

**Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»**

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра інформаційних технологій та комп'ютерної інженерії

(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

**кваліфікаційної роботи ступеня магістра**

(бакалавра, спеціаліста, магістра)

**студентта** Саврицький Олександр Ігорович

(ПІБ)

**академічної групи** 123М-19-1

(шифр)

**спеціальності** 123 «Комп'ютерна інженерія»

(код і назва спеціальності)

**за освітньо-професійною програмою** «Комп'ютерна інженерія»

(офіційна назва)

**на тему «Обґрунтування параметрів комп'ютерної системи наземної перекачної станції ім. Сташкова»**

(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	проф. Коротенко Г.М.			
розділів:				
теоретичний розділ	проф. Цвіркун Л.І			
синтез системи	доц. Ткаченко С.М			
розроблення програмного забезпечення	ас. Бешта Л.В			
експериментальний розділ	доц. Ткаченко С.М.			
<b>Рецензент</b>				
<b>Нормоконтролер</b>	проф. Цвіркун Л.І.			

**Дніпро  
2020**

**ЗАТВЕРДЖЕНО:**  
завідувач кафедри  
Інформаційних технологій та  
комп'ютерної інженерії  
(повна назва)

\_\_\_\_\_

\_\_\_\_\_

« \_\_\_\_ » \_\_\_\_\_ грудня 2020 року

**ЗАВДАННЯ**  
**на кваліфікаційну роботу**  
**ступеня \_\_\_\_\_ магістр \_\_\_\_\_**  
(бакалавра, спеціаліста, магістра)

студенту Саврицький О.І. академічної групи 123М-19-1  
(прізвище та ініціали) (шифр)

спеціальності 123 «Комп'ютерна інженерія»

за освітньо-професійною програмою 123 «Комп'ютерна інженерія»  
(офіційна назва)

на тему «Обґрунтування параметрів комп'ютерної системи наземної перекачної станції ім. Сташкова»,

затверджену наказом ректора НТУ «Дніпровська політехніка» від 22.10.2020 р. № 888-с

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел формулювання завдання, конкретизування предмету та мети роботи	01.11.2020
Синтез системи	Розробка комп'ютерної системи	01.11.2020
Програмний розділ	Розробка програмного забезпечення	01.11.2020
Експериментальний	Проведення і обробка результатів експериментів	01.12.2020

Завдання видано \_\_\_\_\_  
(підпис керівника)

проф. Коротенко Г.М.  
(прізвище, ініціали)

Дата видачі 01 жовтня 2020 р.

Дата подання до екзаменаційної комісії \_\_\_\_\_

10.12.2020 р.

Прийнято до виконання \_\_\_\_\_  
(підпис студента)

Саврицький О.І.  
(прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: \_\_\_\_\_ с., \_\_\_\_\_ рис., \_\_\_\_\_ табл., \_\_\_\_\_ дод., \_\_\_\_\_ джерел.

Об'єкт дослідження: комп'ютерна мережа вугільної шахти ім. Сташкова, що лежить в основі забезпечення роботи менеджменту, фінансових служб, транспорту, безпеки функціонування організаційній структурі та системи управління насосною станцією для відкачування шахтних вод.

Мета роботи: удосконалення комп'ютерної мережі підприємства та системи управління насосною станцією відкачування шахтних вод таким чином, щоб забезпечити мінімальні енерговитрати з дотриманням вимог протікання технологічного процесу.

В роботі вирішені наступні завдання. Визначити вузли мережі, на які найсильніше впливає інформаційне перевантаження мережі. Визначити умови, та параметри властивостей технічних пристроїв мережі за яких може виникнути втрата продуктивності. Розробити рекомендації щодо модернізації мережі для підвищенні її стійкості до перевантажень. Визначити найбільш «слабкі» вузли комп'ютерної мережі, які у випадку перевантаження приводять до втрати працездатності мережі і розробити рекомендацій для вирішення такої проблеми.

В тому числі, синтезована система управління станцією на базі безперервного пропорційного регулятора. При її реалізації використані спеціально розроблені моделі безперервного і релейного управління насосними агрегатами.

Практичні результати: визначені недоліки і шляхи удосконалення комп'ютерної мережі. Застосовані наукові розробки з моделювання комп'ютерних мереж системи управління показали свою достовірність у багатьох прикладах їх застосувань.

СИСТЕМА, КОМП'ЮТЕРНА МЕРЕЖА, НАЛАШТУВАННЯ, МАТЕМАТИЧНА МОДЕЛЬ МЕРЕЖІ, ТЕОРІЯ МАСОВОГО ОБСЛУГОВУВАННЯ, УПРАВЛІННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

## ABSTRACT

Explanatory note: \_\_\_\_ pp., \_\_\_\_ figures, \_\_\_\_ tables, \_\_\_\_ annexes, \_\_\_\_ sources.

Object of research: computer network of the coal mine named after Stashkova, which is the basis for ensuring the work of management, financial services, transport, safety of the organizational structure and control system of the pumping station for pumping mine water..

Objective: To improve the computer network of the enterprise and the control system of the pumping station for pumping mine water in such a way as to ensure minimum energy consumption in compliance with the requirements of the technological process.

Identify the network nodes that are most affected by network information congestion. Determine the conditions and parameters of the properties of technical devices of the network under which there may be a loss of performance. Develop recommendations for network modernization to increase its resistance to congestion. Identify the "weakest" nodes of the computer network, which in case of congestion lead to loss of network performance, and develop recommendations for solving this problem.

Synthesize the station control system on the basis of a continuous proportional controller. At its realization specially developed models of continuous and relay control of pump units are used.

Practical results: the identification of shortcomings and ways to improve the computer network is achieved through the use of a scientific approach to solving problems. Applied scientific developments in the modeling of computer networks of the control system have shown their reliability in many examples of their applications.

SYSTEM, COMPUTER NETWORK, SETTINGS, MATHEMATICAL MODEL OF THE NETWORK, THEORY OF CUSTOM SERVICE, MANAGEMENT, SOFTWARE.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	9
Вступ.....	10
1 Стан питання і постановка завдання дослідження .....	13
1.1 Загальна характеристика об'єкта управління.....	13
1.1.1 Характеристика і організаційна структура виробництва.....	13
1.1.2 Аналіз об'єкта впровадження .....	14
1.1.2.1 Комп'ютерна мережа підприємства .....	14
1.1.2.2 Система управління водовідливом.....	17
1.1.3 Принцип функціонування об'єкту управління .....	24
1.1.3.1 Комп'ютерна мережа підприємства .....	24
1.1.3.2 Система управління водовідливом.....	25
1.2 Структура і основні компоненти системи управління .....	26
1.2.1 Комп'ютерна мережа підприємства .....	26
1.2.2 Система управління водовідливом.....	28
1.3 Формулювання задачі дослідження .....	31
1.3.1 Комп'ютерна мережа підприємства .....	31
1.3.2 Система управління водовідливом.....	31
1.4 Висновки по розділу .....	32
2 Визначення моделі об'єкта управління .....	34
2.1 Комп'ютерна мережа підприємства .....	34
2.1.1 Аналітичні моделі комп'ютерних мереж.....	34
2.1.1.1 Моделювання незамкненої мережі.....	35
2.1.1.2 Моделювання замкнутої мережі.....	39
2.1.2 Алгоритм Бузена .....	43

	7
2.2 Система управління водовідливом.....	46
2.2.1 Розробка структурної схеми інформаційних потоків дослідницької системи .....	46
2.2.2 Визначення моделі об'єкт управління .....	49
2.2.3 Критерії якості функціонування системи управління.....	55
2.3 Висновки по розділу .....	57
3 Синтез комп'ютерної системи .....	58
3.1 Комп'ютерна мережа підприємства .....	58
3.1.1 Структура комплексу технічних засобів комп'ютерної мережі підприємства .....	58
3.1.2 Особливості структурної схеми комп'ютерної мережі.....	59
3.1.3 Обґрунтування і вибір обладнання.....	60
3.1.4 Побудова моделі в середовищі Cisco Packet Tracer .....	62
3.1.3.1 Налаштування основних параметрів пристроїв.....	64
3.1.3.2 Налаштування мереж VLAN, маршрутизації між VLAN і протоколу VTP .....	65
3.1.3.3 Налаштування протоколу STP .....	67
3.1.3.4 Налаштування маршрутизації.....	68
3.1.3.5 Реалізація DHCP та NAT .....	69
3.1.4 Розрахунок показників ефективності найбільшої під мережі .....	71
3.2 Система управління водовідливом.....	72
3.2.1 Модель системи управління з безперервним датчиком рівня.....	72
3.2.2 Модель системи управління з дискретним датчиком рівня .....	80
3.2.3 Оцінка якості функціонування системи управління .....	85
3.4 Висновки по розділу .....	86
4 Експериментальний розділ.....	88

	8
4.1 Комп'ютерна мережа підприємства .....	88
4.1.1 Розробка математичної моделі мережі .....	88
4.1.2 Розрахунок параметрів мережі по її моделі .....	90
4.1.2.1 Параметри роботи мережі без впливу шкідливого програмного забезпечення .....	91
4.1.2.2 Параметри роботи мережі під впливом вірусних програм .....	93
4.1.3 Робота мережі із скоригованими характеристиками проблемних вузлів ..	95
4.2 Система управління водовідливом.....	98
4.2.1 Цифрова модель системи управління .....	98
4.2.2 Розробка програмного забезпечення системи управління.....	101
4.2.3 Розробка програмного забезпечення людино-машинного інтерфейсу ....	104
4.2.4 Перевірка функціонування програмного забезпечення .....	107
4.3 Висновки по розділу .....	109
Висновки .....	110
Перелік посилань.....	113
Додаток А - Тексти програм налаштування мережі комп'ютерної системи ....	116
Додаток Б - Тексти програм реалізації математичної моделі комп'ютерної мережі як замкнутої системи масового обслуговування .....	124
Додаток В – Програмні модулі системи управління .....	140
Додаток Г – Програмное обеспечение системы управления .....	147
Додаток Д – Програмне забезпечення людино-машинного інтерфейсу .....	152
Відгуки консультантів кваліфікаційної роботи .....	155
Відгук.....	157
Рецензія .....	159

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

КСА НА – комп’ютерна система насосних агрегатів;

АРМ – автоматизоване робоче місце;

ТЛ – технологічної лінія;

ПО – пульт оператора;

ТС – телесигналізація;

ТВ – телевимірювання;

ТУ – телеуправління;

ІВ – виконавчий орган;

ПЛК – програмований логічний контролер;

PROFIBUS, CANOpen – відкриті мережеві інтерфейси;

VLAN – віртуальна локальна мережа;

DHCP – протокол динамічної настройки вузла;

NAT – трансляція мережевих адрес

RS-485 – технологія передачі даних по мережі;

КП – контролюючий пристрій.



## ВСТУП

Велику роль в утворенні режиму рівня води відіграють природні фактори - атмосферні опади і водофільтраційна якість порід, але основним джерелом поповнення шахти водою є підземна вода. Водоносні горизонти характеризуються складною циркуляцією, зумовленою водопроникністю породи, як окремих структурних горизонтів, так і в самому горизонті, а також різними тисками підземних вод.

На руднику для попередження затоплення горизонтальних виробок підземними або поверхневими водами шляхом відкачування шахтних вод, називається рудниковим водовідливом.

Стаціонарні рудничні водовідливні установки повинні забезпечувати надійну відкачування води з підземних виробок при можливо менших експлуатаційних витратах [1].

Водовідливні установки гірничих підприємств, що відпрацьовують обводнені родовища, представляють складний енергомеханічний комплекс, що включає: насосні агрегати різної продуктивності, трубні колектори, систему електропостачання і апаратуру автоматизації.

Основне завдання автоматизації цих об'єктів полягає у вивільненні обслуговуючого персоналу і забезпечення надійного і економічного функціонування процесу водовідливу, щоб виключити затоплення гірничих виробок і створити нормальні умови для ведення гірничих робіт.

Тривалий досвід експлуатації автоматизованих водовідливних установок свідчить про значно більшої надійності їх роботи в порівнянні з ручним керуванням. На автоматизованих водовідливних установках ефективніше використовується обладнання і більш ніж удвічі зберігається чисельність обслуговуючого технічного персоналу.

Обрана тема актуальна для вугільних шахт та спрямована на зниження собівартості камінного вугілля.

Одним із важливих напрямів застосування комп'ютерної техніки сьогодні є створення мереж, розроблення методів і засобів розподіленої обробки даних.

Термін "розподілена обробка даних" означає використання комп'ютерів, що пов'язані один з одним для скоординованого виконання одної або кількох задач. Розподілена обробка даних передбачає наявність двох і більше комп'ютерів та засобів комунікації між ними. Нині розподілена обробка даних здійснюється, як правило, за допомогою комп'ютерних мереж, що забезпечують єдиний інформаційний простір для багатьох користувачів. Сьогодні у світі бізнесу комп'ютерна мережа – це більше, ніж набір з'єднаних між собою пристроїв. Для підприємства комп'ютерна мережа - це ресурс, який дає змогу працівникам збирати, аналізувати, організовувати і розповсюджувати інформацію, яка є основою їхнього бізнесу і джерелом прибутковості всього підприємства. Для забезпечення інформаційного обміну підприємства в цілому виконується впровадження структурованої кабельної системи (SCS) як необхідного технічного фундаменту для побудови інформаційних і автоматизованих систем. СКС дозволяє звести в єдину систему безліч інформаційних сервісів різного призначення (локальні обчислювальні і телефонні мережі, електро-мережі, системи безпеки, відеоспостереження, сигналізації й т. п.).

