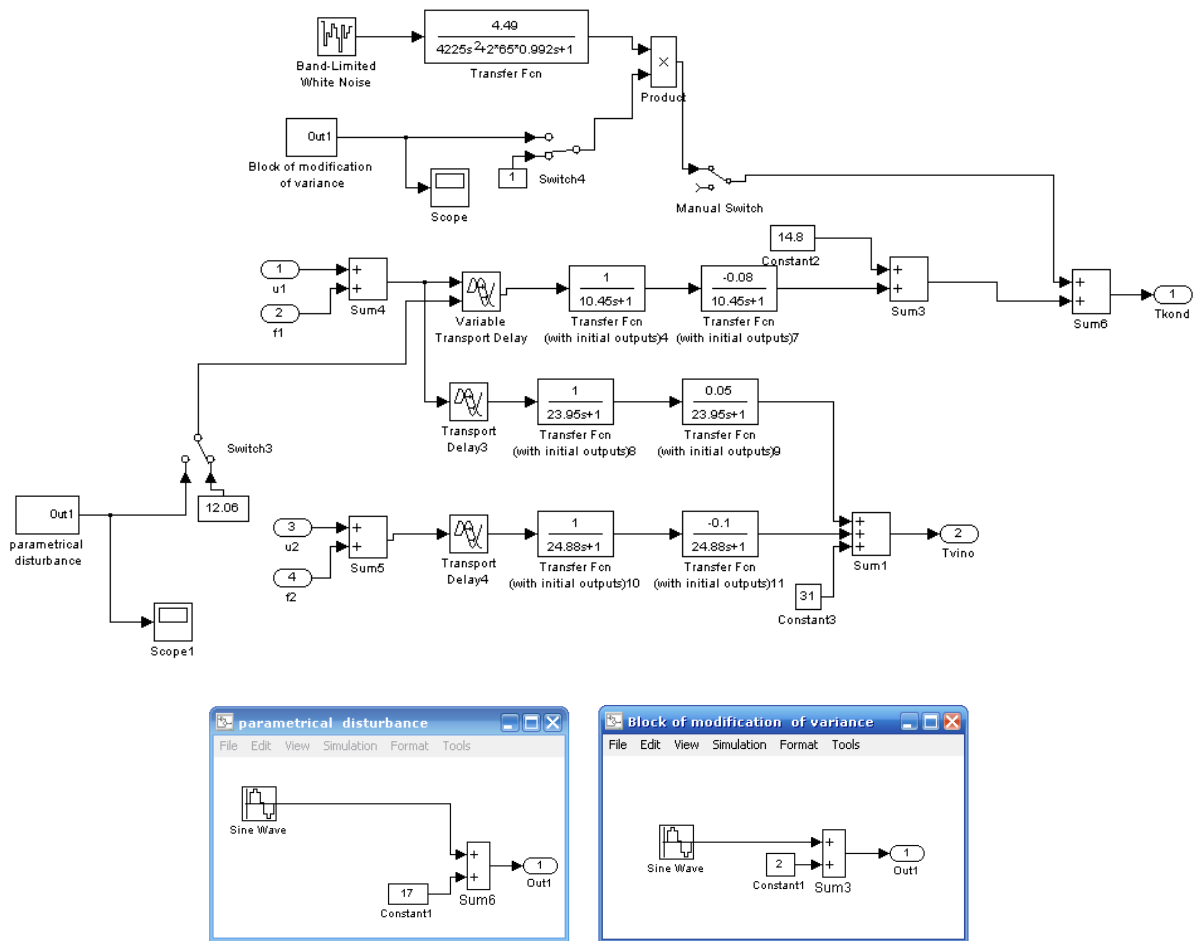


Рис. 4.3 – Схема моделювання об'єкта керування.

Параметричні збурення та зміна дисперсії координатних збурень реалізовуватимемо на основі гармонійного сигналу. Періоди гармонійних сигналів будуть різними.

4.5. Розробка імітаційної моделі системи гарантуючого управління температурою конденсату та дослідження її роботи.

Доповнимо схему моделювання функціональними блоками, що реалізують алгоритми гарантуючого управління і проведемо налаштування їх параметрів. Отримана схема моделювання представлена на рис. 3.2. А схеми внутрішніх структур блоків BRGZD та BRSO представлені на рис. 3.3 та 3.4.

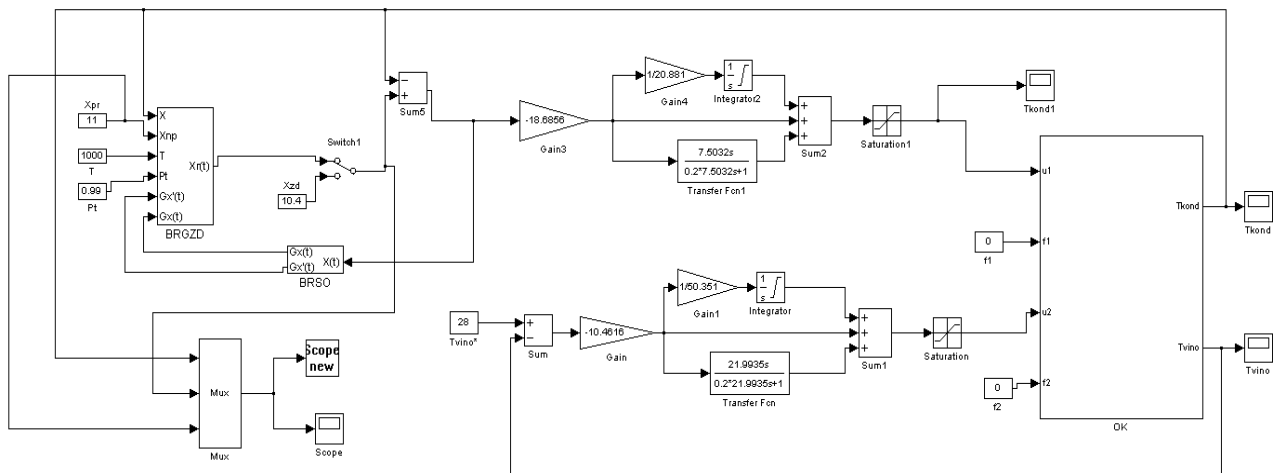


Рис. 4.4 – Схема моделювання САР с алгоритмом гарантуючого управління температури конденсату після дрослювання.

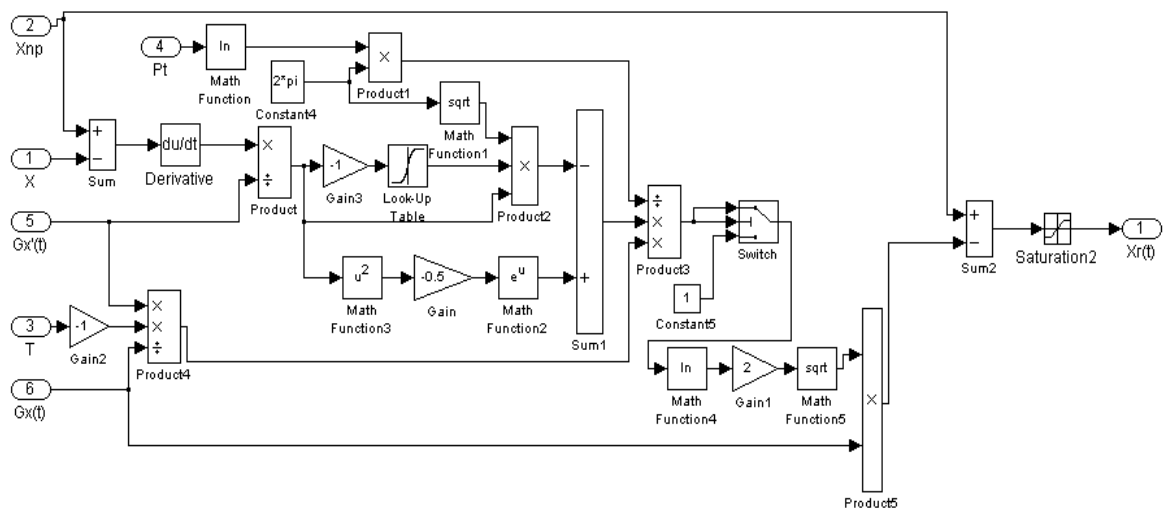


Рис. 4.5 – Внутрішня структура блоку BRGZD.

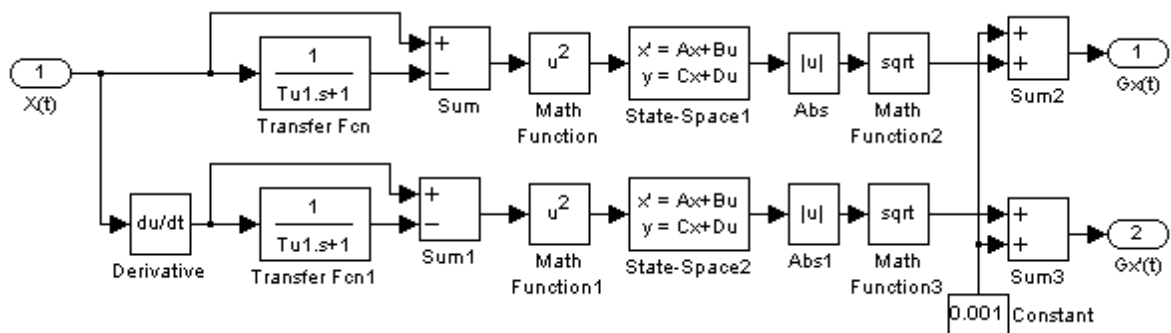


Рис. 4.6 – Внутрішня структура блоку BRSO.

Проведемо дослідження роботи отриманої системи гарантуючого управління порівняно зі стабілізуючою САР в умовах діючих стаціонарних та нестаціонарних (змінюється дисперсія) координатних збурень випадкового характеру.

На рис. 4.7 та 4.8 представлені результати моделювання стабілізуючої САР температури конденсату після дроселювання та системи, що реалізує гарантуюче управління при фіксованому значенні середньоквадратичного відхилення збурень. Задане значення температури конденсату встановлено на рівні 10.4 °С. Як видно з результатів моделювання в СГУ викиди за граничне значення відсутні. Середнє видалення менше в системі, що реалізує гарантуюче управління.

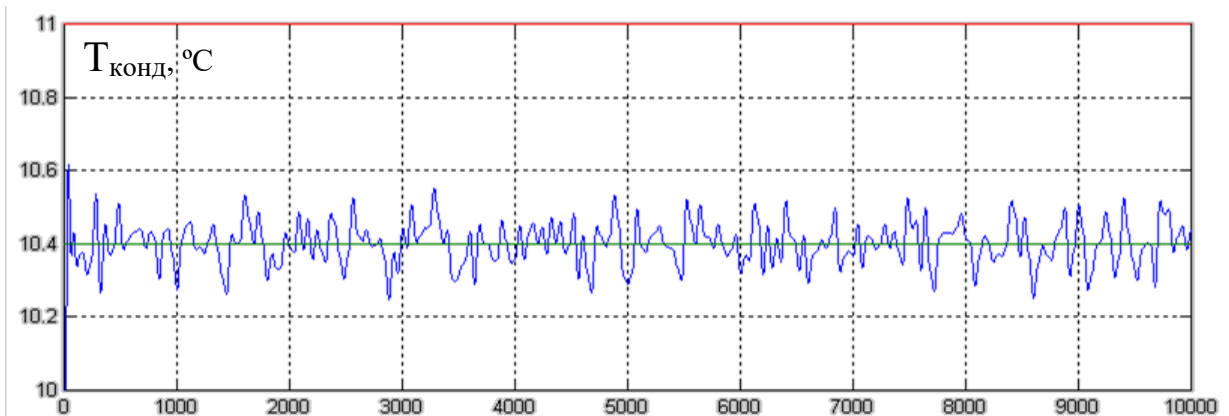


Рис. 4.7. Результати моделювання стабілізуючого САР температури конденсату при координатних збуреннях випадкового характеру з фіксованим середньоквадратичним відхиленням.

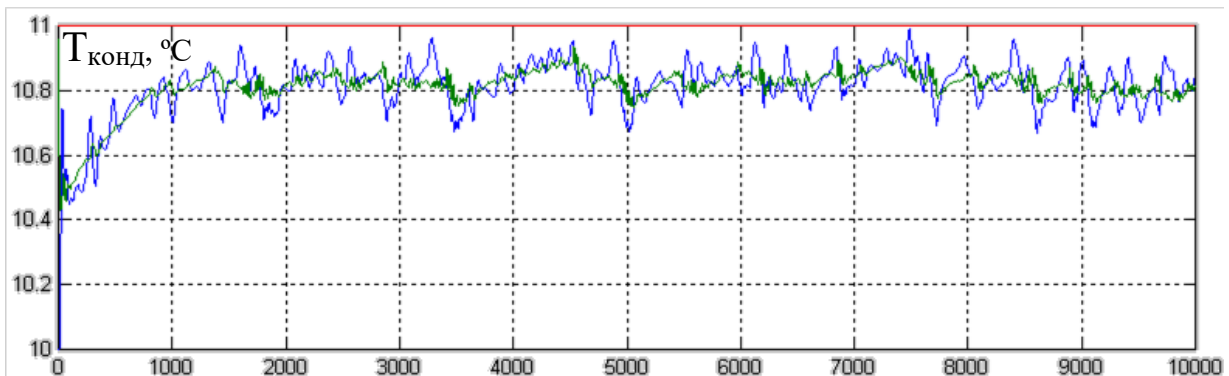


Рис. 4.8. Результати моделювання СГУ температури конденсату при координатних збуреннях випадкового характеру з фіксованим середньоквадратичним відхиленням.

При моделюванні обох варіантів систем управління в умовах, коли середньоквадратичне відхилення координатних збурень змінюється за синусоїдою (рис. 4.9), робота стабілізуючої САР погіршується (рис. 4.10). Система, що реалізує гарантує управління залишається працездатною (рис. 4.11). А середнє видалення температури конденсату від граничного значення збільшується.

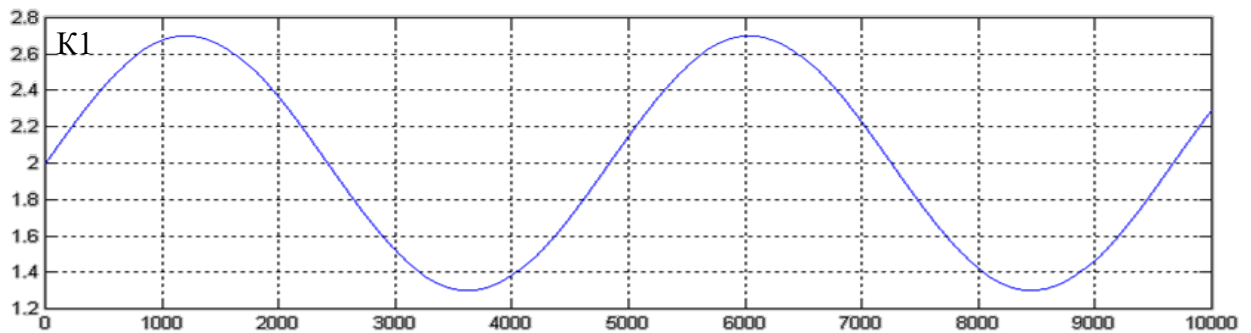


Рис. 4.9. Графік, що відбиває характер зміни середньоквадратичного відхилення координатних неконтрольованих збурень (зміна температури конденсату).

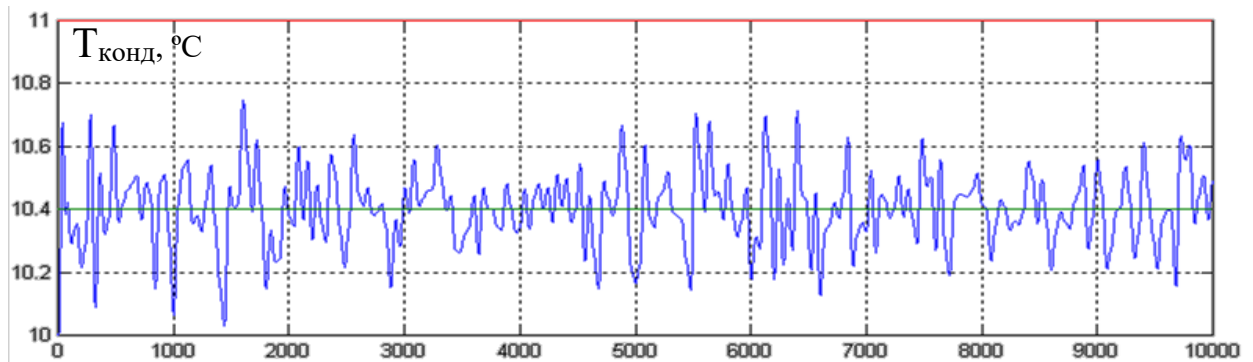


Рис. 4.10. Результати моделювання стабілізуючої САР.

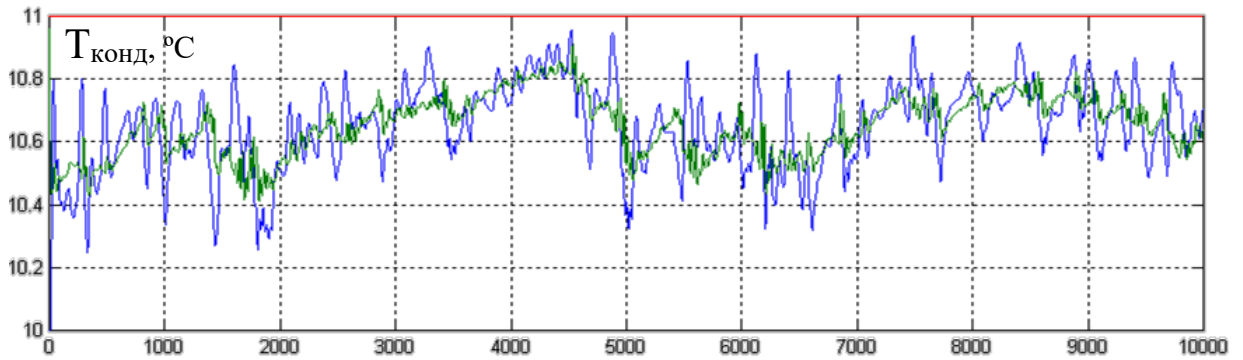


Рис. 4.11. Результати моделювання системи, що реалізує гарантуюче управління.

При моделюванні обох варіантів систем управління в умовах, коли діють параметричні збурення (рис. 4.12) та середньоквадратичне відхилення координатних збурень не змінюється, робота стабілізуючої САР погіршується (рис. 4.13). Система, що реалізує гарантуюче управління залишається працездатною (рис. 4.14). Є один вихід за граничне значення, але середнє віддалення від граничного значення збільшується незначно.

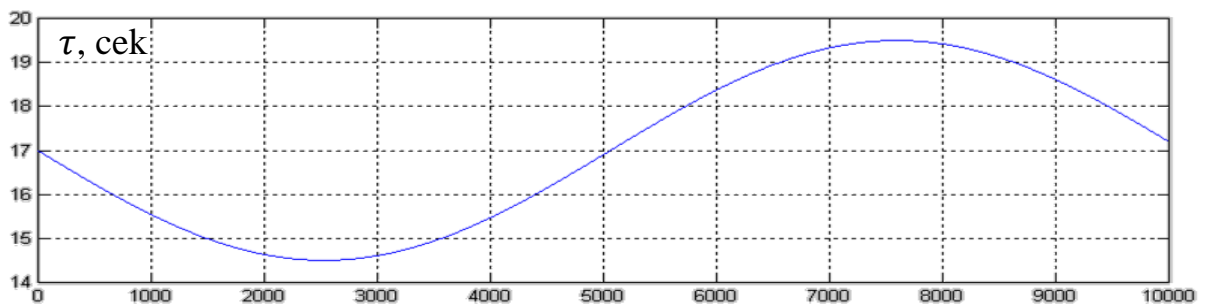


Рис. 4.12. Графік, що відображає характер зміни запізнення у каналі регулювання температури конденсату.

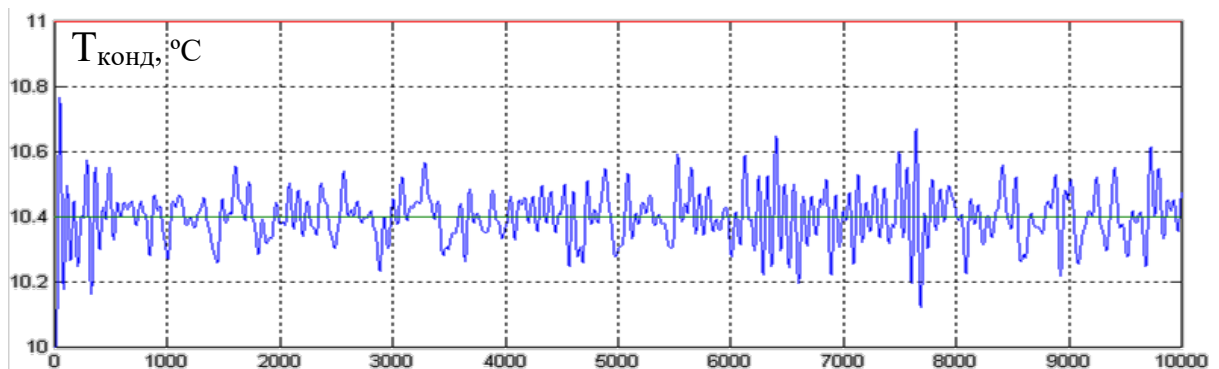


Рис. 4.13. Результати моделювання стабілізуючої САР.

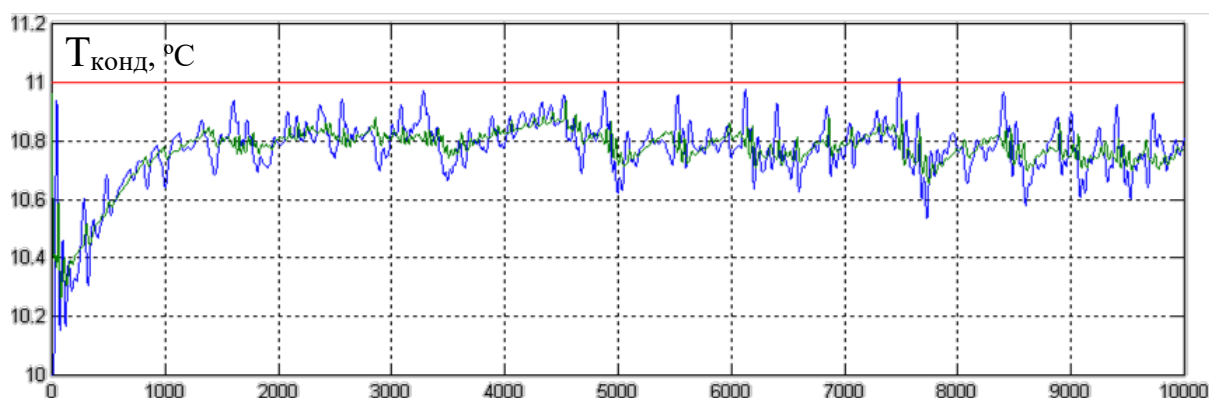


Рис. 4.14. Результати моделювання системи, що реалізує гарантує управління.

Результати моделювання обох варіантів систем управління в умовах, коли діють параметричні збурення (рис. 4.15) та середньоквадратичне відхилення координатних збурень змінюється за гармонічним законом (4.16), робота стабілізуючої САР погіршується (рис. 4.17). СГУ залишається працездатною та середнє видалення температури конденсату від верхньої межі залишається менше, ніж у стабілізуючій САР, хоча є один вихід за гранично припустиме значення тиску дроселювання.

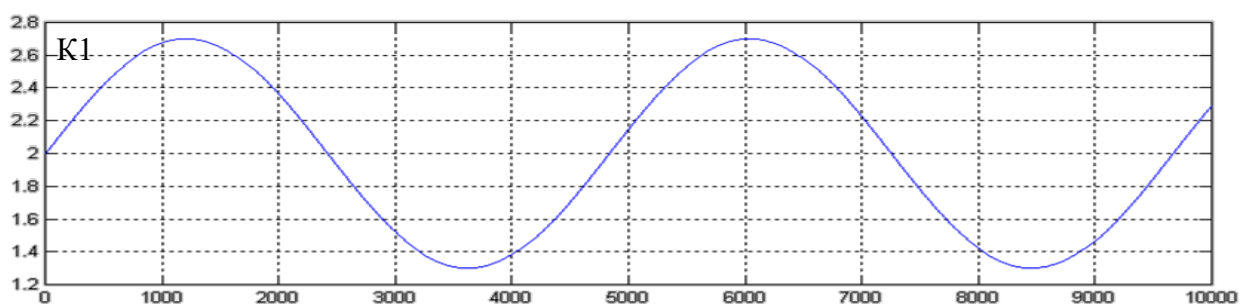


Рис. 4.15. Графік, що відбиває характер зміни середньоквадратичного відхилення координатних неконтрольованих збурень (зміна температури конденсату).

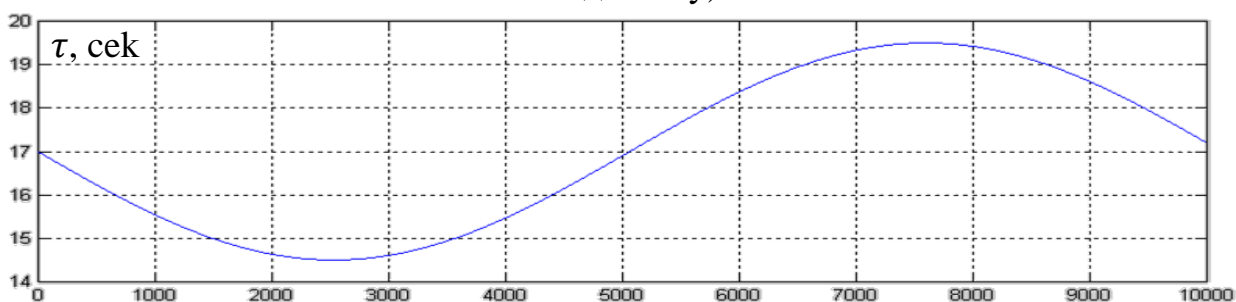


Рис. 4.16. Графік, що відображає характер зміни запізнення у каналі регулювання температури конденсату.

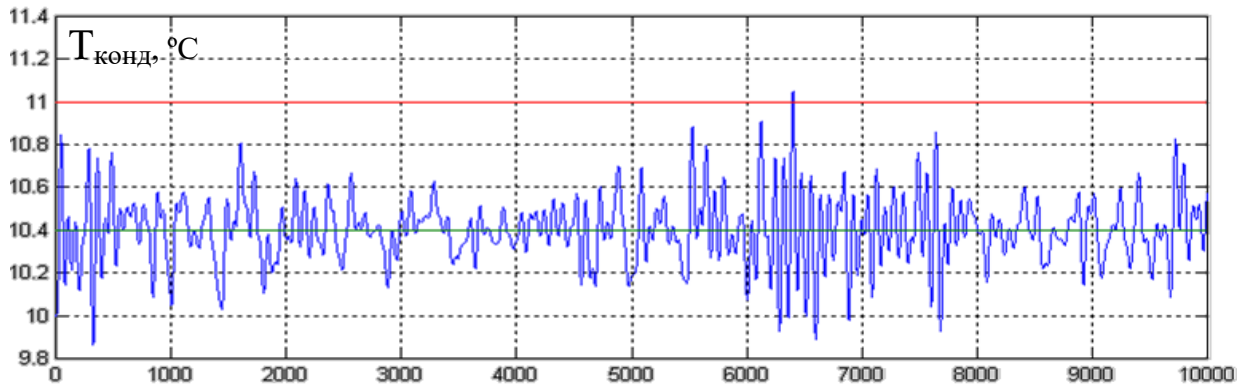


Рис. 4.17. Результати моделювання стабілізуючої САР.

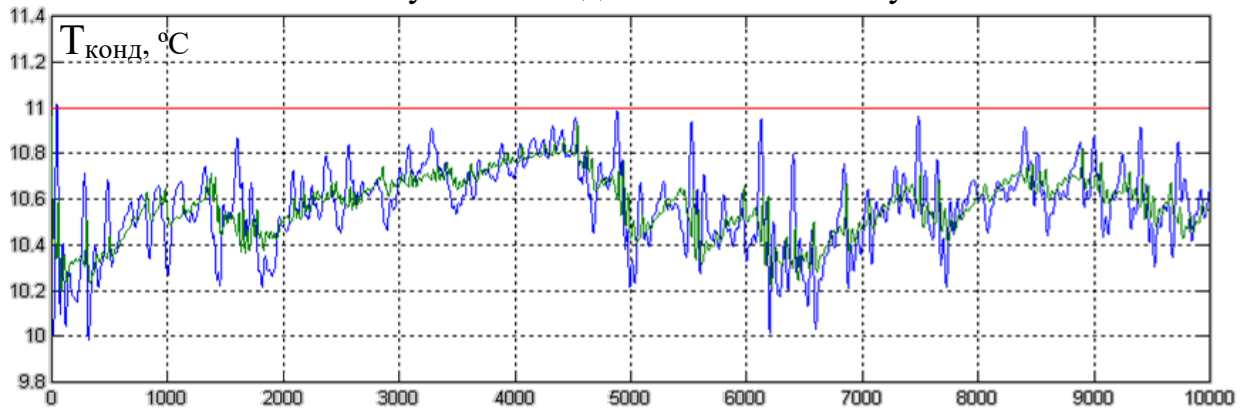


Рис. 4.18. Результати моделювання системи, що реалізує гарантує управління.

За результатами проведених досліджень можна дійти висновку, що система, яка реалізує гарантує управління температури конденсату при дії нестаціонарних збурень забезпечує виконання процесу деалкоголізації вина з використанням парокompресійного теплового насосу, підтримує температуру конденсату максимально близько до верхньої межі регламенту, забезпечуючи зниження питомих енерговитрат. Стабілізуючі САР для вирішення такого завдання не підходять.

Розглянемо вплив значення ймовірності відсутності порушення регламенту на якість роботи системи гарантує керування. Проведемо моделювання системи за різних значень ймовірності безаварійного режиму роботи ( $P_t = [0.9, 0.99, 0.999]$ ). При значенні ймовірності  $P_t = 0.99$  проводилися попередні дослідження, тому проведемо моделювання системи гарантує управління при значеннях ймовірності  $P_t = 0.9$  і  $P_t = 0.999$ . Результати моделювання подано на рис. 3.17 та 3.18.

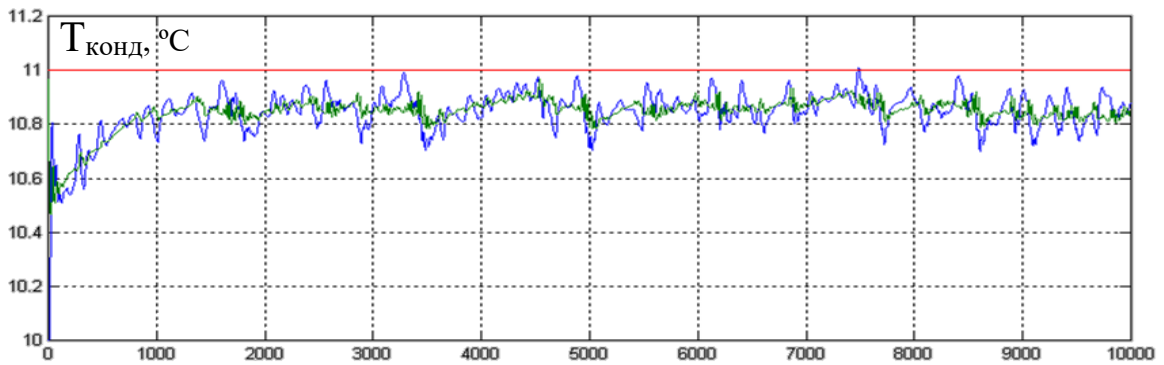


Рис. 4.19. Результати моделювання системи, що реалізує гарантуюче управління температурою конденсату після дроселювання при  $P_t = 0.9$ .

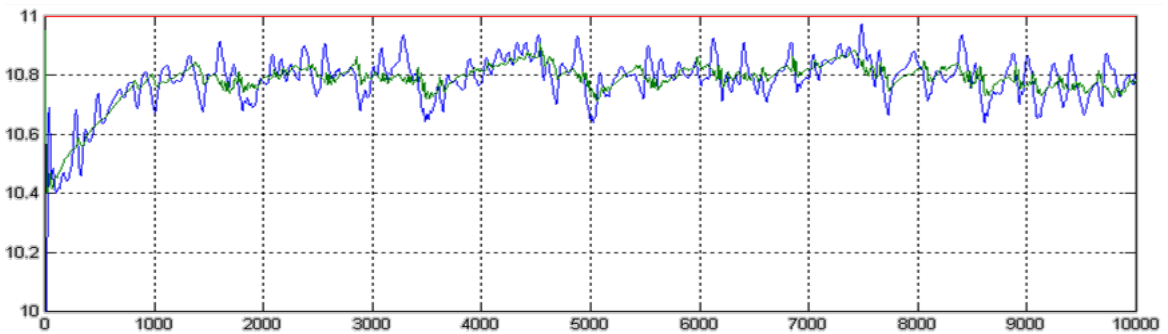


Рис. 4.20. Результати моделювання системи, що реалізує гарантуюче управління температурою конденсату після дроселювання при  $P_t = 0.999$ .