### **Мета роботи і завдання дослідження.**

Удосконалення комп'ютерної мережі підприємства.

Визначити вузли мережі, на які найсильніше впливає інформаційне перевантаження мережі. Визначити умови, та параметри властивостей технічних пристроїв мережі за яких може виникнути втрата продуктивності. Розробити рекомендації що до модернізації мережі для підвищенні її стійкості до перевантажень.

Вирішити основне завдання, що стоїть перед системою управління насосною станцією - відкачування води таким чином, що б забезпечити мінімальні енерговитрати з дотриманням вимог технологічного процесу.

### **Об'єкт дослідження**

Комп'ютерна мережа вугільної шахти ім. Сташкова, що лежить в основі забезпечення роботи, менеджменту, фінансових служб, транспорту та безпеки в організаційній структурі.

Автоматична система управління насосною станцією відкачування води з мінімальними енерговитратами, та дотриманням вимог технологічного процесу.

### **Предмет і методи дослідження**

Предметом дослідження є структура комп'ютерної мережі, інформаційні властивості мережі а також технічні можливості апаратних засобів. Для виконання завдань дослідження використовуються методи теорії масового обслуговування, розроблена математична модель комп'ютерної мережі як мережі масового обслуговування. Проведено дослідження властивостей мережі з різними параметрами інформаційного середовища та апаратних засобів.

Насосна станція - об'єкта управління, який потребує розробки моделі приймального резервуару і модель насосного агрегату. На підставі отриманих моделей розробити узагальнену модель насосної станції, з можливістю завдання витрати води, швидкості обертання насосів, контролю рівня води в резервуарі.

### **Ідея роботи**

Визначення найбільш «слабких» вузлів комп'ютерної мережі, які у випадку перевантаження приводять до втрати працездатності мережею і розробка рекомендацій для вирішення такої проблеми. Практичні результати – визначення недоліків і шляхів удосконалення комп'ютерної мережі досягаються завдяки використанню наукового підходу до вирішення поставлених завдань. Застосовані наукові розробки з моделювання комп'ютерних мереж показали свою достовірність у багатьох прикладах їх застосувань.

Для оцінки можливості, підвищення ефективності функціонування системи управління, необхідно використовувати алгоритм, який дозволяє, виконувати відкачку воде з верхнього рівня з максимальною продуктивністю насосних агрегатів. Дане рішення призведе до загального збільшення часу, коли вода знаходиться на високому рівні, що в свою чергу призводить до зменшення витрат електричної енергії.

# 1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ

## 1.1 Загальна характеристика об'єкта управління

### 1.1.1 Характеристика і організаційна структура виробництва

Розглянемо організаційну структуру відділу головного енергетика вугільної шахти (рис. 1.1).

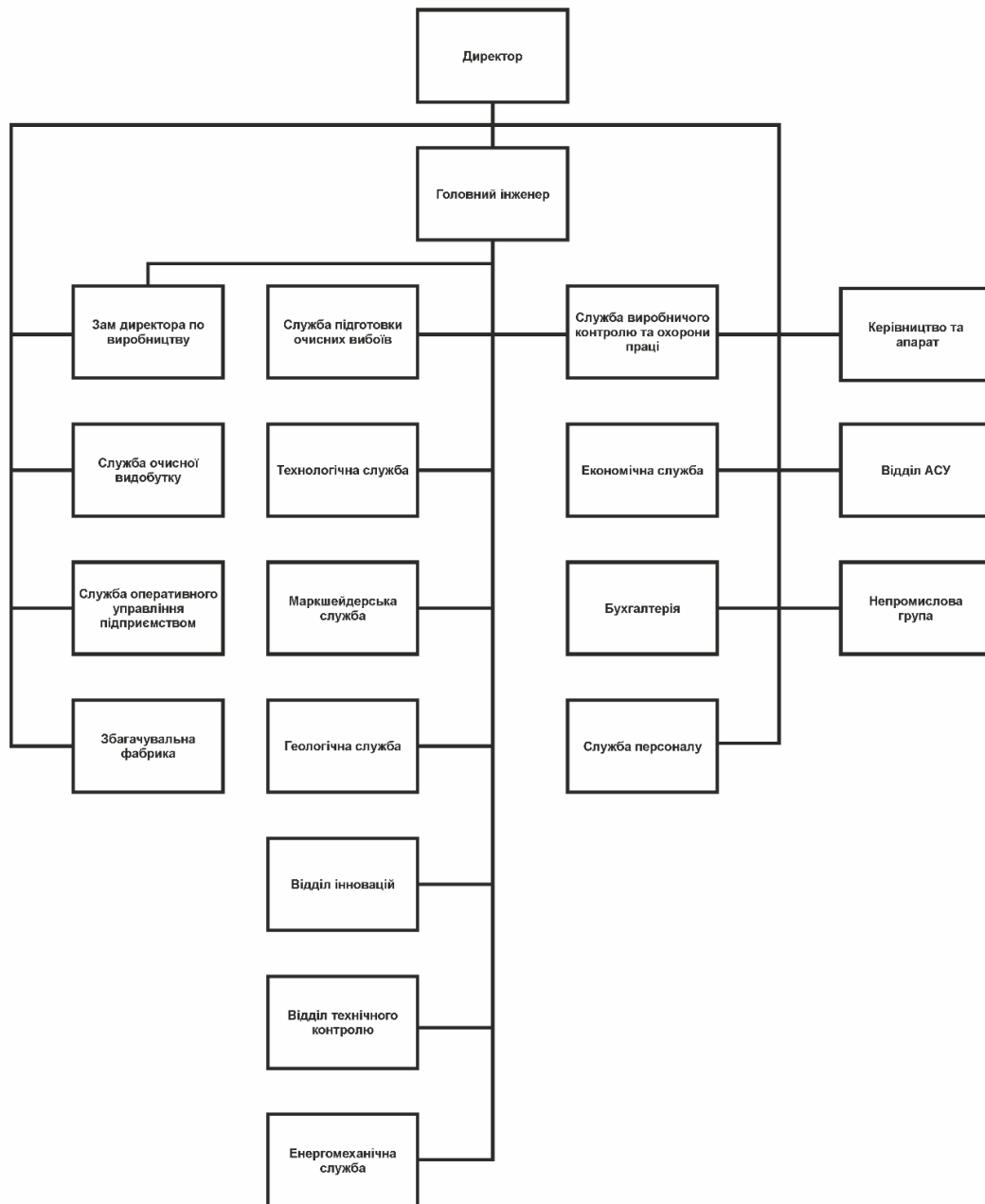


Рисунок 1.1 – Організаційна структура

Організаційна структура шахти є функціонально-лінійною структурою (штабне управління) та являє собою комбінацію лінійної структури з системою виділення певних функцій.

Лінійно-штабна структура є гарним проміжним ступенем при переході від лінійної структури до більш ефективних структур.

На чолі організаційної структури знаходиться директор. Головний інженер підприємства є першим заступником директора та знаходиться в його безпосередньому підпорядкуванні.

Також директору підпорядковуються: заступник директора з виробництва, який керує службами: очисної видобутку, оперативного управління виробництвом та збагачувальної фабрикою.

У безпосередньому підпорядкуванні головного інженера знаходяться:- служба підготовки очисних вибоїв, технологічна служба, маркшейдерська служба; геологічна служба, відділ інновацій, відділ технічного контролю, енергомеханічна служба, служба виробничого контролю та охорони праці, економічна служба, бухгалтерія, служба персоналу та служба з господарських питань.

Безпосередньо директору підпорядковується керівництво та апарат при керівництві: провідний спеціаліст по роботі з цінними паперами, начальник відділу ГО та НС, старший інспектор з діловодства, архіваріус, відділ АСУ, непромислова група.

## **1.1.2 Аналіз об'єкта впровадження**

### **1.1.2.1 Комп'ютерна мережа підприємства**

Сучасні інформаційно-комп'ютерні технології контролю та управління виробництвом на вугільних шахтах з усе більшою інтенсивністю завойовують визнання в вугледобувних країнах світу.

У Німеччині на вугільних шахтах реалізується концепція "частково автоматизованої шахти", яку відповідно економічно доцільно вирішувати на ЕОМ задачі управління окремими технологічними процесами та комплексами з відповідальністю людини за прийняття рішення. Для систем управління на

вугільних шахтах фірмою "Сіменс" розробляються технічні засоби, які включають набір підземних і поверхневих перепрограмованих контролерів.

Найбільш відомі системи контролю і управління для гірничої промисловості фірми "Трансміттон", яка постачає транспортні системи на шахти багатьох країн світу. Такі системи для управління вентиляцією встановлені і на декількох шахтах Росії, в тому числі в Кузбасі, і забезпечують:

- дистанційний контроль і управління;
- генерування і реєстрацію аварійних сигналів;
- висновок інформації на дисплей з активними мікросхемами (стандартного стану, динаміки і діагностики з можливостями оперативної конфігурації).

Об'єднана станція для навколишнього середовища цієї фірми має повний набір датчиків газу, вентиляції та складу навколишнього середовища, включаючи окис вуглецю і метан.

Вугільна шахта - унікальна, складна виробнича система з особливо небезпечними умовами (вибухо-, пожежонебезпечна, небезпечна за проривами води і газу, гідродинамічним явищам і ін.), Де існують не тільки непередбачені і виняткові зміни геологічних умов і природних сил, а й невиконання правил безпеки або невірне дію одного робочого може призвести до катастрофічних наслідків - численних людських жертв і величезних матеріальних втрат.

Автоматизована система управління технологічним процесом представляє собою ієрархічну структуру, що включає в себе рівень операторського управління, рівень автоматичного управління та польовий рівень.

На польовому рівні знаходяться пристрої, що встановлюються безпосередньо на технологічних об'єктах та поблизу від них: датчики, виконавчі пристрої, спеціалізовані контролери, сервоперетворювачі, системи віддаленого вводу-виводу, частотні перетворювачі. Основними завданнями, які розв'язуються на даному рівні, є: вимірювання технологічних параметрів процесу та вироблення керуючих впливів.

На рівні автоматичного управління знаходяться пристрої управління та сигналізації, які розміщуються в шафах управління: програмовані логічні

контролери, системи віддаленого вводу-виводу, пульти управління. Пульти управління можуть бути реалізовані за допомогою кнопок, перемикачів та світлосигнальної арматури; панелей операторів; комп'ютерів або терміналів зі спеціалізованим програмним забезпеченням, що реалізує людино-машинний інтерфейс. Основними завданнями, які розв'язуються на даному рівні, є: обробка даних, що надходять з польового рівня, формування значень керуючих впливів та передача їх на польовий рівень, сигналізація про вихід значень параметрів технологічного процесу за задані межі, блокування помилкових дій персоналу та управляючих пристроїв, реалізація протиаварійного захисту при виникненні нештатних ситуацій.

На рівні операторського управління знаходяться пристрої збору та зберігання інформації та пристрої візуалізації технологічних процесів: сервера SCADA систем, сервера баз даних, автоматизовані робочі місця. Основними завданнями, які розв'язуються на даному рівні, є: збір інформації, що надходить з рівня автоматичного управління, її обробка, зберігання та архівування, формування звітів, передача інформації EMS системі, обчислення параметрів які не можуть бути вимірювані, діагностика та захист від збоїв, налаштування управляючих пристроїв.

SCADA (Supervisory Control And Data Acquisition System – Диспетчерське управління та збір даних) система являє собою програмне забезпечення призначене для забезпечення роботи в реальному часі систем збору, обробки, відображення та архівування інформації про об'єкт контролю або процеси управління. Сервера SCADA системи вирішують завдання збору, резервування, архівування та надання інформації, а автоматизовані робочі місця операторів вирішують завдання візуалізації технологічного процесу за допомогою НМІ (Human Machine Interface - Людино-машинного інтерфейсу), формування сигналів тривоги, запису інформації про нештатні ситуації, формування звітів, організації управління рецептами.

Як видно зі структури автоматизованої системи управління технологічним процесом програмовані логічні контролери знаходяться на рівні автоматичного управління. Вони отримують дані від датчиків, систем віддаленого вводу-виводу

та спеціалізованих контролерів, на підставі отриманої інформації відповідно до алгоритму управління формуються значення керуючих впливів які передаються виконавчим пристроям, як безпосередньо, так і за допомогою спеціалізованих контролерів, систем віддаленого вводу-виводу, перетворювачів. Для контролю окремих ділянок технологічного процесу та зміни режимів роботи локальних систем управління використовуються пульти оператора, а для контролю за технологічним процесом та зміни налаштувань управляючих пристроїв використовуються автоматизовані робочі місця.

Таким чином сучасна система управління базується на використанні у якості пристроїв управління програмованих логічних контролерів. Які отримують інформацію від датчиків формують за заданим алгоритмам сигнали управління які за допомогою виконавчих пристроїв формують керуючі впливи. Зміна параметрів системи управління та візуалізація процесу виконується за допомогою SCADA систем які встановлюються на серверах та автоматизованих робочих місцях операторів. SCADA системи виконують збір, обробку візуалізацію та зберігання інформації отриманої від програмованих логічних контролерів та операторських пультив.

### **1.1.2.2 Система управління водовідливом**

Водовідлив гірничих підприємств є важливим елементом всього комплексу гірничо-технічного обладнання, і його надійна робота під чому визначає безперебійність і безпеку ведення гірських робіт з видобутку корисних копалин. Головна водовідливні установка сучасної вугільної шахти – великий споживач електроенергії (до 25% загальної витрати енергії шахти). Економічна робота водовідливної установки шахти і раціональний режим електроспоживання істотно впливають на економічні показники всього гірського підприємства.

Економічна і надійна робота водовідливної установки багато в чому визначається умовами фактичного режиму роботи насоса на зовнішню мережу, які в процесі експлуатації постійно змінюються внаслідок підвищеного зносу устаткування при перекачуванні вод зі взвєсями, кислих вод, «заростання»



елементів трубопровідної мережі, кавітаційних явищ, зашламлення водозбірних ємностей. В результаті сукупного впливу цих факторів змінюються робочі параметри насосної установки: подача, натиск, К.К.Д., що в свою чергу призводить до перевитрати електроенергії на водовідлив.

Водовідливні установки гірничих підприємств, що відпрацьовують обводнені родовища, представляють складний енергомеханічний комплекс, що включає: насосні агрегати різної продуктивності, трубні колектори, систему електропостачання і апаратуру автоматизації.

Одним з напрямків підвищення надійності і ефективності роботи комплексу водовідливу шахти є управління водовідливними установками технічними засобами автоматизації. Існуючі засоби автоматизації не в повній мірі вирішують проблеми експлуатації водовідливних установок, виконані на «застарілої» елементній базі. Тому розробка системи моніторингу стану і керування комплексом водовідливу шахти є актуальною темою.

У систему шахтного водовідливу входять: пристрої для регулювання внутрішньо шахтного стоку (водовідвідні канавки, трубопроводи, перекачувальних насоси), водозбірники, насосні трубопроводи, перекачувальних насоси), водозбірники, насосні станції з водозабірних колодзями і відпливними установками, з усмоктувальними і нагнітальними трубопроводами.

Головні, дільничні водовідливи та станції перекачування обладнані електроагрегатами, до складу яких входять відцентрові насоси, приводом яких є асинхронні електродвигуни з короткозамкненим ротором.

Схема відкачування шахтних вод насосними станціями шахтного водовідливу шахти ім. Н.І. Сташкова показана на рис. 1.2.

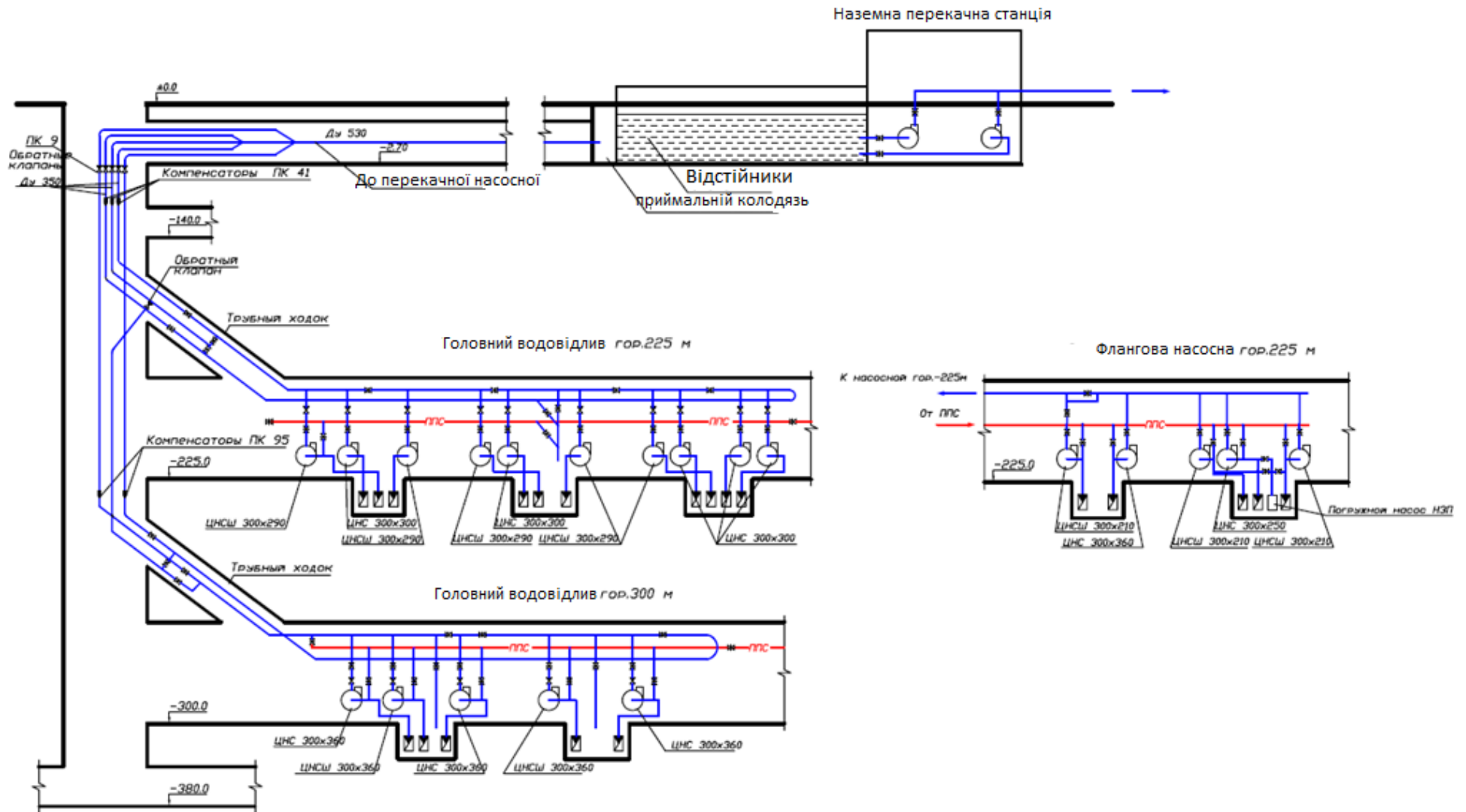


Рисунок 1.2 – Схема відкачування шахтних вод насосними станціями шахтного водовідливу шахти ім. Н.І. Сташкова

Нормальний добовий приплив води в шахту ім. Н.І. Сташкова 1 200 м<sup>3</sup>/добу, глибина шахти складає 300 м.

У систему водовідливних установок шахти ім. Н.І. Сташкова входять: головна водовідливна установка горизонту – 300 м, головна водовідливна установка горизонту – 225 м, флангова водовідливна установка горизонту – 225 м і наземна перекачувальна станція.

На Східному магістральному вентиляційному штреку №3 майданчика С6 гор. 225 м розташований один водозбірник. Стави видачі кожного насоса обладнані засувками і зворотними клапанами для захисту насосів від гідроударів.

Наземна перекачувальна станція обладнана 2-я насосними агрегатами з насосами: ЦНС 300/300 – 2 шт., ВАО2 з електроприводом 6 кВ, 500 кВт.

Дана перекачувальна станція призначена для прийому припливу від водовідливних установок горизонтів -300 м, -225 м і флангової -225 м в приймальний відстійник з подальшою перекачкою в поверхневі відстійники.

Приймальний відстійник розділений на три басейни по 335 м<sup>3</sup> кожний.

До найважливіших операцій з управління водовідливної установкою відноситься заливка насоса перед включенням його в роботу. В даний час найбільшого поширення набув спосіб заливки за допомогою заливного занурювального насосу (ЗЗН), який знаходиться в водозбірнику нижче нижнього рівня і постійно залитий водою.

На рис. 1.3 представлена гідравлічна схема автоматичної водовідливної установки.

Експлуатація стаціонарних шахтних водовідливних установок включає:

1) керування електро- агрегатами – пуск і зупинку їх в залежності від прийнятої організації роботи та контроль за основними параметрами, що визначають режим роботи (тиск в зворотному і підводі патрубках, подачу насоса, напруга і силу струму, який живить електродвигун, температуру підшипників, витрату води через розвантажувальний пристрій);

2) технічне обслуговування, що полягає в огляді всіх водовідливних установок щодоби особами, призначеними головним механіком, перевірці та

забезпеченні нормального стану вузлів, засувок, зворотних клапанів, пусковий і контролює апаратури, організації та проведенні поточних профілактичних ремонтів і встановленні строку направлення насосів на заводи для проведення капітального ремонту.

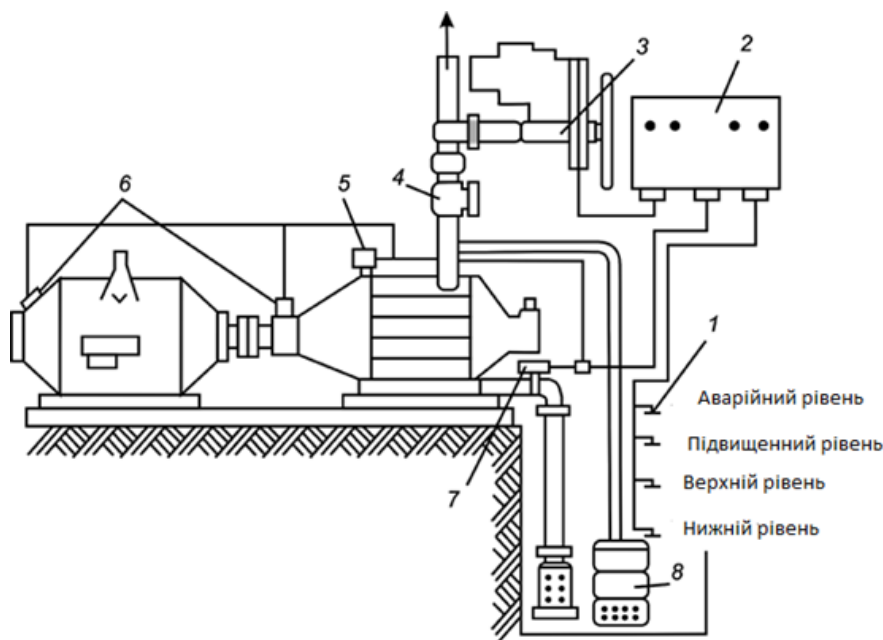


Рисунок 1.3 – Гідравлічна схема автоматичної водовідливної установки

- 1 – датчики рівня;
- 2 – апарат управління (реле контролю рівня);
- 3 – виконавчий механізм засувки з електроприводом;
- 4 – водовідливної трубопровід із зворотним клапаном;
- 5 – датчик-реле тиску;
- 6 – датчики температури;
- 7 – датчик-реле продуктивності;
- 8 – занурювальний заливний насос.

Основними функціями системи керування є запуск, зупинка і автоматичне регулювання електронасосних агрегатів в залежності від рівня води в водозбірнику, який визначає ефективність відкачування води з шахти.

Встановлений верхній рівень води є імпульсом для підготовки і запуску агрегату в роботу, нижній рівень води є імпульсом зупинки.

Таким чином, у другій групі технологічних схем спрощено автоматичне керування електро- агрегатом.

Організація відкачування води водовідливної установкою виконується від рівня води в водозбірнику. При відкачці води в залежності від рівня води в водозбірнику включення і відключення водовідливної установки здійснюється при досягненні заданих фіксованих рівнів.

Стан водовідливної установки, як об'єкта керування, характеризують такі експлуатаційні параметри:

- приплив води в водозбірник;
- рівень води в водозбірнику;
- подача водовідливної установки;
- витрата води через розвантажувальний пристрій насоса;
- частота обертання робочого колеса насоса;
- тиск води в нагнітальному трубопроводі;
- температура підшипників насоса і електродвигуна;
- активна потужність, споживана електродвигуном насоса;
- вібрація насоса і електродвигуна.

Однією з особливостей функціонування водовідливної установки є мінливість її експлуатаційних параметрів, що призводить до виходу робочого режиму насоса із зони промислового використання і відповідно до збільшення експлуатаційних витрат, неможливістю насосом перекачувати необхідний об'єм води.

Експлуатаційними режимами комплексу наземної станції (НС) перекачування шахтних вод є: пуск, робочий режим, зупинка.

Режим пуску водовідливної установки починається при досягненні водою в водозбірнику верхнього допустимого рівня.

В режимі пуску виконується заливка насоса з протипожежного ставу протягом 5 хвилин при закритій засувці. Після чого виконуються запуск приводу насоса.

У робочому режимі роботи наземної станції здійснюється перекачування води насосами, насосними групами.

Основні вимоги до устаткування НС по продуктивності, потужності і режимам функціонування визначаються вимогами СніП 2.04.03-85. Залежно від передбачуваних об'ємів складається технічне завдання для проектування насосної станції.

Застосування сучасних технологічних рішень дозволяє звести до мінімуму терміни монтажу станцій. Модульні станції перекачування вод мають різні характеристики по габаритах і по вазі, по споживаній потужності і по продуктивності. Зовнішній діаметр їх корпусів варіюється в межах від 0,8 до 3,5 метрів при висоті від 3 до 11 метрів.