За результатами можна дійти висновку, що більше значення  $P_t$  тим далі система відводить регульовану змінну від граничного значення. При зниженні значення  $P_t$  можуть з'явитися виходи за граничне значення температури конденсату.

#### 4.6 Висновки

У розділі розроблено та досліджено модернізовану САК процесом вакуумної деалкоголізації вина з використанням парокompресійного теплового насосу. САК реалізує нову функцію: функцію гарантуючого керування температурою конденсату. Реалізація цієї функції забезпечує зниження питомих енерговитрат (витрат електричної енергії) на деалкоголізацію вина за рахунок підтримки температури конденсату поблизу верхньої межі регламенту без її порушення.

У роботі проведено низку порівняльних аналізів САР базової структури та САК, що реалізує функцію гарантуючого управління. Дослідження проводилися за умов постійно діючих обурення випадкового характеру.

За результатами проведених досліджень можна зробити висновок, що система гарантуючого управління при дії нестационарних збурень забезпечує виконання процесу деалкоголізації вина, підтримує температуру конденсату максимально близько до верхньої межі регламенту, забезпечуючи зниження питомих енерговитрат. Стабілізуючі САР для вирішення такого завдання не підходять.

За результатами можна дійти невтішного висновку, що більше значення  $P_t$  тим далі система відводить регульовану змінну від граничного значення. У всіх експериментах САУ, що реалізує функцію гарантуючого управління, є кращою, ніж САР базової структури, і подальше її використання є доцільним.

## РОЗДІЛ 5

### РЕАЛІЗАЦІЯ І ПІДГОТОВКА ДО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ РОБОТИ НА ПІДПРИЄМСТВІ

5.1. Існуюча програмно-технічна структура підсистеми керування процесом вакуумної деалкоголізації вина в потоці і формулювання задач по її розвитку для впровадження розробок.

Систему керування ПВК будемо розробляти на базі контролера S7-300 фірми Siemens. Враховуючи, що центральний процесор і модулі системи мають виконання IP20, доцільно розмістити їх у настінній шафі зі ступенем захисту не менш IP54 в окремому від технологічного устаткування приміщенні.

На наступних рисунках наведені рекомендації з монтажу Simatic S7-300 у шафі з урахуванням зазорів, можливості розсіювання тепла, а також схема заземлення.

Приміщення повинне бути опалювальним, що дозволить використовувати модулі звичайного виконання, а не з розширеним температурним діапазоном, які дорожче.

Враховуючи, що в цей час фірма Siemens випускає найбільш широку гаму устаткування з можливістю підключення по мережі PROFIBUS/MPI, доцільно вибрати CPU із вбудованим портом PROFIBUS/MPI. Це дозволить при необхідності легко розширити функціональні можливості системи керування, зокрема, зв'язати її з іншими системами й з оператором, який працює в Scada-Системі на панелі оператора.


Технічна структура системи управління розробимо на базі контролера Simatic S7-300.

Для реалізації алгоритмів управління в модулях ПЗО контролера повинні бути передбачені: аналогові входи (2); аналогові виходи (2); дискретні входи (10); дискретні виходи (9).


## Компактний центральний процесор CPU 313C-2DP

 <p>Рис.5.1 – Центральний процесор CPU 313C-2DP 6ES7313-6CG04-0AB0</p>	Інтерфейс	Profibus-DP, MPI
	Дискретні входи	16
	Тип вхідного сигналу	24 VDC
	Дискретні виходи	16
	Тип вихідного сигналу	PNP (загальний -)
	Ступінь захисту	IP 20
	Робоча температура, С	0 — 60

## Модуль аналогового виводу SM 332 6ES7332-5HB01-0AB0

 <p>Рис.5.2 – Загальний вигляд</p>	Кількість виходів	4
	Вхідна напруга L	L+ 24V DC
	Вихідні діапазони:	
	- напруги	±10V, 1...5V, 0...10V
	- струму	0...20 ма, 4...20 ма, ±20 mA
	Споживання струму по задній шині	120mA
	Потужність, що розсіюється	3W

Аналоговий модуль вводу SM 331 (6ES7331-7KF02-0AB0)

 <p>Рис.5.3 – Загальний вигляд</p>	Номінальне значення (пост. струм)	24 V
	Вхідний струм з джерела напруги навантаження L+ (без навантаження), макс.	200 mA
	Нормальна потужність, що розсіюється	3 W
	Кількість аналогових входів	8
	Макс. допустима вхідна напруга для входу напруги (межа руйнування)	75 V; 35 при тривалому навантаженні; 75 U макс. протягом 1 с (коефіцієнт заповнення 1:20)
	Макс. допустимий вхідний струм для токового входу (межа руйнування)	40 mA
	Похибки/точність Напруга відносно діапазону вхідних параметрів, (+/-) Струм відносно діапазону вхідних параметрів, (+/-)	0,1 %  0,1 %

Загальний вигляд розробленої схеми технічної структури мікропроцесорного ядра системи керування деалкоголізації вина з використанням парокомпресійного теплового насосу наведено на рис. 5.4.

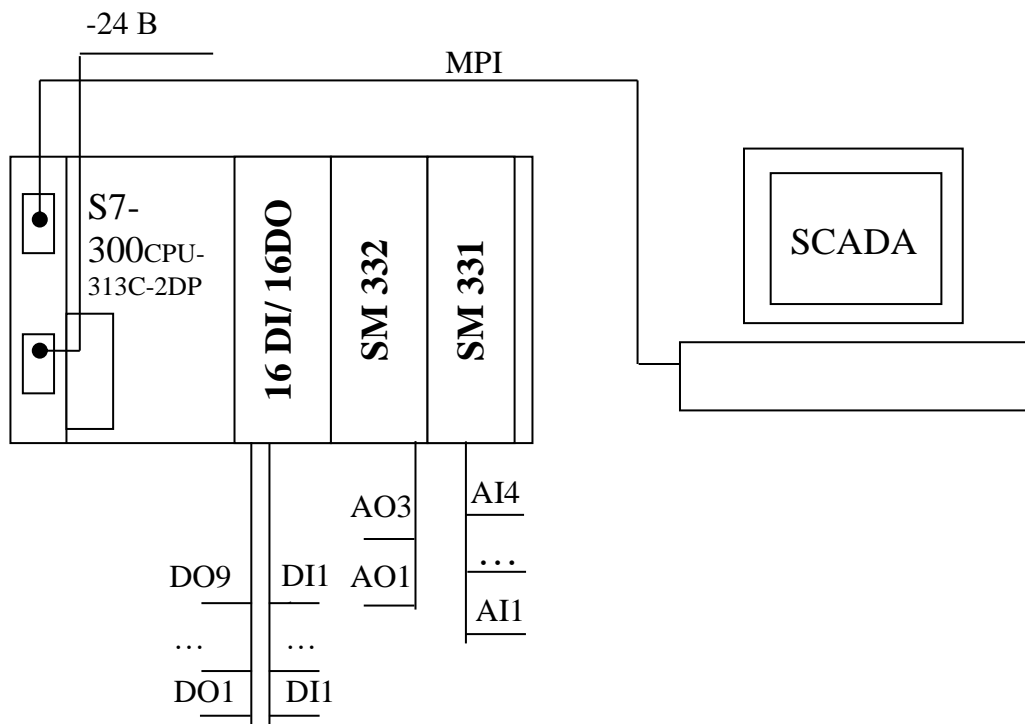


Рис. 5.4 – Схема технічної структури мікропроцесорного ядра системи керування.

У середовищі Simatic Manager Step7 створюємо проект і додаємо до його складу станцію контролера S7-300. У програмі «Hardware» виконуємо конфігурування обраного контролера і модулів ПЗО. Загальний вид вікна програми «Hardware» з встановленим контролером і модулями ПЗО представлений на рис. 5.5. При конфігуруванні необхідні модулі вибиралися з каталогу елементів, що входить в програму «Hardware».

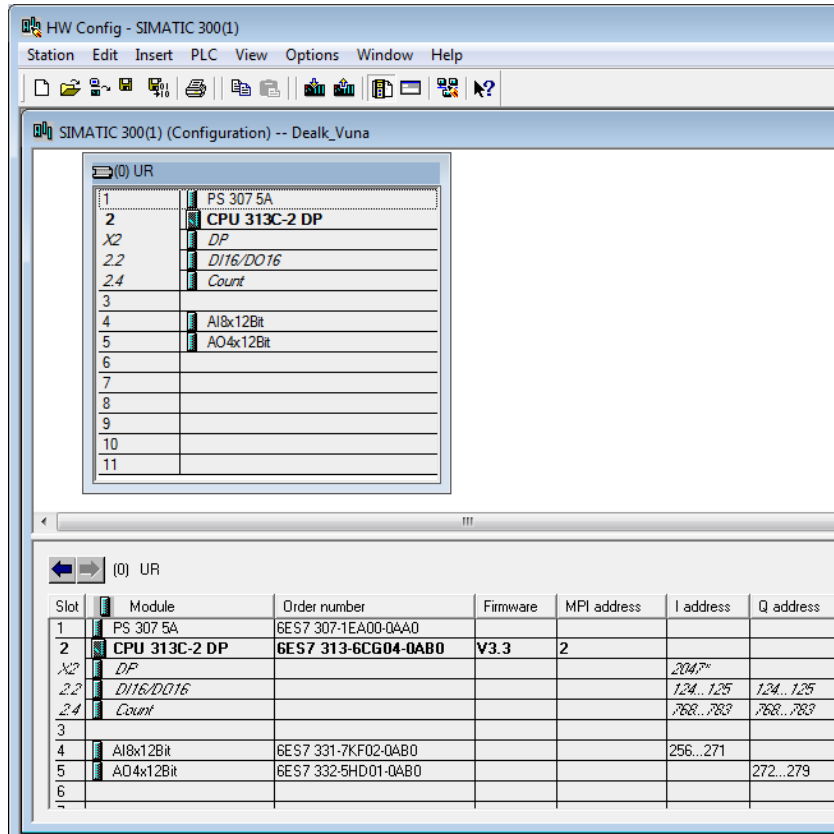


Рис.5.5 – Загальний вид вікна програми «Hardware».

Автоматизоване робоче місце оператора реалізується на базі персонального комп'ютера. Зв'язок комп'ютера з контролером виконується по мережі MPI. Процедура програмної конфігурації мережі представлена на малюнках нижче.

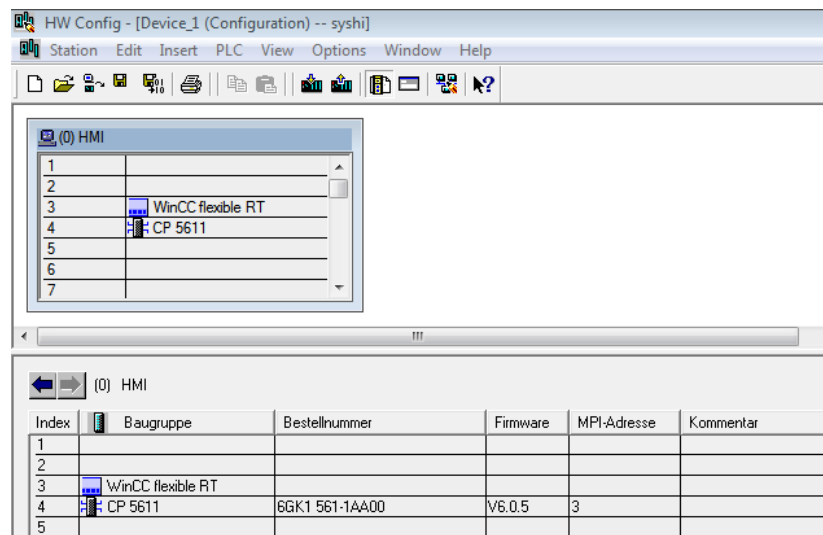


Рис.5.6 – Фрагмент вікна конфігурування PC - станції.

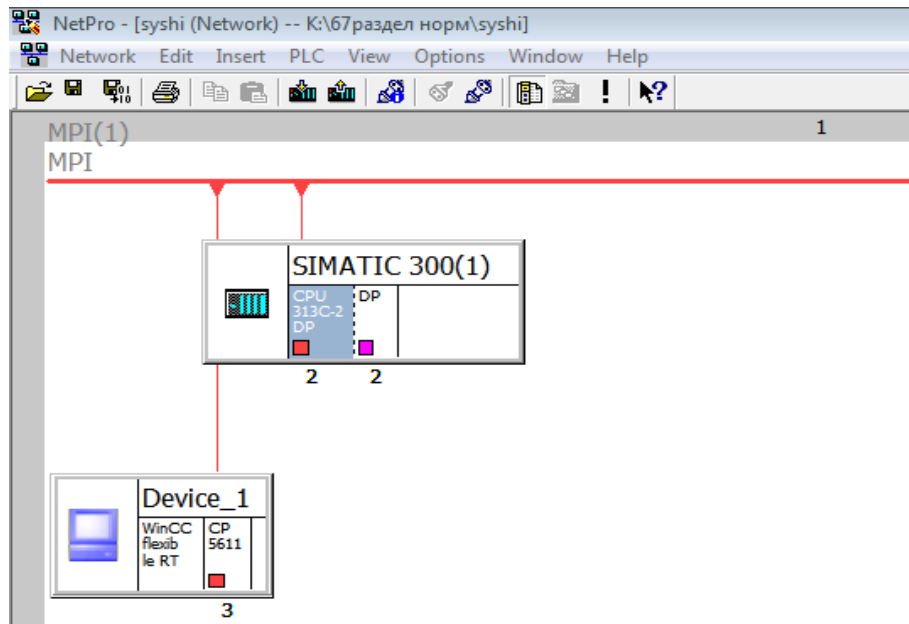


Рис. 5.7. Ілюстрація структури мережі в програмі "Net Pro".

Існуюча система керування яка реалізована на базі наведеної технічної структури доповнена функцією гарантуючого керування. Інтегрована функція гарантування не потребує істотного збільшення обчислюваних ресурсів, та реалізації додаткових каналів вводу виводу сигналів. Тому технічну структуру системи керування змінювати не потрібно.

5.2 Розробка, налаштування і тестування прикладного програмного забезпечення для реалізації алгоритмів управління.

Прикладне програмне забезпечення, що реалізує розроблені алгоритми керування може розроблятися безпосередньо у середовищі Simatic Step 7 з використанням стандартних мов програмування контролерів. Для цього потрібно розроблені у системі Matlab алгоритми керування перевести у цифрову форму та вид придатний для програмування, а потім написати відповідний код. На кожному етапі реалізації можливі помилки і в загалі такий підхід потребує багато часу. Альтернативою цьому є варіант конвертації отриманих алгоритмів з системи Matlab у потрібну середу програмування контролерів. Такий підхід дозволяє отримати компактний перевірений код при малих затратах часу.

Для програмної реалізації розроблених алгоритмів будемо використовувати пакет Simulink PLC Coder з його бібліотекою

функціональних блоків PLCLib. Для цього з використанням блоків бібліотеки PLCLib проведемо цифрову реалізацію алгоритмів керування у моделі Simulink. Отримана модель з алгоритмами підготовленими до конвертації наведена на рис. 5.7.

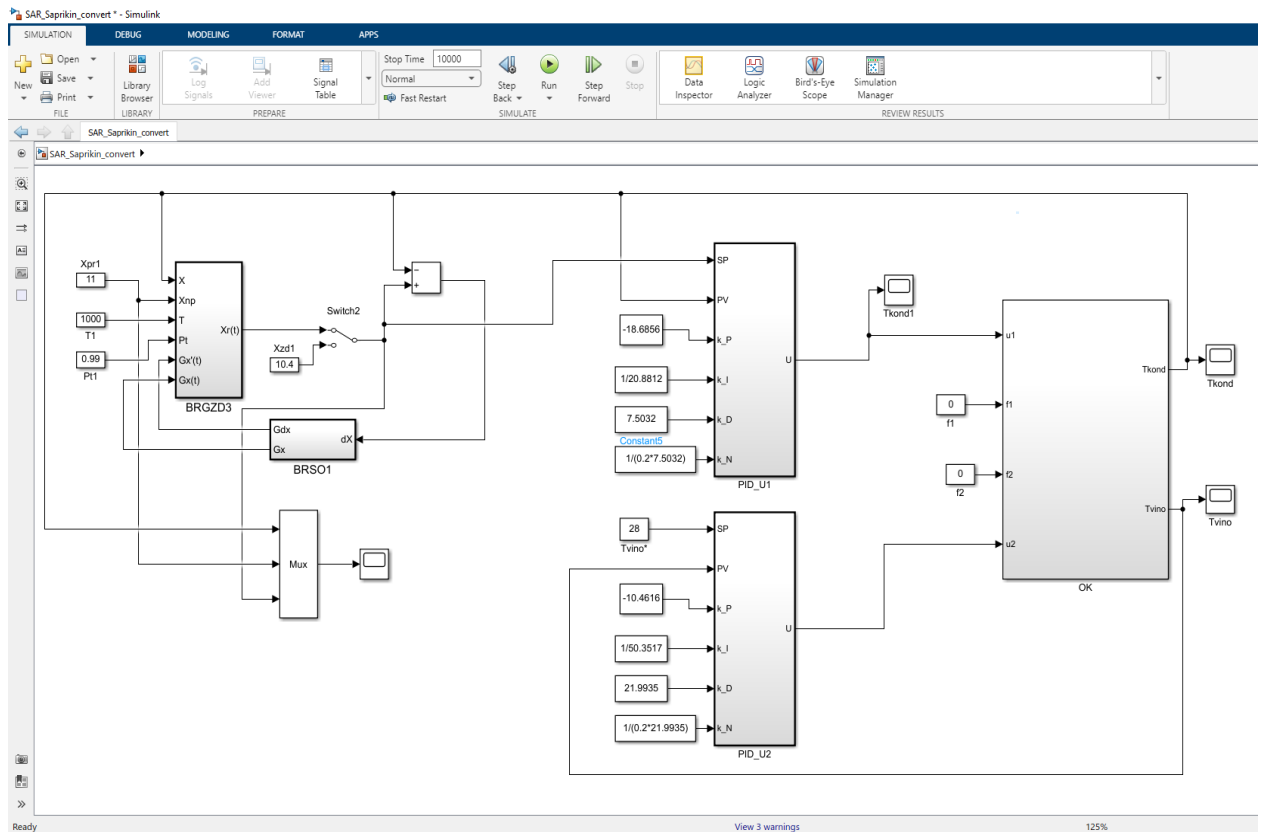


Рис. 5.7. Схема моделювання САК процесом dealкоголізації вина з цифровими алгоритмами керування, які підготовлені до конвертації.

На рис. 5.8 – 5.11 наведені схеми моделювання ПІД регулятора, алгоритмів гарантуючого керування та коригуючого зв'язку.

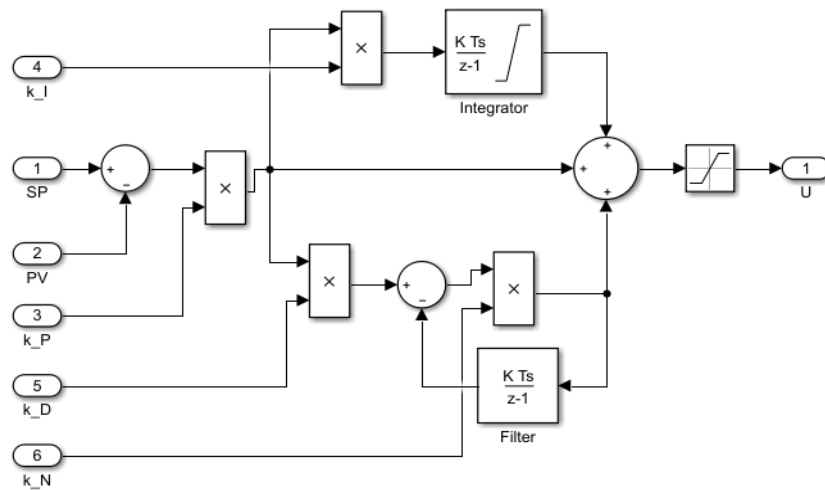


Рис. 5.8. Схема цифрового ПІД-регулятора.

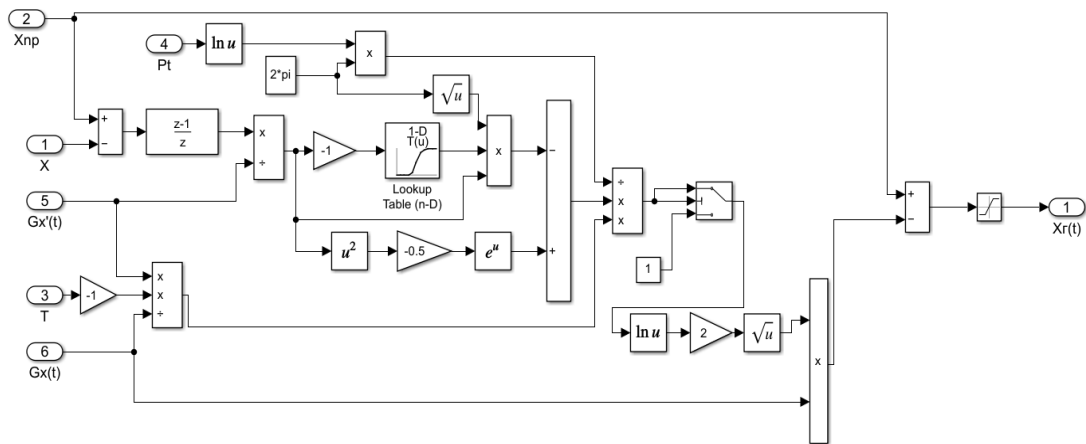


Рис. 5.9. Схема блока BRGZD (БРГЗД – блок розрахунку, гарантуєчого заданого значення)

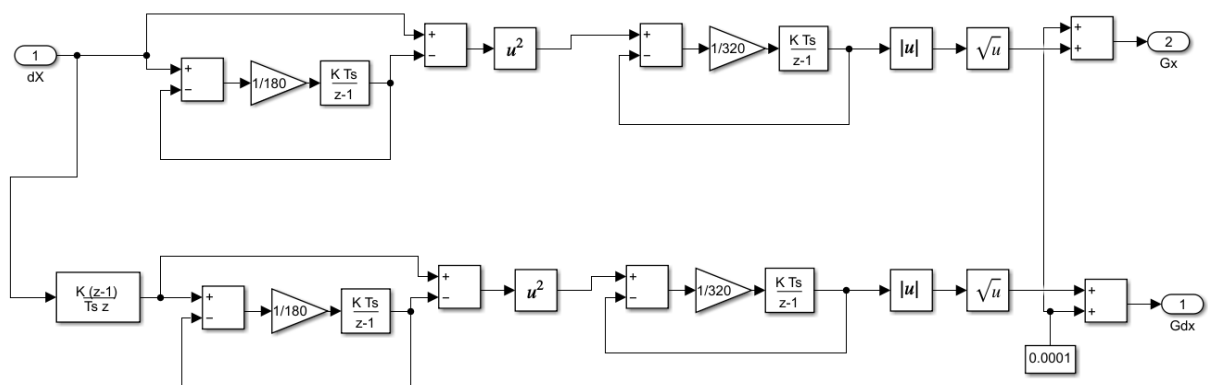


Рис. 5.10. Схема блока BRSO (БРСО – блок розрахунку статистичних оцінок на ковзаючому інтервалі часу).

Отримання z-передатних функцій можна проводити без структурних перетворень ланок системи, а за допомогою вбудованих функцій z-перетворень у системі Matlab. Приклад отримання z-передатної функції наведено на рис. 5.11.

```

New to MATLAB? See resources for Getting Started.

>> csys=tf([2.1 1],[7.6 1])

csys =
    2.1 s + 1
    -----
    7.6 s + 1

Continuous-time transfer function.

>> dsys=c2d(csys,0.1,'foh')

dsys =
    0.2811 z - 0.268
    -----
    z - 0.9869

Sample time: 0.1 seconds
Discrete-time transfer function.

>> csys=tf([1.4 0],[0.2*1.4 1])

csys =
    1.4 s
    -----
    0.28 s + 1

Continuous-time transfer function.

>> dsys=c2d(csys,0.1,'foh')

dsys =
    4.205 z - 4.205
    -----
    z - 0.6997

Sample time: 0.1 seconds
Discrete-time transfer function.

fs >> |
  
```

Рис. 5.11. Приклад отримання z-передатної функції з її неперервного аналогу.

На рис. 5.12 наведена схема моделювання об'єкта керування.

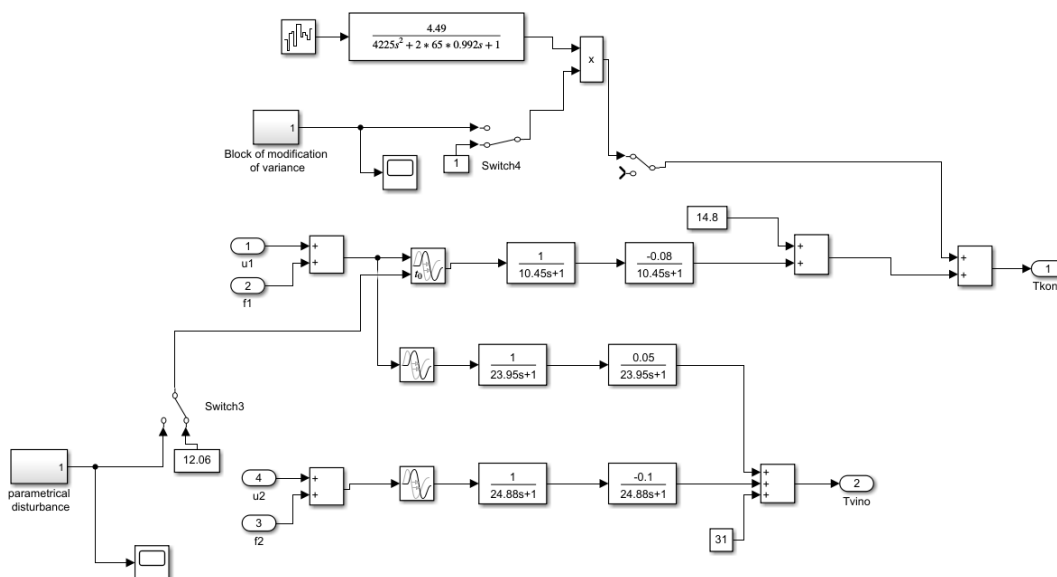


Рис. 5.12. Схема моделювання об'єкта керування.

Результати моделювання системи керування при використанні цифрових алгоритмів які підготовлені до конвертації наведені на рис.5.13 и 5.14.

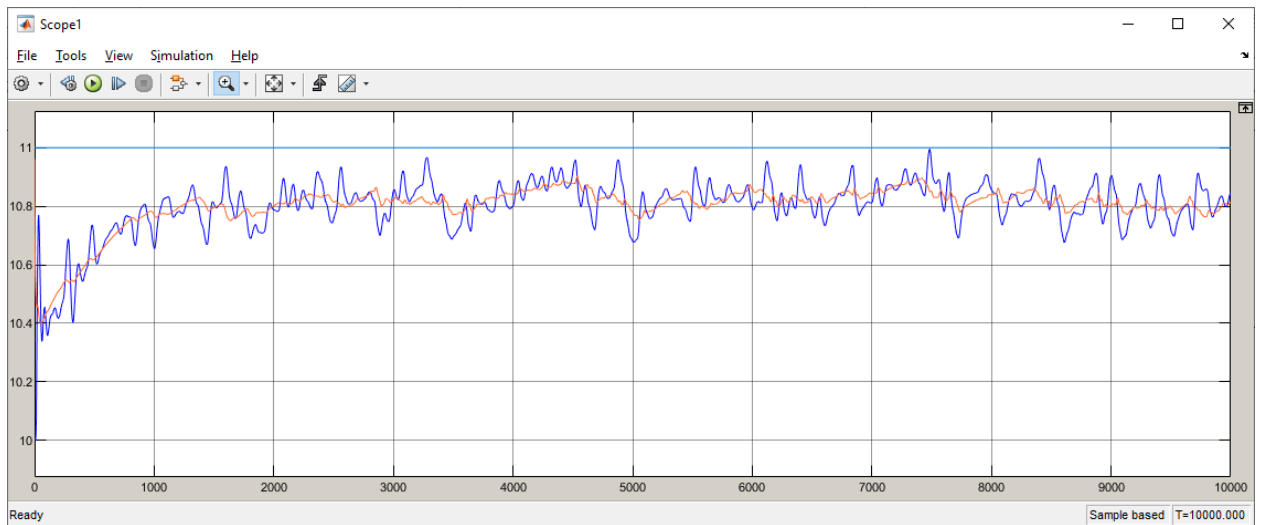


Рис. 5.13. Результати моделювання системи гарантуючого керування за каналом регулювання температури конденсату.

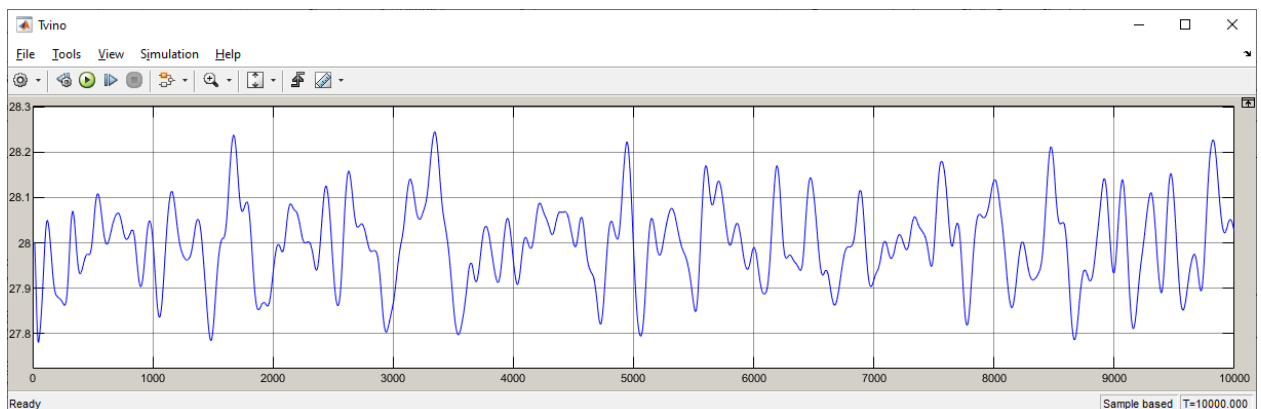


Рис. 5.14. Результати моделювання системи за каналом регулювання температури виноматеріалу на виході з конденсатора.

Для кожної підсистеми яка буде конвертуватися у мову SCL середовища Step 7 проведемо налаштування рис 5.15.

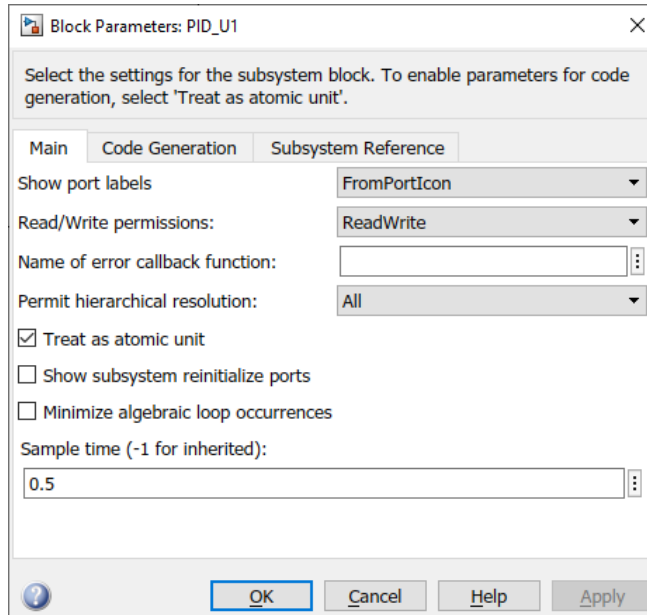


Рис. 5.15. Вікно налаштування підсистем для їх конвертації.

Пакет Simulink PLC Coder налаштуємо для компіляції коду для середовища Siemens SIMATIC Step7 5.4 (рис. 5.16).

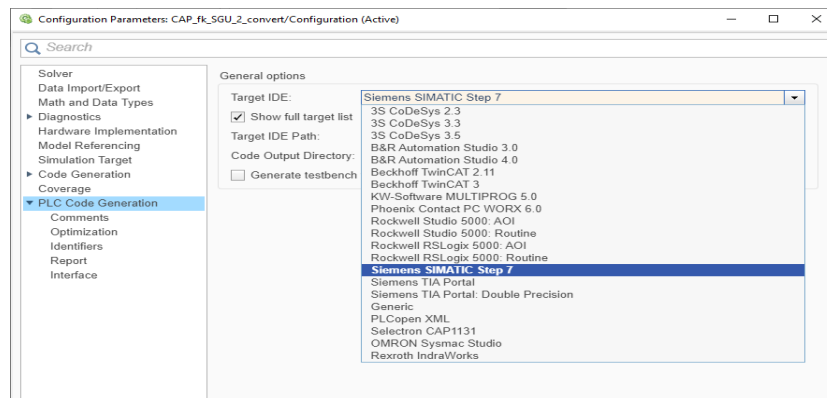


Рис. 5.16. Вікно налаштування пакету SimulinkPLCCoder.