Сучасні насосні станції оснащуються занурюваними насосами, кількість яких визначається об'ємами і характеристиками перекачуванні вод. Робочі агрегати стикуються з трубопроводами через спеціальні фланцеві з'єднання. Облаштування цих механізмів таке, щоб можна було забезпечити дистанційований демонтаж насосів для ремонту. Насосні агрегати піднімаються на поверхню за допомогою ланцюга або троса без необхідності спуску вниз.

Насосні станції, технічні характеристики і характер використання, яких не вимагають втручання людини в їх роботу, називаються автономними. В процесі експлуатації вони не потребують постійної присутності персоналу. Технічне і регламентне обслуговування робиться відповідно до встановленого графіку і експлуатаційної документації.

В якості об'єкту управління в цій роботі виступає насосна станція технічні характеристики якої приведені в табл. 1.1.

Таблиця 1.1 - Технічні характеристики НС

№	Найменування параметра	Значення
1	Максимальна витрата води, що поступає в приймальний резервуар, м <sup>3</sup> /ч	600
2	Максимальна продуктивність насосів, мЗ/ч	1 200
3	Діаметр приймального резервуару, м	46
4	Активна висота приймального резервуару, м	10
5	Довжина напірного колектору, м	1 500
6	Перепад висот напірного колектору, м	12

### **1.1.3 Принцип функціонування об'єкту управління**

#### **1.1.3.1 Комп'ютерна мережа підприємства**

Комп'ютерна система відноситься до класу фінансово-управлінських систем (малі інтегровані системи). Система повинна гнучко налаштуватися на потреби конкретного підприємства, добре інтегрувати діяльність підприємства і призначені, насамперед, для обліку й управління ресурси невиробничих компаній. Хоча у багатьох системах даного класу присутні базові можливості управління виробництвом. Як правило, вони універсальні, функціональні можливості таких систем ширші, ніж локальних.

Комп'ютерна інформаційна система підприємства (КІСП) це сукупність економіко-математичних методів і моделей, технічних, програмних, технологічних засобів і рішень, а також спеціалістів, призначена для обробки інформації й прийняття управлінських рішень [2,15].

Забезпечувальна частина КІСП складається з технічного, інформаційного, математичного, організаційного, правового, ергономічного й іншого видів забезпечення.

Інформаційна система (ІС) – це система, яка організує зберігання і маніпулювання інформацією про проблемну область. Під терміном «маніпулювання» маються на увазі процедури збору, обробки, пошуку, передачі інформації, необхідної в процесі прийняття рішень в будь-якій області. У основі функціонування будь-якої системи лежить процес, а в основі інформаційної системи – процес виробництва інформації [2]. Тому призначення інформаційної системи – це виробництво інформації для потреб організації в забезпеченні ефективного управління її діяльністю. ІС можна розглядати як систему управління, де процес виробництва інформації є об'єктом управління. Як у будь-якій системі управління, в ІС існують органи управління (рис. 1.2).

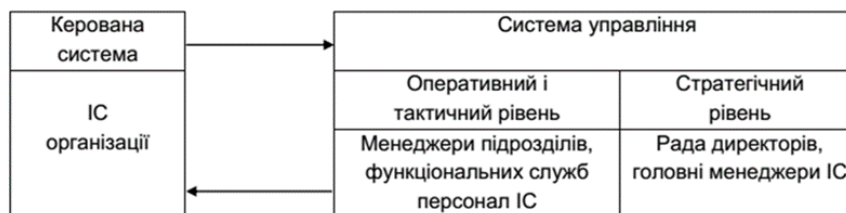


Рисунок 1.5 – Структура інформаційної системи підприємства

Повноцінне функціонування КІСП можливе лише на базі сучасної комп'ютерної мережі [20]. Комп'ютерна система і локальна мережа підприємства повинна забезпечувати менеджмент за наступними напрямками діяльності.

### 1.1.3.2 Система управління водовідливом

Незалежно від типу насосної станції принцип роботи усіх систем однаковий. У приймальну частину зливається вода, які завдяки щільному прокладенню не просочуються в ґрунт, і під тиском примусово прокачуються насосами в напірний трубопровід. Далі води потрапляють в розподільну камеру і по трубах транспортуються на очисні споруди. Щоб води не поверталися назад в трубопровід насоса, передбачено зворотний клапан. Якщо об'єм води істотно підвищується, включається додатковий насос. Якщо насоси не справляються з об'ємами вод, включається сигналізація. Робота НС управляється автоматично. Контроль рівня води, здійснюється за допомогою поплавцевих датчиків, розташованих на різних рівнях, завдяки чому станція працює в такому режимі. Датчики першого рівня вказують на малий об'єм води, насоси не працюють. Датчики другого рівня включають насос на перекачування. Об'єм води в межах нормального. Датчики третього рівня спрацьовують при підвищеному об'ємі вод і включають резервний насос для відкачування надлишку води. Датчики четвертого рівня включають аварійний сигнал, оскільки пристрої для відкачування вод не справляються з їх об'ємом. В цьому випадку обслуговуючій бригаді необхідно вжити заходи для нормалізації роботи НС, оскільки сигналізація може включитися в результаті поломки одного з насосів. Коли



перекачування води завершено, рівень води падає нижче першого датчика, система відключається. При наступному включенні спрацьовує інший насос, що виконував до цього функцію додаткового. Така система роботи дозволяє запобігти передчасному зносу механізмів одного насоса.

## **1.2 Структура і основні компоненти системи управління**

### **1.2.1 Комп'ютерна мережа підприємства**

Локальна мережа не обійдеться без установки сервера. Монтаж кабельної розводки починається з технічного завдання для тимчасової локальної мережі. Так само має бути сформовано технічної завдання на логічному рівні: вимоги до сервера, ПО: база даних, ftp-сервер, інтернет сервер, принт-сервер, реалізація політики безпеки. Зазвичай ці вимоги пред'являються адміністратору обслуговуючому локальну мережу організації або фірмі, яка разом з монтажем виконує настройку локальної мережі. У такій мережі можна встановити свіч більш високого рівня, з гігабітними портами для підключення допустимого до гігабітного адаптера сервера. Наприклад в такій мережі доступ до Інтернету вже буде здійснюватися через сервер на якому встановлюється ПО надання доступу та моніторингу діяльності персоналу в інтернеті. Кожен комп'ютер може мати свої права доступу відповідно з політикою безпеки сервера в домені. Кожен комп'ютер для авторизації в домені повинен вводити ім'я і пароль видані адміністратором мережі.

«Зірка» є найбільш поширеною. При використанні топології кожен вузол (окрема робоча станція) під'єднується до ЛВС за допомогою власного кабелю, один роз'єм якого підключається до мережного адаптера, а інший - до концентратора [14].

Технологія Ethernet швидко завоювала загальне визнання і стала основною технологією для ЛОМ, з часом також поширившись на регіональні обчислювальні мережі. Вона застосовується в різних цілях і середовищах.

Успіх технології Ethernet обумовлений її надзвичайно високою адаптивністю. При будь-якій пропускній здатності використовується один і той же протокол і

формат кадрів MAC. Відмінності спостерігаються на фізичному рівні, у визначенні методу сигналізації і засобах передачі.

Характеристики Ethernet з гігабітними швидкостями передачі даних.

У міру зростання вимог до продуктивності комп'ютерів, компанії стали використовувати системи ЛОМ, пов'язані високошвидкісними магістральними лініями.

Для задоволення таких потреб комітет IEEE 802.3 розробив ряд специфікацій для підвищення пропускної здатності Ethernet до 100 Мбіт/с, а ще через кілька років були створені стандарти для гігабітного Ethernet. У кожній новій специфікації нові засоби передачі і схеми кодування будувалися на основі вже відомої технології Ethernet, що робило перехід на нові стандарти простіше, ніж якби кожен раз специфікації створювалися з нуля.

Гігабітний стандарт включає ряд варіантів передачі даних:

1000BASE-SX: Короткохвильовий варіант. Оптиволоконний багатомодовий кабель діаметром 62,5 мкм і довжиною до 275 м або діаметром 50 мкм і довжиною до 550 м, що підтримує дуплексні лінії. Використовувані довжини хвиль знаходяться в діапазоні від 770 до 860 нм.

1000BASE-LX: Довгохвильовий варіант. Оптиволоконний багатомодовий кабель діаметром 62,5 мкм або 50 мкм, підтримуючий дуплексні лінії довжиною до 550 м або одномодовий кабель діаметром 10 мкм завдовжки до 5 км. Використовувані довжини хвиль знаходяться в діапазоні від +1270 до 1355 нм.

1000BASE-CX: Цей варіант підтримує гігабітні лінії зв'язку між пристроями, розташованими в одному приміщенні або в одній апаратній стійці, для яких використовуються мідні перемички (спеціалізовані екрановані кабелі з кручених пар протяжністю не більше 25 м). Кожна лінія складається з окремої екранованої крученої пари, дані по якій передаються в обидві сторони.

1000BASE-T: Цей варіант використовує чотири неекранованих кручених пари категорії 5 для зв'язку з пристроями на відстані до 100 м, передаючи і отримуючи дані на всі чотири пари одночасно з ехокомпенсацією.

Як правило, при використанні гігабітного Ethernet, опорний комутатор

ЛОМ зі швидкістю 1 Гбіт/с забезпечує зв'язок по магістральній лінії з центральними серверами і комутаторами Ethernet високошвидкісних робочих груп. Кожен комутатор робочої групи підтримує як зв'язок зі швидкістю 1Гбіт/с для з'єднання з опорним комутатором ЛОМ і підтримки високопродуктивних серверів робочих груп, так і зв'язок зі швидкістю 100 Мбіт/с для роботи з високопродуктивними робочими станціями, серверами і комутаторами ЛОМ зі швидкістю 100 Мбіт/с [22].

### 1.2.2 Система управління водовідливом

У рамках кваліфікаційної роботи була розроблена система управління насосною станцією, структура якої приведена на рис. 1.6.

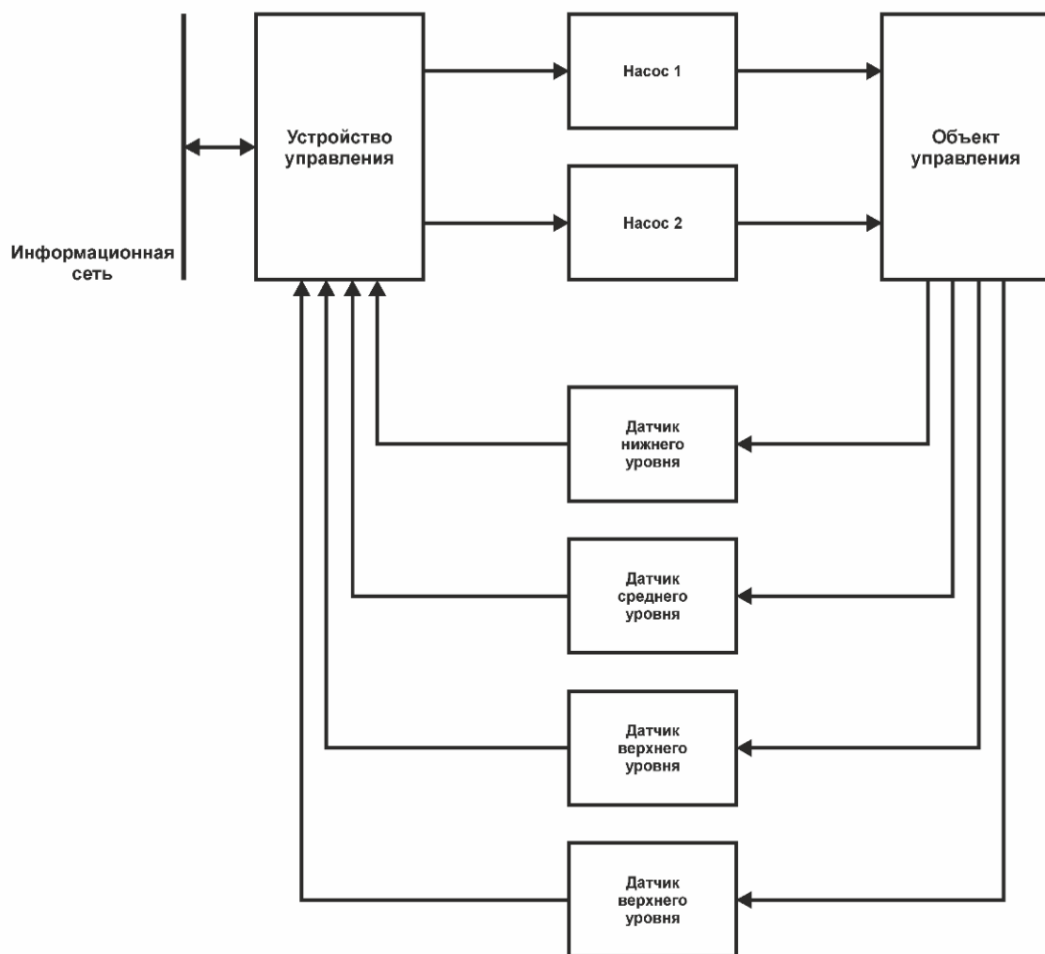


Рисунок 1.6 - Структура системи управління

Облаштуванням управління виступає програмований логічний контролер VIPA 214-1BA03. До нього підключені чотири набори датчиків що відповідають за контроль рівня стічних вод на нижньому, середньому, верхньому і аварійному рівнях.