Далі послідовно проведемо генерацію програмного коду для блоку ПІД регулятора та блоків системи гарантуючого керування. На рис. 5.17 та 5.18 проведена процедура генерації коду для ПІД регулятора.

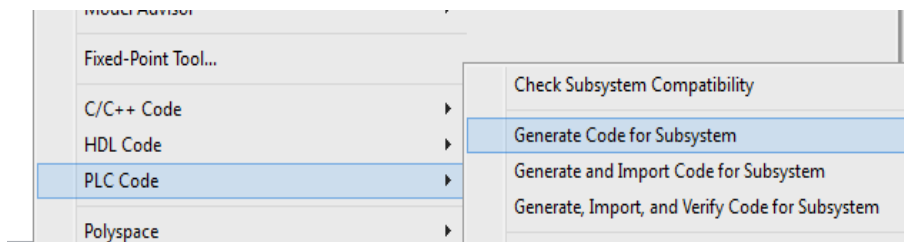


Рис 5.17 – Компіляція коду.

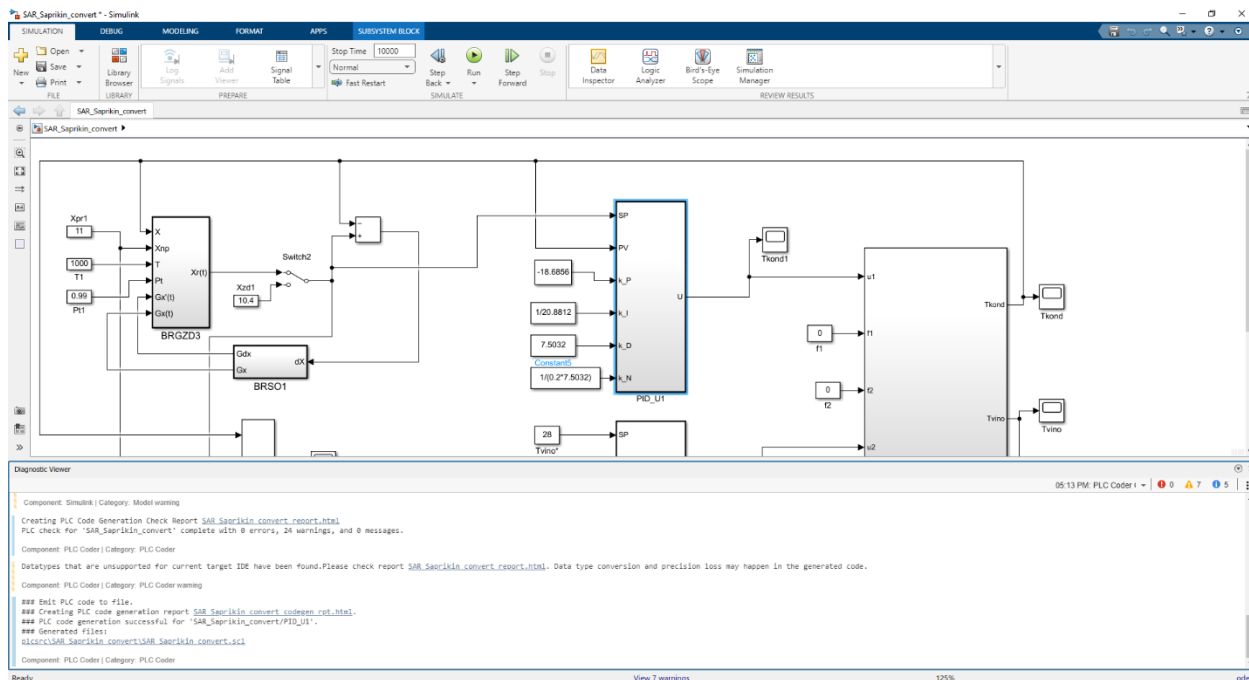


Рис. 5.18 – Результати успішної генерації коду ПІД-регулятора для середовища SIMATIC Step7 5.4.

На рис. 5.21 та 5.22 наведено вихідний генерований код мовою SCL для блоків ПІД регулятора та БРГЗД. Повний список вихідних кодів конвертованих блоків наведено у додатку А.

```

22 FUNCTION_BLOCK PID_U1
23 VAR_INPUT
24     ssMethodType: INT;
25     SP: REAL;
26     PV: REAL;
27     k_P: REAL;
28     k_I: REAL;
29     k_D: REAL;
30     k_N: REAL;
31 END_VAR
32 VAR_OUTPUT
33     U: REAL;
34 END_VAR
35 VAR
36     Integrator_DSTATE: REAL;
37     Filter_DSTATE: REAL;
38     rtb_Add: REAL;
39     rtb_Product: REAL;
40     rtb_Product3: REAL;
41 END_VAR
42 CASE ssMethodType OF
43     0:
44
45         (* InitializeConditions for DiscreteIntegrator: 's1/Integrator' *)
46         Integrator_DSTATE := 60.0;
47
48         (* InitializeConditions for DiscreteIntegrator: 's1/Filter' *)
49         Filter_DSTATE := 0.0;
50     1:
51
52         (* DiscreteIntegrator: 's1/Integrator' *)
53         IF Integrator_DSTATE >= 100.0 THEN
54             Integrator_DSTATE := 100.0;
55         ELSEIF Integrator_DSTATE <= 0.0 THEN
56

```

Рис. 5.21. Приклад генерованого коду ПІД – регулятора для середовища програмування Step7.

```

EDITOR          VIEW
+ Find Files  Insert  fx
New Open Save  Compare Comment  Go To  Breakpoints
Print  Indent  Find
FILE          EDIT          NAVIGATE          BREAKPOINTS

SARGRN_convertscd x +
22 FUNCTION_BLOCK BRGZD2
23 VAR_INPUT
24   ssMethodType: INT;
25   X: REAL;
26   Xnp: REAL;
27   T: REAL;
28   Pt: REAL;
29   Gxt: REAL;
30   Gxt_c: REAL;
31 END_VAR
32 VAR_OUTPUT
33   Xt: REAL;
34 END_VAR
35 VAR
36   UD_DSTATE: REAL;
37   LookupTablenD_tabl: ARRAY [0..1000] OF REAL;
38   LookupTablenD_bp01: ARRAY [0..1000] OF REAL;
39   need_init: BOOL := TRUE;
40   frac: REAL;
41   iRght: DINT;
42   iLeft: DINT;
43   bpIdx: DINT;
44   rtb_Product: REAL;
45   rtb_Sum: REAL;
46   Sqrt1: REAL;
47 END_VAR
48 IF need_init THEN
49   LookupTablenD_tabl[0] := 2.866515718791946E-7;
50   LookupTablenD_tabl[1] := 3.0189646252084903E-7;
51   LookupTablenD_tabl[2] := 3.179213661852819E-7;
52   LookupTablenD_tabl[3] := 3.3476450827361839E-7;
53   LookupTablenD_tabl[4] := 3.5246589817642519E-7;
54   LookupTablenD_tabl[5] := 3.710674079633366E-7;
55   LookupTablenD_tabl[6] := 3.9061285431832642E-7;

```

Рис. 5.22. Приклад генерованого коду БРГЗД для середовища програмування Step7.

Програмну реалізацію алгоритмів керування на базі нейронних мереж проводимо аналогічним чином. На рис. 5.23 наведена схема моделювання нейронної мережі.

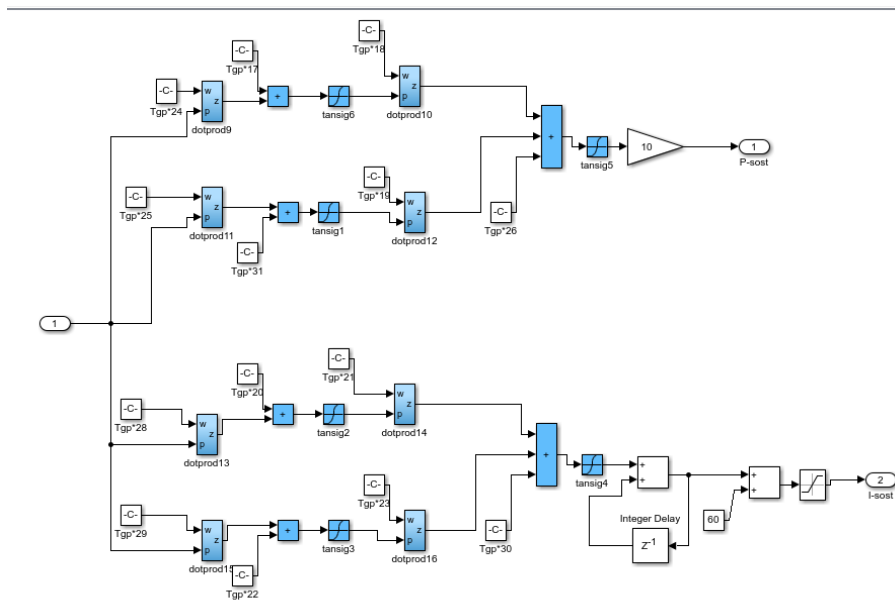


Рис. 5.23. Схема моделювання нейронного регулятора.

Далі послідовно проведемо генерацію програмного коду для середовища SIMATIC Step7 5.4 На рис. 5.24 наведено вихідний генерований код мовою SCL для блоку нейронної мережі. Повний список вихідних кодів конвертованих блоків наведено у додатку А.

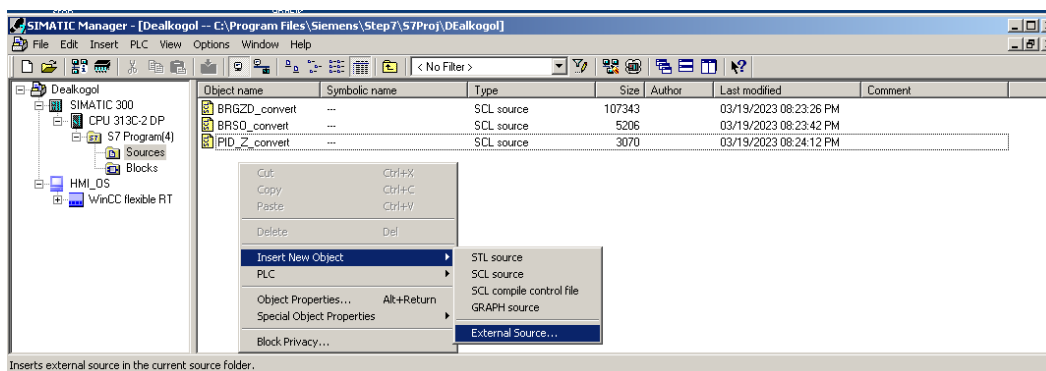
```

11 * Model last modified by      : user
12 * Model last modified on     : Wed Jun 07 17:27:17 2023
13 * Model sample time         : 0.1s
14 * Subsystem name            : SAR_neuro_convert/Neuroregulator
15
16 * Subsystem sample time     : 0.1s|
17 * Simulink PLC Coder version : 3.6 (R2022a) 13-Nov-2021
18 * ST code generated on      : Wed Jun 07 17:28:00 2023
19 *
20 * Target IDE selection       : Siemens SIMATIC Step 7
21 * Test Bench included        : No
22 *
23 *)
24 FUNCTION_BLOCK Neuroregulator
25 VAR_INPUT
26   ssMethodType: INT;
27   In1: REAL;
28 END_VAR
29 VAR_OUTPUT
30   Psoot: REAL;
31   Isoot: REAL;
32 END_VAR
33 VAR
34   IntegerDelay_DSTATE: REAL;
35   rtb_Sum1: REAL;
36   temp1: REAL;
37   temp2: REAL;
38   temp3: REAL;
39   temp4: REAL;
40 END_VAR
41 CASE ssMethodType OF
42   0:
43     (* SystemInitialize for Atomic SubSystem: '<Root>/Neuroregulator ' *)
44     (* InitializeConditions for Delay: '<S1>/Integer Delay' *)
45     IntegerDelay_DSTATE := 0.0;
46     (* End of SystemInitialize for SubSystem: '<Root>/Neuroregulator ' *)
47   1:
48     (* Outputs for Atomic SubSystem: '<Root>/Neuroregulator ' *)
49     (* Sum: '<S1>/Sum1' incorporates:
50      * Constant: '<S11>/one1'
51      * Constant: '<S11>/one1'
52      * Constant: '<S12>/one1'
53      * Constant: '<S12>/one1'
54      * Constant: '<S13>/one1'

```

Рис. 5.24. Вихідний генерований код мовою SCL блоку нейронної мережі для середовища програмування Step7.

Після генерації коду в середовищі MATLAB додаємо новий код у розроблений проект у середовищі Step 7. Процес інтеграції програмних модулів у проект Step 7 наведено на рис. 5.25, а на рис 5.26 приклад імпортованого генерованого коду.



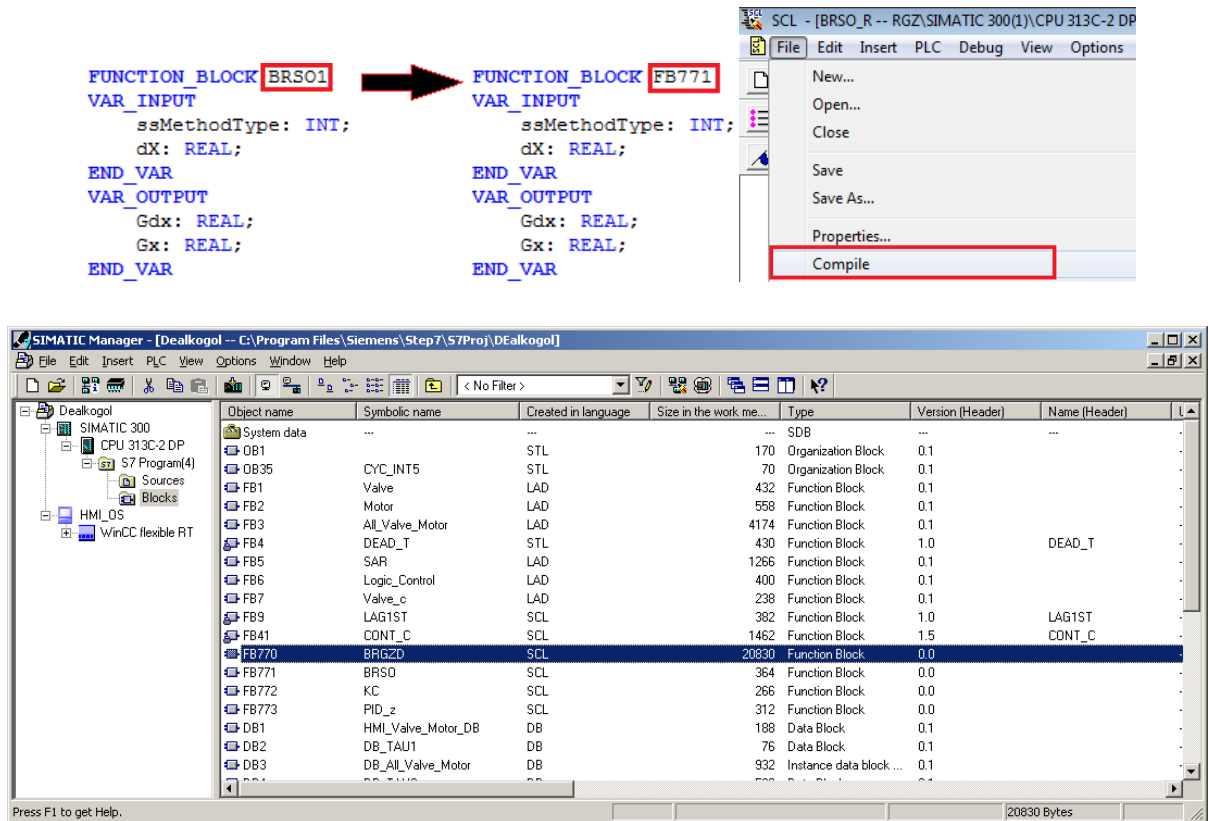


Рис. 5.25. Процедура імпорту генерованого коду в середовище програмування Step 7 і створення відповідного функціонального блоку.

```

FUNCTION_BLOCK FB772
VAR_INPUT
    ssMethodType: INT;
    SP: REAL;
    PV: REAL;
    k_P: REAL;
    k_I: REAL;
    k_D: REAL;
    k_N: REAL;
END_VAR
VAR_OUTPUT
    U: REAL;
    dPV: REAL;
END_VAR
VAR
    Integrator_DSTATE: REAL;
    Filter_DSTATE: REAL;
    rtb_Add: REAL;
    rtb_Add2: REAL;
    rtb_Product: REAL;
    rtb_Product3: REAL;
END_VAR
CASE ssMethodType OF
    0:
        (* InitializeConditions for DiscreteIntegrator: '<S1>/Integrator' *)
        Integrator_DSTATE := 0.0;

        (* InitializeConditions for DiscreteIntegrator: '<S1>/Filter' *)
        Filter_DSTATE := 0.0;
    1:
        (* DiscreteIntegrator: '<S1>/Integrator' *)
        IF Integrator_DSTATE >= 100.0 THEN
            Integrator_DSTATE := 100.0;
        ELSIF Integrator_DSTATE <= -100.0 THEN
            Integrator_DSTATE := -100.0;
        END IF;