Управління рівнем води здійснюється за допомогою двох комутаційних пристроїв, які підключають один насос до пускача, а другий до частотного перетворювача. Це забезпечує можливість зміни працюючого насос, що зменшує загальний знос устаткування. В якості насосів на станції використовуються насоси ЦНС 300/300. Технічні характеристики насоса приведені в табл. 1.2.

Принцип дії високовольтних перетворювачів частоти (ВПЧ) серії PERFECT HARMONY заснований на перетворенні електричної енергії трифазної мережі змінного струму напругою 3, 6 або 10 кВ з постійними параметрами частоти і напруги в трифазну напругу 3, 6 або 10 кВ із змінюваними параметрами частоти і напруги за законом частотного регулювання.

Таблиця 1.2 - Технічні характеристики насосної станції

№	Найменування параметра	Значення
1	Тип	Занурюваний
2	Напруга живлення, В	~6000
3	Потужність, що підводиться, кВт	600
4	Номинальна потужність, кВт	500
5	Кількість полюсів	6
6	Номинальний струм, А	50
7	Пусковий струм, А	70
8	Номинальна швидкість, про/мін	1450
9	Співвідношення моментів, що крутять	2,3
10	Момент інерції, кг·м <sup>2</sup>	0,0227
11	ККД, %	83,4
12	Максимальна витрата, л/з	40,3
13	Максимальний гідростатичний натиск, м	19,1
14	Максимальне ККД, %	48
15	Максимальний робочий тиск, атм	5,92
16	Максимальна кількість пусків в годину	20
17	Момент інерції, кг·м <sup>2</sup>	45

ВПЧ - виріб, що включає увесь комплекс зашит і блокувань, забезпечує частотний пуск/останов і регулювання швидкості обертання високовольтного двигуна від нуля до номінальної (і вище) частоти обертання.

Діапазон потужностей електроприводів ВПЧ - від 250 кВт до 67 000 кВт.



Рисунок 1.5 - Високовольтні регульовані електроприводи виробництва Siemens

ВПЧ є повністю цифровими пристроями, в яких реалізована найсучасніша схема перетворення енергії на базі окремих IGBT -транзисторних інверторних осередків з багаторівневою ШИМ-модуляцією. У ВПЧ реалізовані сучасні векторні методи управління електричними двигунами змінного струму : як асинхронними, так і синхронними.

Таблиця 1.3 - Основні технічні характеристики ВПЧ

Діапазон потужностей, кВт	250...67000
Частота живлячої мережі, Гц	45...55
Живляча напруга	3, 6 або 10 кВ
Вихідна напруга	0...3000 В, 0...6000 В, 0...10000 В
Діапазон регулювання частоти, Гц	0...120
Точність підтримки частоти, Гц	±0,01
Час пуску / останову, сік	10 від 3200 (параметри, що настраюються)
Перевантажувальна здатність	110% протягом 60сек
Коефіцієнт потужності	0,96
Аналогові входи	0~10V / 4~20mA (по вибору)
Аналогові виходи	0~10V / 4~20mA (по вибору)
Дискретні сигнали (вхідні/вихідні)	24 / 16 (з можливістю розширення кількості)
Цифрові і інтерфейси	RS485, Profibus, Ethernet і так далі
Органи управління	промислова панель оператора
Міра захисту	з сенсорним екраном
Конструктивне виконання	IP31, NEMA1 (повітряне охолодження),
Охолодження	IP52, NEMA12 (водяне охолодження),
Робоча температура	шафове одностороннього обслуговування,

ВПЧ мають систему діагностики, що дозволяє визначити місце виникнення несправності в електроприводі на рівні його функціональних частин. У ВПЧ передбачені засоби контролю і виміру вихідного струму і напруги ВПЧ, частоти змінного струму на виході ВПЧ і частоти обертання двигуна.

### **1.3 Формулювання задачі дослідження**

#### **1.3.1 Комп'ютерна мережа підприємства**

Для виконання завдання аналізу роботи комп'ютерної мережі необхідно визначити характеристики мережі у різних умовах роботи. На початковому етапі, який розглянуто у кваліфікаційній роботі, визначимо безрозмірні характеристики.

Головним завданням дослідження у кваліфікаційній роботі є визначення допустимих значень навантаження на існуючу мережу за яких мережа буде забезпечувати обробку запитів з мінімальною чергою.

Для забезпечення запасу по навантаженню необхідно промоделювати мережу з параметрами, що імітують дію шкідливого програмного забезпечення і визначити здатність мережі виконувати свої основні функції в цих умовах.

Розробити варіанти поліпшення роботи мережі в умовах значного інформаційного навантаження.

#### **1.3.2 Система управління водовідливом**

Аналіз літературних джерел показав, що основне споживання електроенергії насосною станцією пов'язане з роботою насосів. При цьому якщо рівень води в приймальному резервуарі знаходиться на рівні верхнього датчика витрати електроенергії зменшуються на 10 %.

Оскільки використовувані на насосних станціях насоси мають обмежену кількість пусків, підтримка рівня вод на заданому значенні за допомогою дискретного управління можливо тільки в пікові періоди. Усю решту часу споживання електроенергії є надмірним.

Розроблена структура системи управління і вибране апаратне забезпечення дозволяють здійснювати безперервне управління насосами. Виходячи з чого, завдання управління полягає в тому, що б за допомогою безперервного управління насосом забезпечити підтримку рівня води в приймальному резервуарі на рівні верхнього датчика. Що дозволить забезпечити мінімальне споживання електричної енергії насосною станцією.

При цьому повинне враховуватися те, що зі зменшенням частоти електроприводу насосного агрегату відбувається зменшення його коефіцієнта корисної дії, тому частота повинна знаходитися в межах від 30 до 50 Гц.

#### **1.4 Висновки по розділу**

Для виконання завдання аналізу роботи комп'ютерної мережі необхідно визначити характеристики мережі у різних умовах роботи. На початковому етапі, який розглянуто у кваліфікаційній роботі, визначимо безрозмірні характеристики.

Головним завданням дослідження у кваліфікаційній роботі є визначення допустимих значень навантаження на існуючу мережу за яких мережа буде забезпечувати обробку запитів з мінімальною чергою.

Для забезпечення запасу по навантаженню необхідно промодельовати мережу з параметрами, що імітують дію шкідливого програмного забезпечення і визначити здатність мережі виконувати свої основні функції в цих умовах.

Розробити варіанти поліпшення роботи мережі в умовах значного інформаційного навантаження.

У структуру системи водовідведення входять насосні станції, які виконують перекачування вод. У роботі цих станцій основним споживачем електроенергії являються насоси.

Забезпечення підвищення ефективності використання електроенергії можливо при максимізації часу роботи насосних агрегатів, коли вода знаходиться на рівні, що відповідає датчику верхнього рівня. Проте при цьому

частота живлення насосів не повинна опускатися нижче 30 Гц, для забезпечення достатнього коефіцієнта корисної дії.

Виходячи з цього, основним завданням, що стоїть перед системою управління насосною станцією являється відкачування вод таким чином, що б забезпечити мінімальні енерговитрати з дотриманням вимог технологічного процесу.



## **2 ВИЗНАЧЕННЯ МОДЕЛІ ОБ'ЄКТА УПРАВЛІННЯ**

### **2.1 Комп'ютерна мережа підприємства**

#### **2.1.1 Аналітичні моделі комп'ютерних мереж**

Вирішуючи задачу оцінки характеристик комп'ютерних мереж в умовах впливу шкідливих програм, необхідно враховувати випадкову природу багатьох факторів, від яких залежать характеристики КМ. Випадковими часто є моменти надходження в мережу вимог, обсяги інформації, послідовність необхідних для її обробки операцій. Складність структури КС і необхідність обліку випадкових факторів роблять задачу оцінки характеристик складної, причому в міру прогресу обчислювальної техніки і шкідливих інформаційних впливів на КМ складність цього завдання зростає. Тому широке поширення для оцінки пропускної здатності мережі отримав метод математичного моделювання.

Моделювання - один з найбільш поширених методів дослідження процесів функціонування складних систем. Найбільш поширеними з методів побудови математичних моделей є системи масового обслуговування (СМО).

Ускладнення структур і режимів реальних КС ускладнює застосування класичних методів теорії масового обслуговування (ТМО) з огляду на зростаючі розмірності вирішуваних завдань [4, 6].

Одним з можливих шляхів подолання розмірності є використання моделей в формі мереж масового обслуговування (МеМО).

МеМО являє собою сукупність кінцевого числа обслуговуючих вузлів, в якій циркулюють заявки, що переходять відповідно до маршрутної матриці з одного вузла в інший. Вузол завжди є розімкнутою системою масового обслуговування (СМО). При цьому окремі СМО відображають функціонально самостійні частини реальної комп'ютерної мережі підприємства, зв'язку між СМО - структурою мережі, а вимоги, що циркулюють по МеМО, - складові матеріальних потоків (повідомлення (пакети) в мережі).

В теорії МеМО фундаментальним є поняття стану мережі. Найважливіша характеристика мереж МО – ймовірності їх станів. Для визначення ймовірностей

станів MeMO досліджують вмережі випадковий процес. В якості моделей MeMO з процесів найбільш часто використовують марківські.

Марківським процесом з безперервним часом описують функціонування експоненційних MeMO. Мережа називається експоненціальною, якщо вхідні потоки вимог в кожному СМО пуассонівським, а час кожного етапу обслуговування, реалізованого на будь-який СМО мережі, мають експоненційний розподіл. Це дозволяє вважати, що етапи обслуговування незалежні між собою і не залежать від параметрів вхідного потоку, від стану мережі, від маршруту вправування вимог.

Аналітичні моделі розрахунку характеристик КМ базуються, як правило, на аналізі експоненційних MeMO. При використанні цього математичного апарату вдається отримати аналітичні моделі для вирішення широкого кола завдань дослідження комп'ютерних мереж.

### 2.1.1.1 Моделювання незамкненої мережі

Нехай дана незамкнута мережа, що складається з джерела пакетів (вузол 0) и К СМО  $M/M/m_1/\infty, M/M/m_2/\infty, \dots, M/M/m_K/\infty$ . Незамкнута мережа – це така відкрита мережа, в яку заявки надходять із зовнішнього середовища і йдуть після обслуговування з мережі в зовнішнє середовище. Іншими словами, особливістю незамкненої MeMO є наявність одного або декількох незалежних зовнішніх джерел, які генерують заявки, що надходять в мережу, незалежно від того, скільки заявок вже знаходиться в мережі. У будь-який момент часу у відкритій MeMO може перебувати довільне число заявок [4].

Задано  $P_R=(p_{ji})$  – маршрутна матриця,  $\mu_i$  – середня інтенсивність обробки пакета в одному конвеєрі  $i$ -го вузла,  $\lambda_0$  – інтенсивність вхідного в мережу потоку пакетів.

Потрібно: знайти інтенсивності потоків що надходять пакетів  $\lambda_i$ , мінімально необхідне число конвеєрів в кожному вузлі  $m_i$ , середню довжину черги  $r_i$ , середнє число пакетів в вузлі  $L_i$ , середній час перебування пакета в вузлі

$T_i$ , середнє число пакетів в мережі  $N$  і середній час перебування пакета в мережі  $T$ .

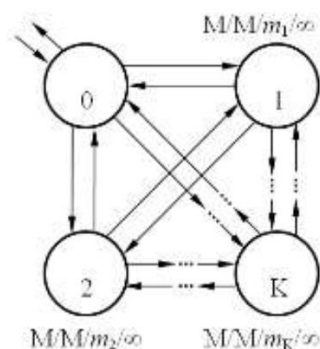


Рисунок 2.1 - Схема незамкнутої мережі

У загальному випадку мережа задається стохастичною маршрутної матрицею:

$$P_R = \begin{pmatrix} 0 & p_{01} & p_{02} & \dots & p_{0K} \\ p_{10} & p_{11} & p_{12} & \dots & p_{1K} \\ p_{20} & p_{21} & p_{22} & \dots & p_{2K} \\ \dots & \dots & \dots & \dots & \dots \\ p_{K0} & p_{K1} & p_{K2} & \dots & p_{KK} \end{pmatrix}, \quad (2.1)$$

Нехай  $N$  - кількість пакетів, які циркулюють в мережі; де  $p_{ij}$  – ймовірність пересилання пакета з  $i$ -го вузла в  $j$ -й вузол причём  $\sum_{j=0}^K p_{ij} = 1 \quad \forall i = \overline{0, K}$ .

Позначимо як  $\lambda_i = e_i \lambda_0$  інтенсивність потоку пакетів, що надходять в  $i$ -й вузол,

де  $e_i$  – передавальні коефіцієнти. Фізичний сенс інтенсивності потоку подій – це середнє число подій, що припадає на одиницю часу (число заявок в одиницю часу), розмірність -  $1/\text{час}$ .