```

```

rtb_Add2 := SP - PV;
(* Product: '<S1>/Product' incorporates:
 * Inport: '<Root>/K_P' *)
rtb_Product := rtb_Add2 * K_P;
(* Product: '<S1>/Product3' incorporates:
 * DiscreteIntegrator: '<S1>/Filter'
 * Inport: '<Root>/k_D'
 * Inport: '<Root>/k_N'
 * Product: '<S1>/Product1'
 * Sum: '<S1>/Add1' *)
rtb_Product3 := (rtb_Product * k_D) - Filter_DSTATE) * k_N;
(* Sum: '<S1>/Add' incorporates:
 * DiscreteIntegrator: '<S1>/Integrator' *)
rtb_Add := (Integrator_DSTATE + rtb_Product) + rtb_Product3;
(* Saturate: '<S1>/Saturation' *)
IF rtb_Add >= 100.0 THEN
  (* Outport: '<Root>/U' *)
  U := 100.0;
ELSIF rtb_Add > -100.0 THEN
  (* Outport: '<Root>/U' *)
  U := rtb_Add;
ELSE
  (* Outport: '<Root>/U' *)
  U := -100.0;
END_IF;
(* End of Saturate: '<S1>/Saturation' *)
(* Outport: '<Root>/dPV' *)
dPV := rtb_Add2;
(* Update for DiscreteIntegrator: '<S1>/Integrator' incorporates:
 * Update for Inport: '<Root>/k_I'
 * Product: '<S1>/Product2' *)
Integrator_DSTATE := ((rtb_Product * k_I) * 0.1) + Integrator_DSTATE;
IF Integrator_DSTATE >= 100.0 THEN
  Integrator_DSTATE := 100.0;
ELSIF Integrator_DSTATE <= -100.0 THEN
  Integrator_DSTATE := -100.0;
END_IF;
(* End of Update for DiscreteIntegrator: '<S1>/Integrator' *)

(* Update for DiscreteIntegrator: '<S1>/Filter' *)
Filter_DSTATE := (0.1 * rtb_Product3) + Filter_DSTATE;
END_CASE;
END_FUNCTION_BLOCK

```

Рис. 5.26. Приклад імпортованого коду ПІД-регулятора в середовище програмування Step 7.

Загальний вигляд розробленого проекту в середовищі Step 7 наведено на рис. 5.27.

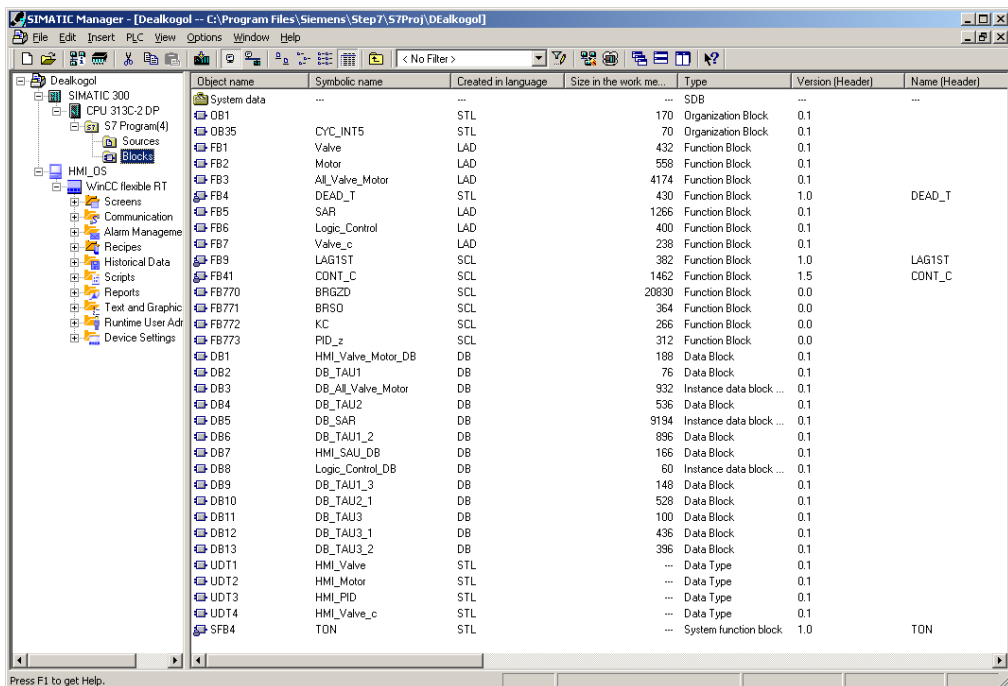


Рис. 5.27. Загальний вид вікна проекту в середовищі Step 7.



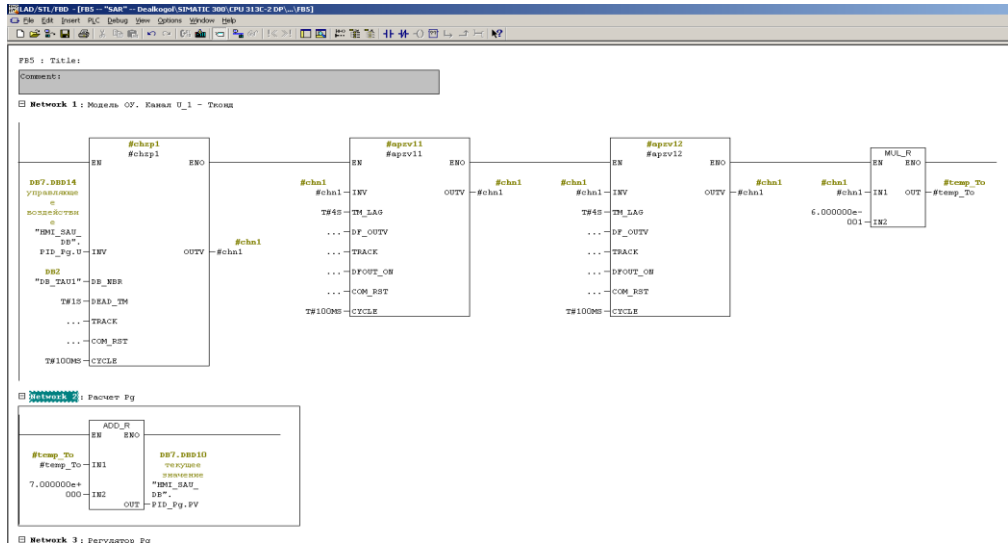


Рис. 5.24. Фрагмент програми яка реалізує розрахунок моделі ОК.

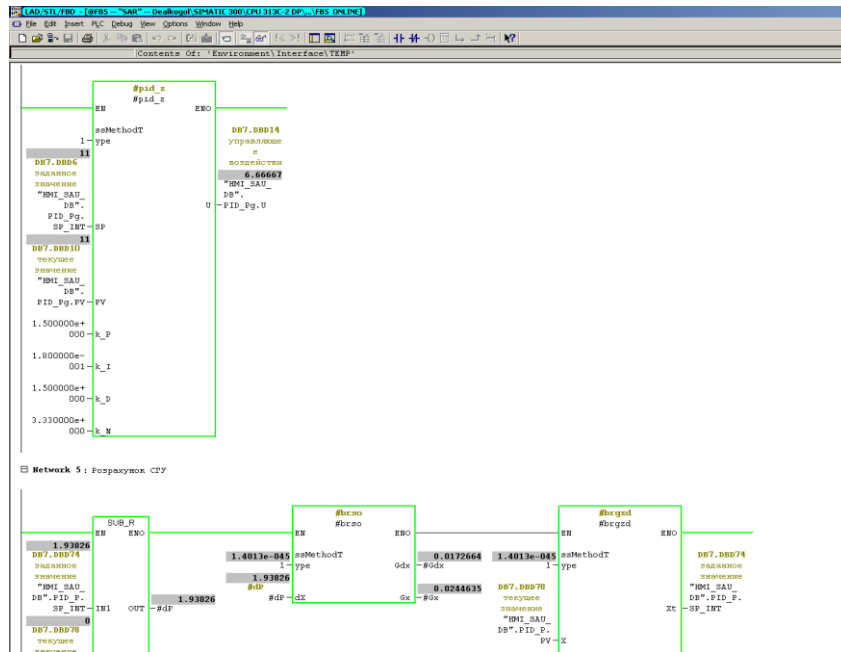


Рис. 5.25. Фрагмент програми конвертованого регулятора температури конденсату у режимі тестування.

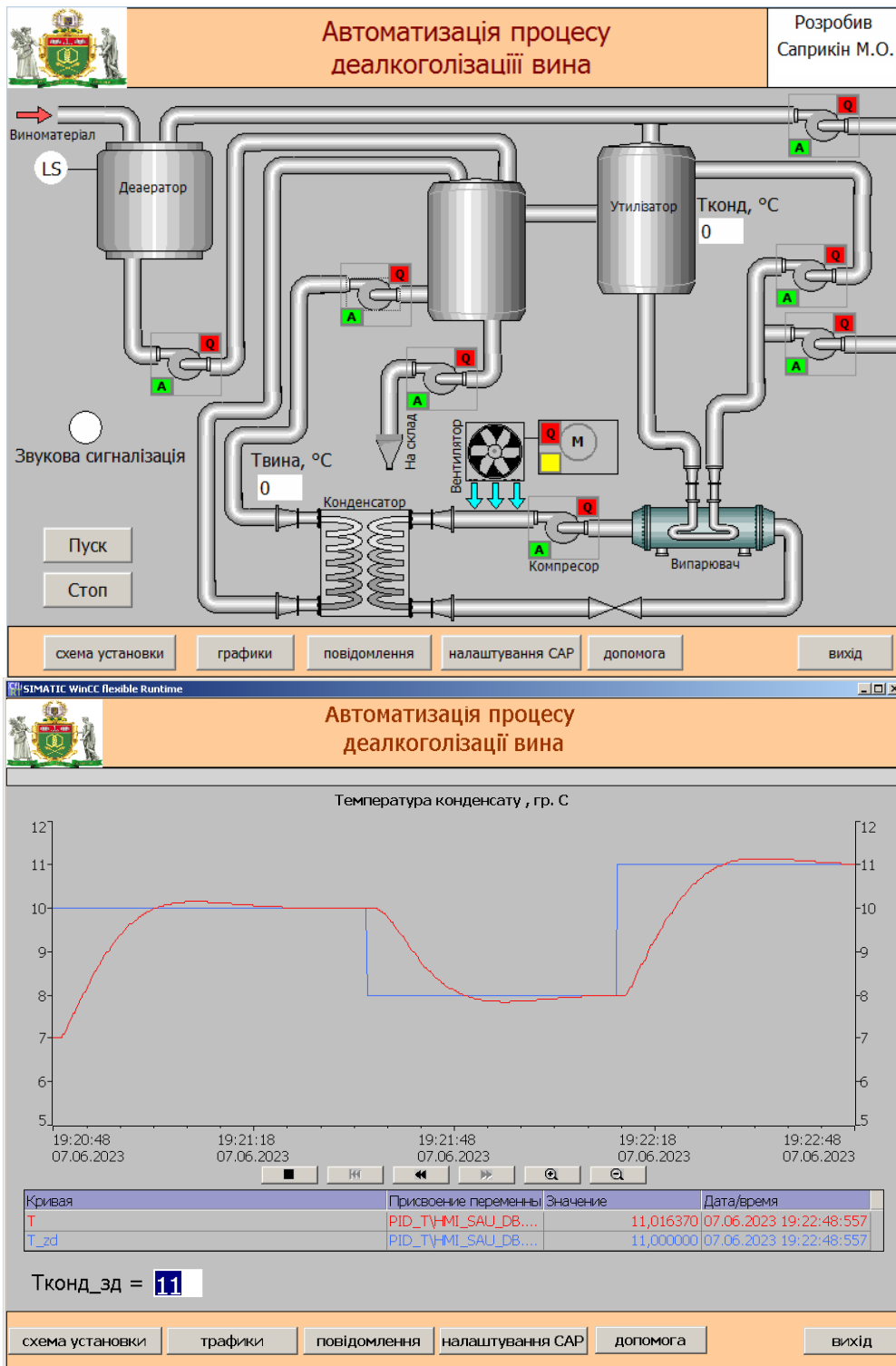


Рис. 5.26. Результати тестування моделі САР температури конденсату.

Результати тестування підтвердили працездатність отриманих в результаті конвертації програмних модулів і ефективність застосування пакету Simulink PLC Coder.

### 5.3. Оцінка економічної ефективності впровадження розробок.

Застосування в системі керування процесом деалкоголізації вина з використанням парокомпресійного теплового насосу алгоритмів гарантуючого керування температурою конденсату дозволить підтримувати температуру конденсату ближче до верхньої межі 11 °С не порушуючи регламенту. Це дозволить забезпечити зниження питомих витрат електричної енергії при деалкоголізації вина до 3 – 5 % .

### 5.4 Висновки за розділом

1. В цьому розділі розглянута програмно-технічна структура системи керування процесом деалкоголізації вина з використанням парокомпресійного теплового насосу. Обґрунтовано обрані контролер, ПЗО та програмне забезпечення фірми Siemens для реалізації системи.

2. Розроблені в системі Matlab моделі нових алгоритмів керування за допомогою пакету Simulink PLC Coder були конвертовані для інтеграції у середовище програмування контролерів фірми Siemens Simatic Step 7. Проведене тестування конвертованого програмного забезпечення підтвердило його працездатність та ефективність застосування конвертору пакета Simulink PLC Coder.

3. Розширення функцій системи керування не потребує зміни технічної структури системи і дозволяє залишити базовий контролер без змін. Тому модернізація САУ на базі Siemens окуповується досить швидко.

## ВИСНОВКИ

Проаналізовано технологічний процес вакуумної деалкоголізації вина в потоці з використанням теплового насоса, як об'єкт керування. На основі аналізу, що визначає якість готового продукту, визначено параметри, які мають найбільший вплив на процес, обґрунтовані вимоги до системи керування та обрані технічні засоби для реалізації системи. Вибраний об'єкт керування – технологічний об'єкт вакуумна деалкоголізація. Рівень цих показників не може бути досягнутий шляхом ручного керування та автоматичного керування з використанням базових алгоритмів. Тому актуальність створення системи автоматичного керування процесом вакуумної деалкоголізації є беззаперечною.

Проведено синтез і аналіз САР системи вакуумної деалкоголізації в якій забезпечується регулювання температури конденсату на вході в утилізатор та температури виноматеріалу на виході з конденсатора, обґрунтовано принципи автоматичного регулювання, проведено аналіз показників якості варіантів розроблених САР.

У результаті синтезу САР підвищеної динамічної точності ми одержали систему, показники якості якої, з більш високою динамічною точністю в порівнянні із САР найпростішої структури, а головне більше грубу. Це означає підвищення працездатності й надійності системи.

Було досліджено, що використання нейрорегулятора доречно, коли об'єкт є суттєво нелінійним і нестационарним, саме за цієї умови якість регулювання вищою, ніж при типових САК.

Використання нечіткої логіки в системах керування дозволяє зменшити втручання оператора в процес керування і, отже, дозволяє розробити нові методики керування, більш адаптовані до промислового середовища.

Також, було розроблено систему, яка окрім традиційних функцій реалізує функцію гарантуючого керування, що є окремим випадком системи автоматичної оптимізації, це дає можливість працювати системі у своєму максимально ефективному режимі, тобто за мінімальних затрат

електроенергії. Було доведено актуальність створення такої системи, раціональність її використання на даному об'єкті.

Була розроблена технічна структура мікропроцесорного ядра системи керування та програмне забезпечення системи автоматичного керування системою вакувної деалкоголізації вина.

Були вивчені основні характеристики контролерів фірми Siemens сімейства S7-30. Отримано практичний досвід програмного конфігурування контролерів в середовищі Step7.