Для стаціонарного режиму інтенсивність потоку, що входить у вузол, дорівнює інтенсивності вихідного [5]. Складемо систему рівнянь:

$$\lambda_i = \sum_{j=0}^K \lambda_j p_{ij} \quad \forall i = \overline{0, K}. \quad (2.2)$$

Ураховуючи, що  $\lambda_i = e_i \lambda_0$  и  $\lambda_j = e_j \lambda_0$ , отримуємо:

$$e_i = \sum_{j=0}^K e_j p_{ij}$$

або в розвернутому вигляді:

$$\begin{cases} -e_0 + p_{10}e_1 + p_{20}e_2 + \dots + p_{K0}e_K = 0 \\ p_{01}e_0 + (p_{11} - 1)e_1 + p_{21}e_2 + \dots + p_{K1}e_K = 0 \\ p_{02}e_0 + p_{12}e_1 + (p_{22} - 1)e_2 + \dots + p_{K2}e_K = 0 \\ \dots \\ p_{0K}e_0 + p_{1K}e_1 + p_{2K}e_2 + \dots + (p_{KK} - 1)e_K = 0 \end{cases} \quad (2.3)$$

Система лінійних рівнянь в матричній формі:  $P_1 E = 0$ , де матриця  $P_1$  отримана шляхом транспонування матриці і зменшення елементів головної діагоналі на 1:

$$P_1 = \begin{pmatrix} -1 & p_{10} & p_{20} & \dots & p_{K0} \\ p_{01} & p_{11} - 1 & p_{21} & \dots & p_{K1} \\ p_{02} & p_{12} & p_{22} - 1 & \dots & p_{K2} \\ \dots \\ p_{0K} & p_{1K} & p_{2K} & \dots & p_{KK} - 1 \end{pmatrix} \text{ и } E = \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ \dots \\ e_K \end{pmatrix}. \quad (2.4)$$

Застосувавши Метод Гауса, знайдемо пере давальні коефіцієнти  $e_1, e_2, \dots, e_K$ . Підставимо знайдені значення  $e_i$  в вихідну систему і переконаємося, що рівняння звертаються в вірні рівності.

Тепер можемо знайти інтенсивність потоку пакетів, що надходять в  $i$ -й вузол:  $\lambda_i = e_i \lambda_0 \quad \forall i = \overline{1, K}$ .

Розглянемо один з вузлів мережі для загального випадку (рис 2.2).

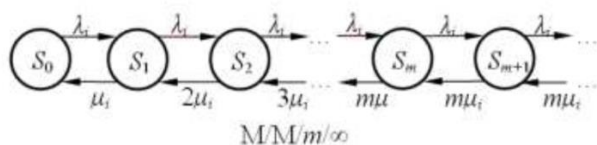


Рисунок 2.2 - Схема вузла мережі в загальному випадку

Припустимо, щовінмає  $m$  конвеєрів і необмежену чергу. Можливими станами цього вузла будуть  $\{S_k\} = \{S_0, S_1, S_2, \dots, S_m, S_{m+1}, \dots\}$ , де  $k$  – число пакетів

(обробляються або очікують) в вузлі. Процес блукання по цих станах буде Марківським процесом загибелі і розмноження.

Ймовірність знаходження вузла при стаціонарному режимі в стані  $S_k$  позначимо як  $P_i(k)$ . Виразимо вірогідність цих станів через  $P_i(0)$ .

Ймовірності знаходження вузла в різних станах			
$n$	$P_i(n)$	$r_i(n)$	$k_i(n)$
0	$P_i(0)$	0	0
1	$P_i(0) \frac{\lambda_i}{\mu_i}$	0	1
...	...	...	...
$m$	$P_i(0) \frac{\lambda_i^m}{m! \mu_i^m}$	0	$m$
$m+1$	$P_i(0) \frac{\lambda_i^{m+1}}{m! m \mu_i^{m+1}}$	1	$m$
$m+2$	$P_i(0) \frac{\lambda_i^{m+2}}{m! m^2 \mu_i^{m+2}}$	2	$m$
...	...	...	...

Вирази для чисельника виходять перемноженням інтенсивностей надходження пакетів (розмноження), для знаменника – інтенсивностей їх обслуговування (загибель). У загальному випадку:

$$P_i(n) = \frac{\lambda_i^n}{\mu_i^n \beta_i(n)} P_i(0) \quad \forall i = \overline{1, K}, \quad \beta_i(n) = \begin{cases} n!, & n \leq m \\ m! m^{n-m}, & n > m \end{cases} \quad (2.5)$$

де  $m$  – число конвеєрів в  $i$ -му вузлі.

З огляду на  $\sum_{n=0}^{\infty} p_i(n) = 1$ , отримаємо:

$$P_i(0) \left( \frac{\lambda_i}{\mu_i} + \frac{\lambda_i^2}{2! \mu_i^2} + \dots + \frac{\lambda_i^m}{m! \mu_i^m} + \frac{\lambda_i^{m+1}}{m! m \mu_i^{m+1}} + \frac{\lambda_i^{m+2}}{m! m^2 \mu_i^{m+2}} + \dots \right) = 1.$$

Введем позначення  $\chi_i = \frac{\lambda_i}{m \mu_i}$ ,  $\rho_i = \frac{\lambda_i}{\mu_i}$ ,

$$P_i(0) \left( 1 + \rho_i + \frac{\rho_i^2}{2!} + \dots + \frac{\rho_i^m}{m!} + \frac{\rho_i^{m+1}}{m! m} (1 + \chi_i + \chi_i^2 + \dots) \right) = 1$$

Сума нескінченної геометричної прогресії  $(1 + \chi_i + \chi_i^2 + \dots)$  буде кінцевою величиною за умови  $\chi_i < 1$ . Звідси випливає, що число конвеєрів  $m_i$  в  $i$ -му вузлі

$$m_i > \frac{\lambda_i}{\mu_i} \quad \forall i = \overline{1, K}$$

слід вибирати як мінімальне ціле число, яке задовольняє умові

, інакше мережа не впорається із заданим входять потоком пакетів. повертаючись до  $P_i(0)$ , отримуємо

$$P_i(0) = \left( \sum_{n=0}^{m_i} \frac{\rho_i^n}{n!} + \frac{\rho_i^{m_i+1}}{m_i! m_i (1 - \chi_i)} \right)^{-1} \quad \forall i = \overline{1, K}, \quad (2.6)$$

де  $m_i$  – число конвеєрів в  $i$ -му вузлі.

### 2.1.1.2 Моделювання замкнутої мережі

Нехай дана замкнута мережа, що складається з  $K$  систем масового обслуговування (рис. 2.3). У замкнутій МеМО циркулює фіксоване число заявок [4], а зовнішнє незалежне джерело відсутнє.  $PR = (p_{ji})$  – маршрутна матриця;  $m_i$  – кількість обробних конвеєрів в  $i$ -му вузлі;  $t_i$  – середній час обробки пакета в одному конвеєрі  $i$ -го вузла  $\forall i = \overline{1, K}$ .

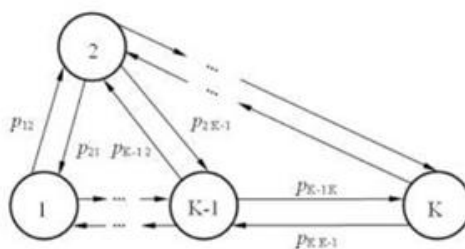


Рисунок 2.3 - Схема мережі в загальному випадку

Потрібно: знайти всі можливі стани мережі в цілому і стану кожного вузла окремо; розрахувати ймовірності цих станів  $P_i(n_i)$ ; знайти середнє число пакетів у вузлі  $L_i$ ; інтенсивності потоків надходять пакетів  $\lambda_i$ ; середній час перебування пакета в вузлі  $T_i$  і середній час циклу  $V_i$ .

В загальному випадку мережа задається стохастичною маршрутною матрицею:

$$P_R = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1K} \\ p_{21} & p_{22} & \dots & p_{2K} \\ \dots & \dots & \dots & \dots \\ p_{K1} & p_{K2} & \dots & p_{KK} \end{pmatrix}, \quad (2.7)$$

де  $p_{ij}$  – інтенсивність потоку пакетів, що надходять в позначимо як ймовірність пересилання пакета з  $i$ -го вузла в  $j$ -й вузол, причому  $\lambda_i = e_i \Lambda$ .

$$\sum_{j=1}^K p_{ij} = 1 \quad \forall i = \overline{1, K}.$$

де  $e_i$  – передаточні коефіцієнти.  $\lambda_i$  – фізична величина, яку можна виміряти.

Фізичний сенс інтенсивності потоку подій – це середнє число подій, що припадає на одиницю часу (число заявок в одиницю часу), розмірність –  $1/\text{час}$ . Для стаціонарного режиму інтенсивність потоку, що входить у вузол, дорівнює інтенсивності вихідного. Складемо систему рівнянь:

$$\lambda_j = \sum_{i=1}^K \lambda_i p_{ij} \quad \forall i = \overline{1, K} \quad (2.8)$$

Враховуючи, що  $\lambda_i = e_i \Lambda$  і  $\lambda_j = e_j \Lambda$ , скоротимо на  $\Lambda$ :

$$e_j = \sum_{i=1}^K e_i p_{ij} \quad (2.9)$$

або в розвернутом вигляді:

$$\begin{cases} (p_{11} - 1)e_1 + p_{21}e_2 + \dots + p_{K1}e_K = 0 \\ p_{12}e_1 + (p_{22} - 1)e_2 + \dots + p_{K2}e_K = 0 \\ \dots \\ p_{1K}e_1 + p_{2K}e_2 + \dots + (p_{KK} - 1)e_K = 0 \end{cases} \quad (2.10)$$

Система лінійних рівнянь (2.10) в матричній формі:

$P_1 E = 0$ , де матриця  $P_1$  отримана шляхом транспонування матриці (2.7) і зменшенням елементів головної діагоналі на 1:

$$P_1 = \begin{pmatrix} p_{11}-1 & p_{21} & \dots & p_{K1} \\ p_{12} & p_{22}-1 & \dots & p_{K2} \\ \dots & \dots & \dots & \dots \\ p_{1K} & p_{2K} & \dots & p_{KK}-1 \end{pmatrix} \text{ и } E = \begin{pmatrix} e_1 \\ e_2 \\ \dots \\ e_K \end{pmatrix}.$$

Щоб отримати єдине рішення, покладемо  $e_1 = 1$ . Тоді складемо 1-у строчку матриці  $P_1$  почленно з  $k$ -ю, де  $k = 2, K$  і отримаємо:

$$P_2 E = Q,$$

$$P_2 = \begin{pmatrix} p_{21} + p_{22} - 1 & \dots & p_{K1} + p_{K2} \\ p_{21} + p_{23} & \dots & p_{K1} + p_{K3} \\ \dots & \dots & \dots \\ p_{21} + p_{2K} & \dots & p_{K1} + p_{KK} - 1 \end{pmatrix}$$

$$K-1 \text{ и } Q = \begin{pmatrix} p_{11} - 1 + p_{12} \\ p_{11} - 1 + p_{13} \\ \dots \\ p_{11} - 1 + p_{1K} \end{pmatrix}.$$

Використовуємо Метод Гауса для вирішення, знайдемо передаточні коефіцієнти  $e_2, e_3, \dots, e_K$ .

Розглянемо вузли замкнутої мережі окремо (рис. 2.4).

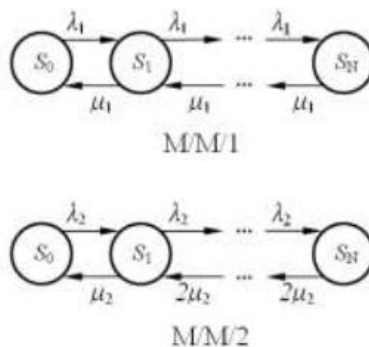


Рисунок 2.4 – Вузли замкнутої мережі

Позначимо як  $\mu_i = 1/\tau_i$  інтенсивність обробки пакетів в  $i$ -м вузлі, де  $\tau_i$  – середній час обробки пакета в  $i$ -му вузлі, розподілений по експоненційному закону:  $P(t) = \mu e^{-\mu t}, t \geq 0$ . Можливі стани 1-го вузла (1 конвеєр)  $\{S_k\} = \{S_0, S_1, S_2, \dots, S_N\}$ , де  $k$  – число пакетів в вузлі. Процес блукання по цих станах буде Марковським процесом загибелі і розмноження. Імовірність



знаходження 1-го вузла при стаціонарному режимі в стані  $S_k$  позначимо як  $P_1(k)$ . Виразимо вірогідність цих станів через  $P_1(0)$ :

$$P_1(1) = \frac{\lambda_1}{\mu_1} P_1(0), \quad P_1(2) = \frac{\lambda_1^2}{\mu_1^2} P_1(0), \quad P_1(3) = \frac{\lambda_1^3}{\mu_1^3} P_1(0), \quad \text{іт.д.}, \quad \sum_{n=0}^N P_1(n) = 1$$

Вираз для чисельника виходять перемноженням інтенсивностей надходження пакетів (розмноження), для знаменника – інтенсивностей їх обслуговування (загибель).  $P_1(0)$  поки залишається невідомим. Розглянемо 2-й вузол з 2-ма конвеєрами. Якщо у вузлі 2 конвеєра, то вузол починає обробляти пакети з подвоєною інтенсивністю (здіянні обидва конвеєра), коли в ньому знаходиться 2 пакети і більше.

$$P_2(1) = \frac{\lambda_2}{\mu_2} P_2(0), \quad P_2(2) = \frac{\lambda_2^2}{2\mu_2^2} P_2(0), \quad P_2(3) = \frac{\lambda_2^3}{4\mu_2^3} P_2(0)$$

І т.д., при цьому

$$\sum_{n=0}^N P_2(n) = 1.$$

В загальному випадку, імовірність знаходження  $i$ -го вузла в стані  $S_k$ :

$$P_i(n) = \frac{\lambda_i^n}{\mu_i^n \beta_i(n)} P_i(0) \quad \forall i = \overline{1, K},$$

$$\text{где } \beta_i(n) = \begin{cases} n!, & n \leq m \\ m! m^{n-m}, & n > m \end{cases}$$

Розглянемо всі можливі стани мережі

$$\vec{n} = (n_1, n_2, \dots, n_K) : n_1 + n_2 + \dots + n_K = N$$

де  $n_i$  – число пакетів у вузлі. Позначимо множину всіх станів мережі як  $S(N, K)$ .

По теоремі декомпозиції (Джексона), в стаціонарному режимі стан всієї мережі визначається станом її вузлів:

$$P(\vec{n}) = \frac{\prod_{i=1}^K P_i(n_i)}{\sum_{\vec{n}' \in S(N, K)} \prod_{i=1}^K P_i(n'_i)} \quad \forall \vec{n} = (n_1, n_2, \dots, n_K) \in S(N, K), \quad (2.11)$$

де  $P_i(n_i)$  – ймовірність знаходження  $i$ -го вузла в стані  $S_{ni}$ , а підсумовування проводиться по всій множині станів мережі  $S(N,K)$ .

Підставимо сюди вирази для  $P_i(n_i)$ , скоротимо дріб на  $P_1(0), P_2(0), \dots, P_K(0)$ .

Підставляючи сюди вирази для  $\lambda_i$  і враховуючи, що  $n_1 + n_2 + \dots + n_K = N$ , скоротимо дріб на  $AN$ . В результаті отримуємо:

$$P(\vec{n}) = \frac{\prod_{i=1}^K \frac{e_i^{n_i}}{\mu_i^{n_i} \beta_i(n_i)}}{\sum_{\vec{n}' \in S(N,K)} \prod_{i=1}^K \frac{e_i^{n'_i}}{\mu_i^{n'_i} \beta_i(n'_i)}} \quad \forall \vec{n} = (n_1, n_2, \dots, n_K) \in S(N,K) \quad (2.12)$$

Коли всі величини відомі, можна розрахувати ймовірності всіх станів мережі  $S(N, K)$ . Можна переконатися, що

$$\sum_{\vec{n} \in S(N,K)} P(\vec{n}) = 1.$$

Знайдемо також  $P_i(k)$  - всі ймовірності знаходження кожного  $i$ -го вузла в стані  $S_k$ :

$$P_i(k) = \sum_{\substack{\vec{n}' \in S(N,K): \\ n'_i = k}} P(\vec{n}') \quad \forall i = \overline{1, K}, \quad \forall k = \overline{0, N}.$$

### 2.1.2 Алгоритм Бузена

Наведені вище обчислення ймовірностей станів замкнутої мережі СМО за класичними розрахунковим формулами Гордона-Ньювелла виявилися не настільки вже складними. Однак в реальних обчислювальних системах і мережах число запитів і число систем обслуговування значно більше, ніж 4 запити і 3 СМО. Тому постає питання про те, як зробити процес обчислень простішим, уникнувши повного перебору всіх можливих станів замкнутої мережі. Дозволяє вирішити цю проблему алгоритм Бузена [5, 6].

Спочатку перетворимо формулу Гордона - Ньювела. Введемо функцію:

$$A_i(N_i) = \begin{cases} N_i!, m_i \geq N, \\ 1, m_i = 1 \\ N_i!, 1 < m_i < N, N_i \leq m_i \\ m_i! m_i^{N_i - m_i}, 1 < m_i < N, N_i > m_i \end{cases} \quad (2.13)$$

$$X_i = \frac{\alpha_i}{\mu_i}, i = 1, 2, \dots, n$$

Позначимо

Тоді формула може бути записана у вигляді:

$$P_{\vec{N}} = \frac{\prod_{i=1}^n P_{N_i, i}}{\sum_{\vec{N} \in S(n, N)} \prod_{i=1}^n P_{N_i, i}} = \frac{\prod_{i=1}^n X_i^{N_i} / A_i(N_i)}{\sum_{\vec{N} \in S(n, N)} \prod_{i=1}^n X_i^{N_i} / A_i(N_i)}. \quad (2.14)$$

$$g(m, M) = \sum_{\vec{N} \in S(m, M)} \prod_{i=1}^m X_i^{N_i} / A_i(N_i), \quad 1 \leq m \leq n, \quad 0 \leq M \leq N. \quad (2.15)$$

З фізичної точки зору функція  $g(m, M)$  відповідає частині вихідної замкнутої мережі, що містить  $i$  СМО і  $N$  запитів. Від вихідної замкнутої мережі СМО взяті  $m$  перших систем обслуговування, між якими циркулює  $M$  запитів. Неважко бачити, що  $g(m, 0) = 1, m = 1, 2, \dots, n$ .  $g(1, M) = X_1^M / A_1(M), M = 0, 1, 2, \dots, N$ .

Таким чином, якщо розглядати функцію  $g(m, M)$  як двовимірний масив, то в ньому однозначно визначаються нульовий стовпець і перший рядок шляхом простих обчислень. Потрібно знайти формулу, яка б пов'язала між собою елементи цього двовимірного масиву. Для цього зафіксуємо в останній,  $i$ -ой, системі обслуговування усічену мережу обслуговування до запитів. Тоді всі складові, що становлять суму, якої визначена функція  $g(m, M)$ , можуть бути розбиті на групи, для кожної з яких  $N_m = k$ . Кожна така група міститимуть співмножник  $X_m^k / A_m(k)$ , тому можна записати:

$$\begin{aligned} g(m, M) &= \sum_{\vec{N} \in S(m, M)} \prod_{i=1}^m X_i^{N_i} / A_i(N_i) = \\ &= \sum_{k=0}^M \sum_{\vec{N} \in S(m, M) \& N_m = k} \prod_{i=1}^m X_i^{N_i} / A_i(N_i) = \\ &= \sum_{k=0}^M \frac{X_m^k}{A_m(k)} \left( \sum_{\vec{N} \in S(m-1, M-k)} \prod_{i=1}^{m-1} X_i^{N_i} / A_i(N_i) \right) = \\ &= \sum_{k=0}^M \frac{X_m^k}{A_m(k)} g(m-1, M-k). \end{aligned} \quad (2.16)$$

Таким чином, отримана рекурентна формула, що дозволяє розрахувати значення масиву  $g(m, M)$  рядок за рядком:

$$g(m, M) = \sum_{k=0}^M \frac{X_m^k}{A_m(k)} g(m-1, M-k); m = 2, 3, \dots, n; M = 1, 2, \dots, N \quad (2.17)$$

Вираз разом з умовами заповнення першого шпальти і нульовий рядки двовимірного масиву  $g(m, M)$ , по суті справи, задає алгоритм Бузена.

Елемент в останньому рядку і останньому стовпчику масиву  $g(n, N) = G(N)$  дасть ту саму константу нормування, яка потрібна для обчислення ймовірностей знаходження замкнутої мережі в станах, визначених умовою  $\sum_{i=1}^n N_i = N$ , при цьому:

$$P_N = \text{Вер}\{N_1, \dots, N_n\} = \frac{1}{G(N)} \cdot \prod_{i=1}^n X_i^{N_i} / A_i(N_i), \quad (2.18)$$

Однак, як це було проілюстровано при розгляді прикладу розрахунку за класичною формулою Гордона-Ньювела, необхідно знайти ще й ймовірності того, що в окремо взятій системі обслуговування буде точно до запитів. Спочатку запишемо вираз для СМО з номером  $n$ :

$$P_k^{(n)} = P^{(n)}(N_n = k) = \sum_{\vec{N} \in S(n, N) \& N_n = k} P_{\vec{N}} \quad (2.19)$$

Для знаходження необхідно підсумувати ймовірності всіх тих станів замкнутої мережі, при яких в  $n$ -й останньої СМО. Якщо проводити розрахунок за формулою, то необхідний перебір станів замкнутої мережі обслуговування. Уникнути його можна, якщо записати вираз для шуканої ймовірності наступним чином:

$$\begin{aligned}
P_k^{(n)} &= \sum_{\tilde{N} \in S(n, N) \& N_n = k} \frac{X_n^k}{A_n(k)G(N)} \prod_{i=1}^{n-1} \frac{X_i^{N_i}}{A_i(N_i)} = \\
&= \frac{X_n^k}{A_n(k)G(N)} \sum_{\tilde{N} \in S(n-1, N-k)} \prod_{i=1}^{n-1} \frac{X_i^{N_i}}{A_i(N_i)} = \\
&= \frac{X_n^k}{A_n(k)G(N)} g(n-1, N-k)
\end{aligned}
\tag{2.20}$$

Знайдені елементи двовимірного масиву  $g(m, M)$  можуть бачь використані і для розрахунку ймовірностей знаходження заданого числа запитів в системах обслуговування замкнутої мережі.

## 2.2 Система управління водовідливом

### 2.2.1 Розробка структурної схеми інформаційних потоків дослідницької системи

Основними елементами насосної станції є приймальний резервуар, насосні агрегати і датчики рівнів. Виходячи з цього, модель об'єкту управління складатиметься з моделі приймального резервуару, моделей двох насосних агрегатів і моделей датчиків нижнього, середнього, верхнього і аварійного рівня.

Розглянемо модель насосного агрегату. Насосний агрегат структурно складається з двох елементів: асинхронного електродвигуна і насоса. Отже, модель насосного агрегату складатиметься з моделі асинхронного електродвигуна і моделі насоса. Оскільки управління насосним агрегатом повинно здійснюватися за допомогою частотного перетворювача, модель асинхронного електродвигуна повинна забезпечувати управління швидкістю. Структурна схема моделі асинхронного електродвигуна з частотним управлінням приведена на рис. 2.2.

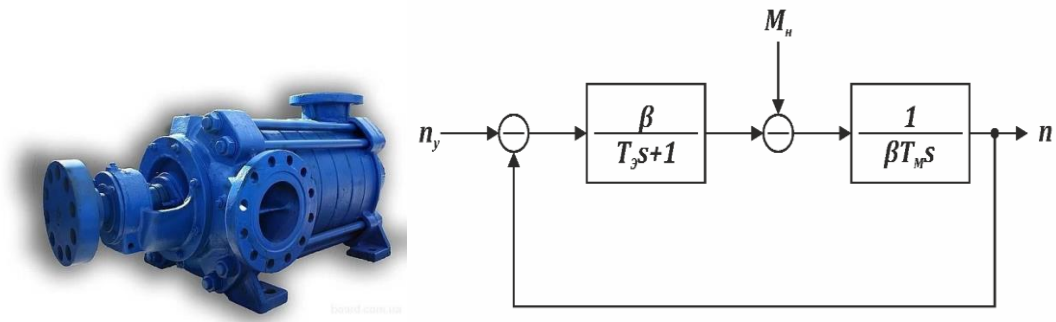


Рисунок 2.5 - Структурна схема моделі асинхронного електродвигуна

На вхід моделі подається уставка швидкості обертання валу електродвигуна. На виході моделі формується швидкість обертання електродвигуна. Аперіодична ланка відповідає за формування моменту на валу двигуна. На вихід аперіодичної ланки подається збурююча дія, що відповідає моменту навантаження. На підставі сформованого моменту на валу двигуна інтегруюча ланка розраховує швидкість обертання електродвигуна. У моделі використовуються модуль жорсткості  $\beta$ , електромагнітна постійна електродвигуна  $T_{\text{Э}}$  і електромеханічна постійна електродвигуна  $T_{\text{М}}$ . Розглянемо розрахунок параметрів моделі асинхронного електродвигуна, починаючи з модуля жорсткості:

$$\beta = \frac{2 \cdot M_k}{\omega_n \cdot s_k}, \quad (2.21)$$

де  $\beta$  - модуль жорсткості (Н·м·с/рад);

$M_k$  – критичний момент на валу електродвигуна (Н·м);

$\omega_n$  – швидкість обертання ротора (рад/с);

$s_k$  – критичне ковзання електродвигуна.

Критичний момент на валу електродвигуна визначається виходячи з номінального моменту:

$$M_k = \lambda \cdot M_n, \quad (2.22)$$

де  $\lambda$  - співвідношення критичного моменту до номінального моменту на валу електродвигуна;

$M_n$  – номінальний момент на валу (Н·м).

У свою чергу номінальний момент:

$$M_n = 9,55 \cdot \frac{P_H}{n_2}, \quad (2.23)$$

де  $P_H$  - номінальна потужність електродвигуна (Вт);

$n_2$  – частота обертання фактична (об/хв).

Швидкість обертання ротора електродвигуна розраховується на підставі фактичної частоти обертання:

$$\omega_n = \frac{2 \cdot \pi \cdot n_2}{60}. \quad (2.24)$$

Для розрахунку модуля жорсткості так само необхідно визначити критичне ковзання електродвигуна :

$$s_k = s_n \cdot (\lambda \pm \sqrt{\lambda^2 - 1}), \quad (2.25)$$

де  $s_n$  - номінальне ковзання.

Номінальне ковзання:

$$s_n = \frac{n_1 - n_2}{n_1}, \quad (2.26)$$

де  $n_1$  - частота обертання умовна (об/хв).