Також було проаналізовано техніко-економічну ефективність від впровадження нової системи автоматичного регулювання.

## Додаток А

Вихідний код конвертованих модулів регулятора та алгоритмів гарантуючого керування.

Вихідний текст модулю БРГЗД.

```
SCL - [BRGZD_R -- RGZSIMATIC 300(1)CPU 313C-2 DP]
FUNCTION_BLOCK FB770
VAR_INPUT
    ssMethodType: INT;
    X: REAL;
    Xnp: REAL;
    T: REAL;
    Pt: REAL;
    Gxt: REAL;
    Gxt_1: REAL;
END_VAR
VAR_OUTPUT
    Xt: REAL;
END_VAR
VAR
    UD_DSTATE: REAL;
    LookupTablenD_tabl: ARRAY [0..1000] OF REAL;
    LookupTablenD_bp01: ARRAY [0..1000] OF REAL;
    need_init: BOOL := TRUE;
    frac: REAL;
    iRight: DINT;
    iLeft: DINT;
    bpIdx: DINT;
    rtb_Product: REAL;
    rtb_Sum: REAL;
    Sqrt1: REAL;
END_VAR
IF need_init THEN
    LookupTablenD_tabl[0] := 2.866515718791946E-7;
    LookupTablenD_tabl[1] := 3.0189646252084903E-7;
    LookupTablenD_tabl[2] := 3.179213661852819E-7;
    LookupTablenD_tabl[3] := 3.3476450827361839E-7;
    LookupTablenD_tabl[4] := 3.5246589817642519E-7;
    LookupTablenD_tabl[5] := 3.710674079633366E-7;
    LookupTablenD_tabl[6] := 3.9061285431832642E-7;
    LookupTablenD_tabl[7] := 4.11148083843931E-7;
    LookupTablenD_tabl[8] := 4.32721061861702E-7;
    LookupTablenD_tabl[9] := 4.5538196484073189E-7;
    LookupTablenD_tabl[10] := 4.7918327659032045E-7;
    LookupTablenD_tabl[11] := 5.041798883575366E-7;
    LookupTablenD_tabl[12] := 5.3042920297509489E-7;
    LookupTablenD_tabl[13] := 5.5799124320978274E-7;
    LookupTablenD_tabl[14] := 5.8692876446663828E-7;
    LookupTablenD_tabl[15] := 6.1730737200919884E-7;
    LookupTablenD_tabl[16] := 6.491956428613363E-7;
    LookupTablenD_tabl[17] := 6.8266525256166436E-7;
    LookupTablenD_tabl[18] := 7.1779110694690021E-7;
    LookupTablenD_tabl[19] := 7.5465147914637377E-7;
    LookupTablenD_tabl[20] := 7.9332815197559734E-7;
    LookupTablenD_tabl[21] := 8.3390656592291234E-7;
    LookupTablenD_tabl[22] := 8.7647597292920517E-7;
    LookupTablenD_tabl[23] := 9.2112959616714359E-7;
    LookupTablenD_tabl[24] := 9.6796479603273579E-7;
    LookupTablenD_tabl[25] := 1.0170832425687061E-6;
    LookupTablenD_tabl[26] := 1.06885910945459359E-6;
    LookupTablenD_tabl[27] := 1.1225991854361736E-6;
    LookupTablenD_tabl[28] := 1.1792232165163983E-6;
    LookupTablenD_tabl[29] := 1.2385839573524714E-6;
    LookupTablenD_tabl[30] := 1.3008074539172811E-6;
    LookupTablenD_tabl[31] := 1.3660252446061406E-6;
    LookupTablenD_tabl[32] := 1.4343745844201367E-6;
    LookupTablenD_tabl[33] := 1.5059986775961571E-6;
    LookupTablenD_tabl[34] := 1.5810469189705114E-6;
    LookupTablenD_tabl[35] := 1.6596751443714621E-6;
    LookupTablenD_tabl[36] := 1.7420458903446622E-6;
    LookupTablenD_tabl[37] := 1.8283286635241654E-6;
    LookupTablenD_tabl[38] := 1.9187002199708997E-6;
    LookupTablenD_tabl[39] := 2.01334485480934E-6;
    LookupTablenD_tabl[40] := 2.1124547025028592E-6;
    LookupTablenD_tabl[41] := 2.2162300481175476E-6;
    LookupTablenD_tabl[42] := 2.3248796499344139E-6;
    LookupTablenD_tabl[43] := 2.4386210737794265E-6;
    LookupTablenD_tabl[44] := 2.5576810394515308E-6;
    LookupTablenD_tabl[45] := 2.6822957796388553E-6;
    LookupTablenD_tabl[46] := 2.8127114117242163E-6;
    LookupTablenD_tabl[47] := 2.9491843228915214E-6;
    LookupTablenD_tabl[48] := 3.091981568956187E-6;
    LookupTablenD_tabl[49] := 3.241381287353944E-6;
    LookupTablenD_tabl[50] := 3.3976731247300615E-6;
    LookupTablenD_tabl[51] := 3.5611586795975568E-6;
    LookupTablenD_tabl[52] := 3.7321519605144834E-6;
    LookupTablenD_tabl[53] := 3.91097986028071E-6;
    LookupTablenD_tabl[54] := 4.0979826466363635E-6;
    LookupTablenD_tabl[55] := 4.2935144699718707E-6;
    LookupTablenD_tabl[56] := 4.4979438885679361E-6;
    LookupTablenD_tabl[57] := 4.7116544118972468E-6;
    LookupTablenD_tabl[58] := 4.9350450625332788E-6;
    LookupTablenD_tabl[59] := 5.1685309572241428E-6;
```

```
SCL - [BRGZD_R -- RGZ/SIMATIC 300(1)CPU 313C-2 DP]
File Edit Insert PLC Debug View Options Window Help
LookupTablenD_tabl[972] := 0.99999882077678348;
LookupTablenD_tabl[973] := 0.99999887740081461;
LookupTablenD_tabl[974] := 0.9999989314089055;
LookupTablenD_tabl[975] := 0.99999898291675748;
LookupTablenD_tabl[976] := 0.999999032035204;
LookupTablenD_tabl[977] := 0.99999907887040385;
LookupTablenD_tabl[978] := 0.99999912352402709;
LookupTablenD_tabl[979] := 0.99999916609343409;
LookupTablenD_tabl[980] := 0.999999206671848;
LookupTablenD_tabl[981] := 0.9999992453485208;
LookupTablenD_tabl[982] := 0.999999282208893;
LookupTablenD_tabl[983] := 0.99999931733474745;
LookupTablenD_tabl[984] := 0.99999935080435709;
LookupTablenD_tabl[985] := 0.999999382692628;
LookupTablenD_tabl[986] := 0.99999941307123552;
LookupTablenD_tabl[987] := 0.99999944200875679;
LookupTablenD_tabl[988] := 0.999999469570797;
LookupTablenD_tabl[989] := 0.99999949582011161;
LookupTablenD_tabl[990] := 0.99999952081672339;
LookupTablenD_tabl[991] := 0.9999995446180352;
LookupTablenD_tabl[992] := 0.99999956727893813;
LookupTablenD_tabl[993] := 0.9999995886519162;
LookupTablenD_tabl[994] := 0.99999960938714572;
LookupTablenD_tabl[995] := 0.99999962893259208;
LookupTablenD_tabl[996] := 0.9999996475340187;
LookupTablenD_tabl[997] := 0.99999966523549177;
LookupTablenD_tabl[998] := 0.99999968207863377;
LookupTablenD_tabl[999] := 0.9999996981035375;
LookupTablenD_tabl[1000] := 0.99999971334842808;
LookupTablenD_bp01[0] := -5.0;
LookupTablenD_bp01[1] := -4.99;
LookupTablenD_bp01[2] := -4.98;
LookupTablenD_bp01[3] := -4.97;
LookupTablenD_bp01[4] := -4.96;
LookupTablenD_bp01[5] := -4.95;
LookupTablenD_bp01[6] := -4.94;
LookupTablenD_bp01[7] := -4.93;
LookupTablenD_bp01[8] := -4.92;
LookupTablenD_bp01[9] := -4.91;
LookupTablenD_bp01[10] := -4.9;
LookupTablenD_bp01[11] := -4.89;
LookupTablenD_bp01[12] := -4.88;
LookupTablenD_bp01[13] := -4.87;
LookupTablenD_bp01[14] := -4.86;

LookupTablenD_bp01[970] := 4.7;
LookupTablenD_bp01[971] := 4.71;
LookupTablenD_bp01[972] := 4.72;
LookupTablenD_bp01[973] := 4.73;
LookupTablenD_bp01[974] := 4.74;
LookupTablenD_bp01[975] := 4.75;
LookupTablenD_bp01[976] := 4.76;
LookupTablenD_bp01[977] := 4.77;
LookupTablenD_bp01[978] := 4.78;
LookupTablenD_bp01[979] := 4.79;
LookupTablenD_bp01[980] := 4.8;
LookupTablenD_bp01[981] := 4.81;
LookupTablenD_bp01[982] := 4.82;
LookupTablenD_bp01[983] := 4.83;
LookupTablenD_bp01[984] := 4.84;
LookupTablenD_bp01[985] := 4.85;
LookupTablenD_bp01[986] := 4.86;
LookupTablenD_bp01[987] := 4.87;
LookupTablenD_bp01[988] := 4.88;
LookupTablenD_bp01[989] := 4.89;
LookupTablenD_bp01[990] := 4.9;
LookupTablenD_bp01[991] := 4.91;
LookupTablenD_bp01[992] := 4.92;
LookupTablenD_bp01[993] := 4.93;
LookupTablenD_bp01[994] := 4.94;
LookupTablenD_bp01[995] := 4.95;
LookupTablenD_bp01[996] := 4.96;
LookupTablenD_bp01[997] := 4.97;
LookupTablenD_bp01[998] := 4.98;
LookupTablenD_bp01[999] := 4.99;
LookupTablenD_bp01[1000] := 5.0;
need_init := FALSE;
END_IF;
CASE asMethodType OF
0:
(* InitializeConditions for UnitDelay: '<S2>/UD'
*
* Block description for '<S2>/UD':
*
* Store in Global RAM *)
UD_DSTATE := 0.0;
1:

```

```

SCL - [BRGZD_R -- RGZ/SIMATIC 300(1)CPU 313C-2 DP]
File Edit Insert PLC Debug View Options Window Help

1:

(* Sum: '<S1>/Sum' incorporates:
 * Inport: '<Root>/X'
 * Inport: '<Root>/Xnp' *)
rtb_Sum := Xnp - X;

(* Product: '<S1>/Product' incorporates:
 * Inport: '<Root>/Gx' (t)
 * Sum: '<S2>/Diff'
 * UnitDelay: '<S2>/UD'
 *
 * Block description for '<S2>/Diff':
 *
 * Add in CPU
 *
 * Block description for '<S2>/UD':
 *
 * Store in Global RAM *)
rtb_Product := (rtb_Sum - UD_DSTATE) / Gxt;

(* Sqrt: '<S1>/Sqrt1' incorporates:
 * Constant: '<S1>/Constant4' *)
Sqrt1 := 2.5066282746310002;

(* Lookup_n-D: '<S1>/Lookup Table (n-D)' incorporates:
 * Gain: '<S1>/Gain3' *)
(* Lookup 1-D
Search method: 'binary'
Use previous index: 'off'
Interpolation method: 'Linear'
Extrapolation method: 'Linear'
Use last breakpoint for index at or above upper limit: 'off'
Remove protection against out-of-range input in generated code: 'off' *)
(* Prelookup - Index and Fraction
Index Search method: 'binary'
Extrapolation method: 'Linear'
Use previous index: 'off'
Use last breakpoint for index at or above upper limit: 'off'
Remove protection against out-of-range input in generated code: 'off' *)
IF (-rtb_Product) <= -5.0 THEN
  iLeft := 0;
  frac := ((-rtb_Product) - -5.0) / 0.0099999999999997868;
ELSIF (-rtb_Product) < 5.0 THEN
  (* Binary Search *)
  bpIdx := 500;
  iLeft := 0;
  iRight := 1000;
  WHILE (iRight - iLeft) > 1 DO
    IF (-rtb_Product) < LookupTablenD_bp01[bpIdx] THEN
      iRight := bpIdx;
    ELSE
      iLeft := bpIdx;
    END IF;
    bpIdx := DWORD_TO_DINT(SHR(IN:=DINT_TO_DWORD(iRight + iLeft), N:=1));
  END WHILE;
  frac := ((-rtb_Product) - LookupTablenD_bp01[iLeft]) / (LookupTablenD_bp01[iLeft + 1] - LookupTablenD_bp01[iLeft]);
ELSE
  iLeft := 999;
  frac := ((-rtb_Product) - 4.99) / 0.0099999999999997868;
END IF;
(* Interpolation 1-D
Interpolation method: 'Linear'
Use last breakpoint for index at or above upper limit: 'off'
Overflow mode: 'wrapping' *)

(* Product: '<S1>/Product3' incorporates:
 * Constant: '<S1>/Constant4'
 * Gain: '<S1>/Gain'
 * Gain: '<S1>/Gain2'
 * Inport: '<Root>/Gx(t)'
 * Inport: '<Root>/Gx' (t)
 * Inport: '<Root>/Pt'
 * Inport: '<Root>/T'
 * Lookup_n-D: '<S1>/Lookup Table (n-D)'
 * Math: '<S1>/Math Function'
 * Math: '<S1>/Math Function2'
 * Math: '<S1>/Math Function3'
 * Product: '<S1>/Product1'
 * Product: '<S1>/Product2'
 * Product: '<S1>/Product4'
 * Sum: '<S1>/Sum1'
 *
 * About '<S1>/Math Function':
 * Operator: log
 *)

```

```

SCL - [BRGZD_R -- RGZSIMATIC 300(1)CPU 313C-2 DP]
File Edit Insert PLC Debug View Options Window Help

* Operator: log
*
* About '<S1>/Math Function2':
* Operator: exp *)
rtb_Product := (((((LookupTablenD_tabl[iLeft + 1] - LookupTablenD_tabl[iLeft]) * frac) + LookupTablenD_tabl[iLeft]) *
  Sqrt1) * rtb_Product) + EXP((rtb_Product * rtb_Product) * -0.5)) * (1.0 / (LN(Pt * 6.2831853071795862))) * (
  (Gxt * (-T)) / Gxt_i);

(* Switch: '<S1>/Switch' incorporates:
* Constant: '<S1>/Constant5' *)
IF NOT (rtb_Product >= 1.0) THEN
  rtb_Product := 1.0;
END_IF;
(* End of Switch: '<S1>/Switch' *)

(* Math: '<S1>/Math Function4'
*
* About '<S1>/Math Function4':
* Operator: log *)
rtb_Product := LN(rtb_Product);

(* Gain: '<S1>/Gain1' *)
rtb_Product := 2.0 * rtb_Product;

(* Sqrt: '<S1>/Sqrt' *)
rtb_Product := Sqrt(rtb_Product);

(* Sum: '<S1>/Sum2' incorporates:
* Input: '<Root>/Gx(t)'
* Input: '<Root>/Xnp'
* Product: '<S1>/Product5' *)
rtb_Product := (rtb_Product * Gxt_i) + Xnp;

(* Saturate: '<S1>/Saturation2' *)
IF rtb_Product >= 120.0 THEN
  (* Output: '<Root>/Xp(t)' *)
  Xt := 120.0;
ELSIF rtb_Product > -20.0 THEN
  (* Output: '<Root>/Xp(t)' *)
  Xt := rtb_Product;
ELSE
  (* Output: '<Root>/Xp(t)' *)
  Xt := -20.0;
END_IF;

(* End of Saturate: '<S1>/Saturation2' *)

(* Update for UnitDelay: '<S2>/UD'
*
* Block description for '<S2>/UD':
*
* Store in Global RAM *)
UD_DSTATE := rtb_Sum;
END_CASE;
END_FUNCTION_BLOCK

```

## Вихідний текст модулю BRCO.

```

SCL - [BRSO_R -- RGZSIMATIC 300(1)CPU 313C-2 DP]
File Edit Insert PLC Debug View Options Window Help

FUNCTION_BLOCK FB771
VAR_INPUT
  ssMethodType: INT;
  dX: REAL;
END_VAR
VAR_OUTPUT
  Gdx: REAL;
  Gx: REAL;
END_VAR
VAR
  c_DiscreteTimeIntegr: REAL;
  d_DiscreteTimeIntegr: REAL;
  e_DiscreteTimeIntegr: REAL;
  f_DiscreteTimeIntegr: REAL;
  UD_DSTATE: REAL;
  rtb_Sum: REAL;
  rtb_TSamp: REAL;
  rtb_Sum4: REAL;
  rtb_Sum1: REAL;
END_VAR
CASE ssMethodType OF
  0:
    (* InitializeConditions for DiscreteIntegrator: '<S1>/Discrete-Time Integrator2' *)
    c_DiscreteTimeIntegr := 0.010000000000000002;
    (* InitializeConditions for DiscreteIntegrator: '<S1>/Discrete-Time Integrator3' *)
    d_DiscreteTimeIntegr := 0.0006250000000000012;
    (* InitializeConditions for DiscreteIntegrator: '<S1>/Discrete-Time Integrator' *)
    e_DiscreteTimeIntegr := 0.0;
    (* InitializeConditions for DiscreteIntegrator: '<S1>/Discrete-Time Integrator1' *)
    f_DiscreteTimeIntegr := 0.0;
    (* InitializeConditions for UnitDelay: '<S2>/UD'
    *
    * Block description for '<S2>/UD':
    *
    * Store in Global RAM *)
    UD_DSTATE := 0.0;
  1:
    (* SampleTimeMath: '<S2>/TSamp' incorporates:

```

```

SCL - [BRSO_R -- RGZASIMATIC 300(1)CPU 313C-2 DP]
File Edit Insert PLC Debug View Options Window Help
* About '<S2>/TSamp':
* y = u * K where K = 1 / ( w * Ts ) *)
rtb_TSamp := dX * 4.0;

(* Sum: '<S2>/Diff' incorporates:
* UnitDelay: '<S2>/UD'
*
* Block description for '<S2>/Diff':
*
* Add in CPU
*
* Block description for '<S2>/UD':
*
* Store in Global RAM *)
rtb_Sum1 := rtb_TSamp - UD_DSTATE;

(* Sum: '<S1>/Sum4' incorporates:
* DiscreteIntegrator: '<S1>/Discrete-Time Integrator' *)
rtb_Sum4 := rtb_Sum1 - e_DiscreteTimeIntegr;

(* Sum: '<S1>/Sum' incorporates:
* DiscreteIntegrator: '<S1>/Discrete-Time Integrator1'
* Inport: '<Root>/dX' *)
rtb_Sum := dX - f_DiscreteTimeIntegr;

(* Sum: '<S1>/Sum1' incorporates:
* DiscreteIntegrator: '<S1>/Discrete-Time Integrator' *)
rtb_Sum1 := rtb_Sum1 - e_DiscreteTimeIntegr;

(* Output: '<Root>/Gdx' incorporates:
* Abs: '<S1>/Abs2'
* Constant: '<S1>/Constant'
* DiscreteIntegrator: '<S1>/Discrete-Time Integrator3'
* Sqrt: '<S1>/Sqrt'
* Sum: '<S1>/Sum3' *)
Gdx := Sqrt(Abs(d_DiscreteTimeIntegr)) + 0.01;

(* Output: '<Root>/Gx' incorporates:
* Abs: '<S1>/Abs1'
* Constant: '<S1>/Constant'
* DiscreteIntegrator: '<S1>/Discrete-Time Integrator2'
* Sqrt: '<S1>/Sqrt1'
* Sum: '<S1>/Sum2' *)
Gx := 0.01 + Sqrt(Abs(c_DiscreteTimeIntegr));

(* Output: '<Root>/Gx' incorporates:
* Abs: '<S1>/Abs1'
* Constant: '<S1>/Constant'
* DiscreteIntegrator: '<S1>/Discrete-Time Integrator2'
* Sqrt: '<S1>/Sqrt1'
* Sum: '<S1>/Sum2' *)
Gx := 0.01 + Sqrt(Abs(c_DiscreteTimeIntegr));

(* Update for DiscreteIntegrator: '<S1>/Discrete-Time Integrator2' incorporates:
* DiscreteIntegrator: '<S1>/Discrete-Time Integrator2'
* Gain: '<S1>/Gain2'
* Math: '<S1>/Math Function'
* Sum: '<S1>/Sum6' *)
c_DiscreteTimeIntegr := (((rtb_Sum * rtb_Sum) - c_DiscreteTimeIntegr) * 0.02) * 0.25 + c_DiscreteTimeIntegr;

(* Update for DiscreteIntegrator: '<S1>/Discrete-Time Integrator3' incorporates:
* DiscreteIntegrator: '<S1>/Discrete-Time Integrator3'
* Gain: '<S1>/Gain3'
* Math: '<S1>/Math Function1'
* Sum: '<S1>/Sum7' *)
d_DiscreteTimeIntegr := (((rtb_Sum1 * rtb_Sum1) - d_DiscreteTimeIntegr) * 0.02) * 0.25 + d_DiscreteTimeIntegr;

(* Update for DiscreteIntegrator: '<S1>/Discrete-Time Integrator' incorporates:
* Gain: '<S1>/Gain' *)
e_DiscreteTimeIntegr := ((0.2 * rtb_Sum4) * 0.25) + e_DiscreteTimeIntegr;

(* Update for DiscreteIntegrator: '<S1>/Discrete-Time Integrator1' incorporates:
* DiscreteIntegrator: '<S1>/Discrete-Time Integrator1'
* Gain: '<S1>/Gain1'
* Inport: '<Root>/dX'
* Sum: '<S1>/Sum5' *)
f_DiscreteTimeIntegr := (((dX - f_DiscreteTimeIntegr) * 0.2) * 0.25) + f_DiscreteTimeIntegr;

(* Update for UnitDelay: '<S2>/UD'
*
* Block description for '<S2>/UD':
*
* Store in Global RAM *)
UD_DSTATE := rtb_TSamp;
END_CASE;
END_FUNCTION_BLOCK

```

## Вихідний текст модулю ПД регулятора..

```
SCL - [PID_R - RGZASIMATIC 300(1)ICPU 313C-2 DP]
File Edit Insert PLC Debug View Options Window Help
FUNCTION_BLOCK FB772
VAR_INPUT
    ssMethodType: INT;
    SP: REAL;
    PV: REAL;
    k_P: REAL;
    k_I: REAL;
    k_D: REAL;
    k_N: REAL;
END_VAR
VAR_OUTPUT
    U: REAL;
    dPV: REAL;
END_VAR
VAR
    Integrator_DSTATE: REAL;
    Filter_DSTATE: REAL;
    rtb_Add: REAL;
    rtb_Add2: REAL;
    rtb_Product: REAL;
    rtb_Product3: REAL;
END_VAR
CASE ssMethodType OF
    0:
        (* InitializeConditions for DiscreteIntegrator: '<S1>/Integrator' *)
        Integrator_DSTATE := 0.0;

        (* InitializeConditions for DiscreteIntegrator: '<S1>/Filter' *)
        Filter_DSTATE := 0.0;
    1:
        (* DiscreteIntegrator: '<S1>/Integrator' *)
        IF Integrator_DSTATE >= 100.0 THEN
            Integrator_DSTATE := 100.0;
        ELSIF Integrator_DSTATE <= -100.0 THEN
            Integrator_DSTATE := -100.0;
        END_IF;

        (* Sum: '<S1>/Add2' incorporates:
         * Import: '<Root>/PV'
         * Import: '<Root>/SP' *)
        rtb_Add2 := SP - PV;

        (* Product: '<S1>/Product' incorporates:
         * Import: '<Root>/k_P' *)
        rtb_Product := rtb_Add2 * k_P;
        (* Product: '<S1>/Product3' incorporates:
         * DiscreteIntegrator: '<S1>/Filter'
         * Import: '<Root>/k_D'
         * Import: '<Root>/k_N'
         * Product: '<S1>/Product1'
         * Sum: '<S1>/Add1' *)
        rtb_Product3 := ((rtb_Product * k_D) - Filter_DSTATE) * k_N;
        (* Sum: '<S1>/Add' incorporates:
         * DiscreteIntegrator: '<S1>/Integrator' *)
        rtb_Add := (Integrator_DSTATE + rtb_Product) + rtb_Product3;
        (* Saturate: '<S1>/Saturation' *)
        IF rtb_Add >= 100.0 THEN
            (* Output: '<Root>/U' *)
            U := 100.0;
        ELSIF rtb_Add > -100.0 THEN
            (* Output: '<Root>/U' *)
            U := rtb_Add;
        ELSE
            (* Output: '<Root>/U' *)
            U := -100.0;
        END_IF;
        (* End of Saturate: '<S1>/Saturation' *)
        (* Output: '<Root>/dPV' *)
        dPV := rtb_Add2;
        (* Update for DiscreteIntegrator: '<S1>/Integrator' incorporates:
         * Update for Import: '<Root>/k_I'
         * Product: '<S1>/Product2' *)
        Integrator_DSTATE := ((rtb_Product * k_I) * 0.1) + Integrator_DSTATE;
        IF Integrator_DSTATE >= 100.0 THEN
            Integrator_DSTATE := 100.0;
        ELSIF Integrator_DSTATE <= -100.0 THEN
            Integrator_DSTATE := -100.0;
        END_IF;
        (* End of Update for DiscreteIntegrator: '<S1>/Integrator' *)
        (* Update for DiscreteIntegrator: '<S1>/Filter' *)
        Filter_DSTATE := (0.1 * rtb_Product3) + Filter_DSTATE;
END_CASE;
END_FUNCTION_BLOCK
```

# Вихідний текст модулю нейронного регулятора.

```
SCL - [SAR_neuro_convert -- Dealkogol/SIMATIC 300/CPU 313C-2 DP]
File Edit Insert PLC Debug View Options Window Help
[*]
*
* File: SAR_neuro_convert.scl
*
* IEC 61131-3 Structured Text (ST) code generated for subsystem "SAR_neuro_convert/Neuroregulator"
*
*
* Model name : SAR_neuro_convert
* Model version : 1.1
* Model creator : user
* Model last modified by : user
* Model last modified on : Wed Jun 07 17:27:17 2023
* Model sample time : 0.1s
* Subsystem name : SAR_neuro_convert/Neuroregulator
*
* Subsystem sample time : 0.1s
* Simulink PLC Coder version : 3.6 (R2022a) 13-Nov-2021
* ST code generated on : Wed Jun 07 17:28:00 2023
*
* Target IDE selection : Siemens SIMATIC Step 7
* Test Bench included : No
*
*)
FUNCTION_BLOCK Neuroregulator
VAR_INPUT
ssMethodType: INT;
In1: REAL;
END_VAR
VAR_OUTPUT
Psoist: REAL;
Isoist: REAL;
END_VAR
VAR
IntegerDelay_DSTATE: REAL;
rtb_Sum11: REAL;
temp1: REAL;
temp2: REAL;
temp3: REAL;
temp4: REAL;
END_VAR
CASE ssMethodType OF
0:
(* SystemInitialize for Atomic SubSystem: '<Root>/Neuroregulator' *)
(* InitializeConditions for Delay: '<S1>/Integer Delay' *)
IntegerDelay_DSTATE := 0.0;
(* End of SystemInitialize for SubSystem: '<Root>/Neuroregulator' *)
1:
(* Outputs for Atomic SubSystem: '<Root>/Neuroregulator' *)
(* Sum: '<S1>/Sum11' incorporates:
* Constant: '<S11>/one'
* Constant: '<S12>/one1'
* Constant: '<S12>/one1'
* Constant: '<S13>/one'
* Constant: '<S13>/one1'
* Constant: '<S1>/Tgp*20'
* Constant: '<S1>/Tgp*21'
* Constant: '<S1>/Tgp*22'

```

```
SCL - [SAR_neuro_convert -- Dealkogol/SIMATIC 300/CPU 313C-2 DP]
File Edit Insert PLC Debug View Options Window Help
Sum: '<S1>/netsumo'
*
* About '<S11>/Math Function':
* Operator: exp
*
* About '<S11>/Math Function':
* Operator: reciprocal
*
* About '<S12>/Math Function':
* Operator: exp
*
* About '<S12>/Math Function':
* Operator: reciprocal
*
* About '<S13>/Math Function':
* Operator: exp
*
* About '<S13>/Math Function':
* Operator: reciprocal *)
temp1 := EXP(((0.035204 * In1) + -1.1792957133831861) * -2.0);
temp2 := EXP((-0.68709 * In1) + -1.4280608043713381) * -2.0);
rtb_Sum11 := (((1.0 / EXP((((1.0 / (temp1 + 1.0)) * 2.0) - 1.0) * -1.4533740908584305) + (((1.0 / (temp2 + 1.0)
) * 2.0) - 1.0) * 0.0017308618413412904)) + -1.2007733579053821) * -2.0) + 1.0) * 2.0) - 1.0) + IntegerDelay_DSTATE;
(* Sum: '<S1>/Sum12' incorporates:
* Constant: '<S1>/Constant' *)
Isoist := rtb_Sum11 + 60.0;
(* Saturate: '<S1>/Saturation4' incorporates:
* Constant: '<S1>/Constant'
* Sum: '<S1>/Sum12' *)
IF (rtb_Sum11 + 60.0) > 100.0 THEN
(* Output: '<Root>/I-soist' *)
Isoist := 100.0;
ELSEIF (rtb_Sum11 + 60.0) < 0.0 THEN
(* Output: '<Root>/I-soist' *)
Isoist := 0.0;
END_IF;
(* End of Saturate: '<S1>/Saturation4' *)
(* Update for Delay: '<S1>/Integer Delay' *)
IntegerDelay_DSTATE := rtb_Sum11;
(* Output: '<Root>/P-soist' incorporates:
* Constant: '<S10>/one'
* Constant: '<S10>/one1'
* Constant: '<S14>/one'
* Constant: '<S14>/one1'
* Constant: '<S15>/one'
* Constant: '<S15>/one1'
* Constant: '<S1>/Tgp*17'
* Constant: '<S1>/Tgp*18'
* Constant: '<S1>/Tgp*19'
* Constant: '<S1>/Tgp*24'
* Constant: '<S1>/Tgp*25'
* Constant: '<S1>/Tgp*26'
* Constant: '<S1>/Tgp*31'
* Gain: '<S10>/Gain'
* Gain: '<S10>/Gain1'
* Gain: '<S14>/Gain'
* Gain: '<S14>/Gain1'

```



## СПИСОК ЛІТЕРАТУРИ

1. Хобін В. А. Конспект курсу лекцій по дисципліні «Идентификация и моделирование объектов Автоматизации». Одеса-2008 р.
2. Хобін В. А. Методичні вказівки до виконання курсової роботи з курсу «Идентификация и моделирование технологических объектов», Одеса-2000 г.
3. Киричков В.Н. Идентификация объектов систем управления технологическими процессами. - К.: Высшая шк., 1990. - 263 с.
4. Конспект курса лекций по дисциплине “Теория автоматического управления” (часть I)– Хобин В.А. ОНАПТ. Одесса – 2012 г.
5. Конспект курса лекций по дисциплине “Теория автоматического управления” (часть II)– Хобин В.А. ОНАПТ. Одесса – 2012 г.
6. Методические указания к выполнению и оформлению курсовой работы по дисциплине «Теория автоматического управления» для студентов специальностей 2105 дневной и заочной форм обучения / Сост. В.А. Хобин, - Одесса: ОТИПП, 1992 г. – 15 с.
7. Конспект лекцій з курсу „Автоматизація технологічних процесів та виробництв” для студентів напряму підготовки бакалаврів 6.050202 „Автоматизація та комп’ютерно-інтегровані технології” всіх форм навчання / Укл. Л.М.Сичук, І.М.Світий. – Одеса: ОНАХТ, 2013. – 88 с.
8. User's Manual .WinCC flexible 2008 Compact / Standard / Advanced.
9. «Интеллектуальные технологии управления. Искусственные нейронные сети и нечеткая логика». – Усков А.А., Кузьмин А.В. 2004. – 143 с.
10. «MATLAB 6.5 SP1/7/7 SP1/7 SP2 + Simulink 5/6. Инструменты искусственного интеллекта и биоинформатики». – Дьяконов В.П., Круглов В.В М.: СОЛОН-ПРЕСС, 2006. – 456 с.
11. Методические указания к лабораторной работе «Моделирование нейронных сетей в matlab». – Федотов А.В., 2002. – 22с.

12. Конспект курса лекций по дисциплине «Специальные разделы теории управления» для студентов специальности 8.092501. – Хобин В.А- Одеса: ОНАХТ 2010. – 62 стр.

13. Каталог ИК PI «Промышленная связь для систем автоматизации и приводов». – Siemens AG, 2005. – 666с.

14. Каталог ST70 «SIMATIC – Компоненты для комплексной автоматизации». – Siemens AG, 2007. – 862с.

15. «Автоматизация с помощью программ Step 7 LAD и FBD». – Ганс Бергер, 2001. – 605с.

16. «Економіка підприємства: Підручник» / За заг. редакторка С.Ф. Покропивного-Вид. 2-ге, перероб. але доп. – К.: Київ, 2000. – 528 с.