Електромагнітна постійна часу розраховується як:

$$T_{\text{Э}} = \frac{1}{\omega \cdot s_k}, \quad (2.27)$$

де  $T_{\text{Э}}$  - електромагнітна постійна часу (с);

$\omega$  – кутова частота напруги живлення статора (рад/с).

Кутова частота напруги живлення статора:

$$\omega = 2 \cdot \pi \cdot f, \quad (2.28)$$

де  $f$  - частота напруги живлячої мережі, Гц.

Електромеханічна постійна електродвигуна:

$$T_M = \frac{\omega \cdot s_k \cdot (J + J_n)}{2 \cdot p \cdot M_k}. \quad (2.29)$$

Розрахунок параметрів моделі асинхронного електродвигуна виконується виходячи з технічних характеристик електродвигуна (табл. 2.1). Результати розрахунку параметрів приведені в табл. 2.2.

Таблиця 2.1 - Технічні характеристики асинхронного електродвигуна насосної станції

№	Найменування параметра	Значення
1	Номінальна потужність, кВт	500
2	Число пар полюсів	6
3	Частота обертання умовна, об/хв	1500
4	Частота обертання фактична, об/хв	1450
5	$M_k/M_n$	2,3
6	$JJ, \text{кг}\cdot\text{м}^2$	0,0227
7	$J_n J_n, \text{кг}\cdot\text{м}^2$	0,0465

Таблиця 2.2 - Параметри моделі

№	Найменування параметра	Значення
1	Модуль жорсткості, Н·м·с/рад	10,27
2	Електромагнітна постійна часу, з	0,02185
3	Електромеханічна постійна електродвигуна, з	0,00697

### 2.2.2 Визначення моделі об'єкт управління

На підставі структури моделі і отриманих параметрів розроблена модель асинхронного електродвигуна насоса в графічному середовищі імітаційного моделювання Simulink, математичного пакету MATLAB, приведена на рис. 2.6.

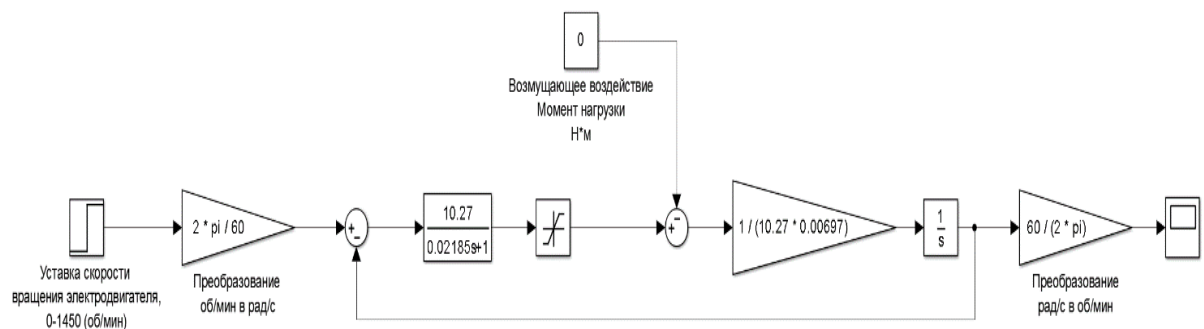


Рисунок 2.6 - Модель асинхронного електродвигуна насоса

На вхід моделі подається уставка швидкості обертання електродвигуна в діапазоні від 0 до 1 450 об/хв. Далі виконується перетворення уставки швидкості з про/мін в рад/с. Після чого модель розраховує значення моменту на валу двигуна. В якості збурюючої дії задається момент навантаження. На підставі



моменту розраховується швидкість обертання електродвигуна, яка перетвориться з рад/с в об/хв. Результат моделювання, отриманий при подання на вхід моделі у момент часу 1 з уставки рівної 1 450 об/хв приведений на рис. 2.7.

Модель насоса може бути представлена у вигляді аперіодичної ланки першого порядку. Коефіцієнт посилення ланки розраховується виходячи з витратно-напірної характеристики насоса (рис. 2.8), а постійна часу відповідає електромеханічною постійною часу електродвигуна.



Рисунок 2.7 - Результат моделювання асинхронного електродвигуна насоса

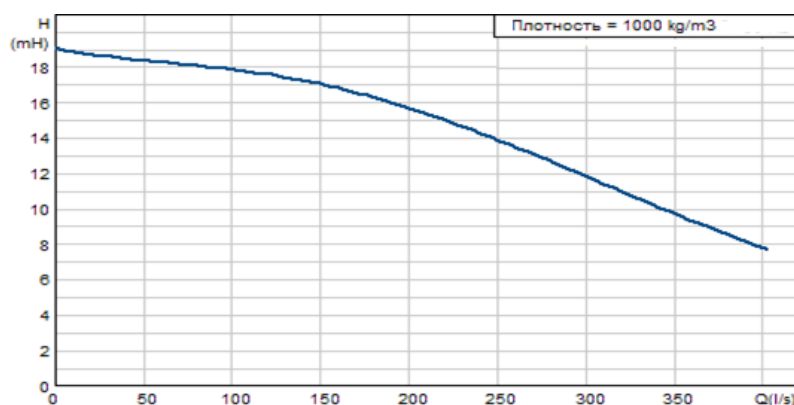


Рисунок 2.8 - Напірне-витратна характеристика насоса

Насос повинен забезпечувати підйом води на рівень, що відповідає перепаду висот напірного колектору 12 м. Відповідно до напірне-витратної характеристики при перепаді 12 м продуктивність насоса складає 300 л/с, що відповідає швидкості обертання асинхронного електродвигуна 1 450 об/хв. На

підставі цього, коефіцієнт посилення дорівнює відношенню продуктивності насоса до швидкості обертання електродвигуна, або 300/1 450. Електромеханічна постійна часу електродвигуна рівна 0,00697 с.

На підставі структури моделі і отриманих параметрів розроблена модель насоса в графічному середовищі імітаційного моделювання Simulink, математичного пакету MATLAB, приведена на рис. 2.5.

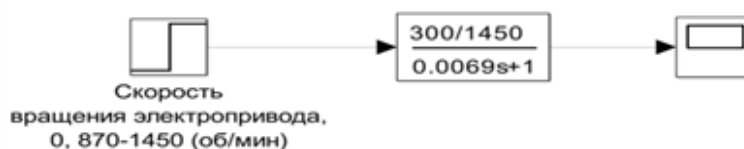


Рисунок 2.9 - Модель насоса

На вхід моделі задається швидкість обертання асинхронного електродвигуна 0 про/мін що відповідає вимкненому стану насоса і швидкість обертання від 870 до 1 450 об/хв при якій підтримується тиск 12 м водяного стовпа. На виході моделі формується витрата л/с. Результат моделювання, отриманий при подання на вхід моделі у момент часу 1 зі швидкості обертання електродвигуна 1 450 об/хв приведений на рис. 2.10.

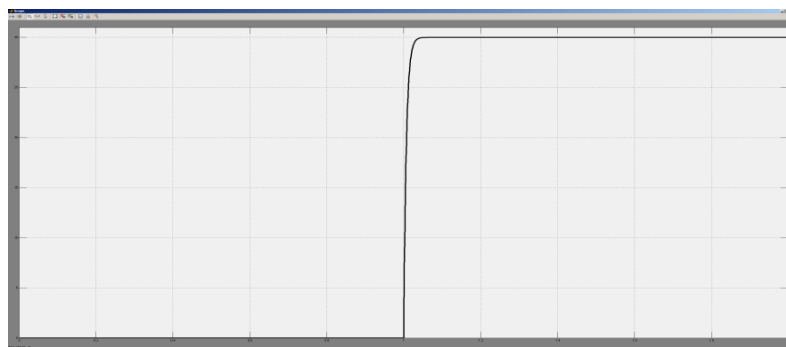


Рисунок 2.10 - Результат моделювання насоса

На підставі моделей асинхронного електродвигуна і насоса розроблена модель насосного агрегату. Для зручності модель асинхронного електродвигуна і модель насоса були представлені у вигляді блоків підсистем. Розроблена модель насосного агрегату приведена на рис. 2.11.

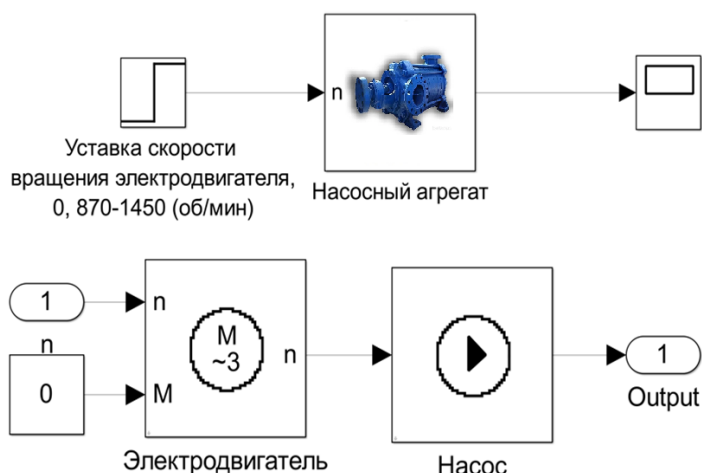


Рисунок 2.11 - Модель насосного агрегату

На вхід моделі задається уставка швидкості обертання асинхронного електродвигуна 0 про/мін що відповідає вимкненому стану і швидкість обертання від 870 до 1 450 об/хв при якій підтримується тиск 12 м водяного стовпа. Результат моделювання, отриманий при подання на вхід моделі у момент часу 1 с уставки швидкості обертання електродвигуна 1 450 об/хв приведений на рис. 2.12.

Модель приймального резервуару повинна приймати як вхідний параметр величину витрати в л/с, на підставі якої, повинен розраховуватися рівень вод в резервуарі в метрах. Таким чином, приймальний резервуар може бути представлений як інтегруючу ланку, на вхід якого подається зміна рівня води в м/с, а на виході формується рівень води в метрах. Перетворення л/с в м/с може бути виконано через об'єм резервуару.

Діаметр резервуару дорівнює 46 м, висота резервуару 10 м. Для того що б перерахувати витрата з л/с в м/с, необхідно спочатку перетворити витрату з л/с в  $\text{м}^3/\text{с}$  розділивши витрату на 1 000 л, після чого для перетворення  $\text{м}^3/\text{с}$  в м/с необхідно розділити отримане значення на площу резервуару. В якості обмежень інтегратора мають бути задані мінімальний рівень вод 0 м і максимальний рівень води 10 м.

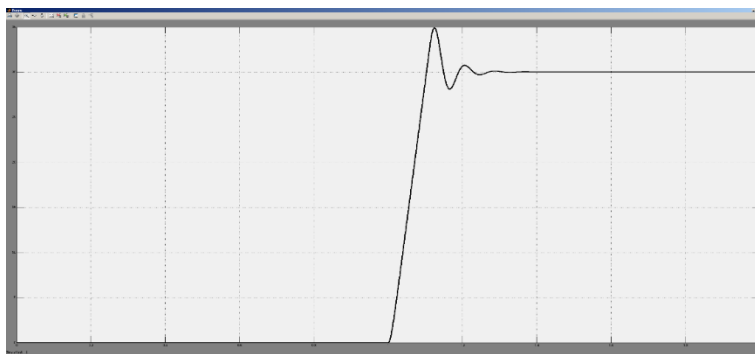


Рисунок 2.12 - Результат моделювання насосного агрегату

На підставі структури моделі і отриманих параметрів розроблена модель приймального резервуару в графічному середовищі імітаційного моделювання Simulink, математичного пакету MATLAB, приведена на рис. 2.13.

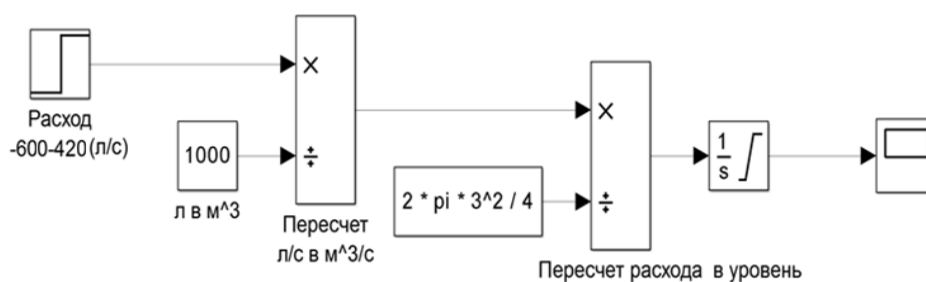


Рисунок 2.13 - Модель приймального резервуару

На вхід моделі задається витрата 0 л/с відповідає тому, що приплив води дорівнює продуктивності насосів, - 600 л/с відповідає максимальній продуктивності насосів, 420 л/с відповідає максимальному припливу води (при висоті водяного стовпа до 25 м). Результат моделювання, отриманий при подання на вхід моделі у момент часу 60 с витрати 100 л/с приведений на рис. 2.14.

Оскільки датчики рівня безпосередньо пов'язані з рівнем вод в приймальному резервуарі їх доцільно ввести в модель приймального резервуару. Кожен з датчиків повинен спрацьовувати по досягненню води відповідного рівня. На підставі цього розроблена уточнена модель приймального резервуару приведена на рис. 2.15.

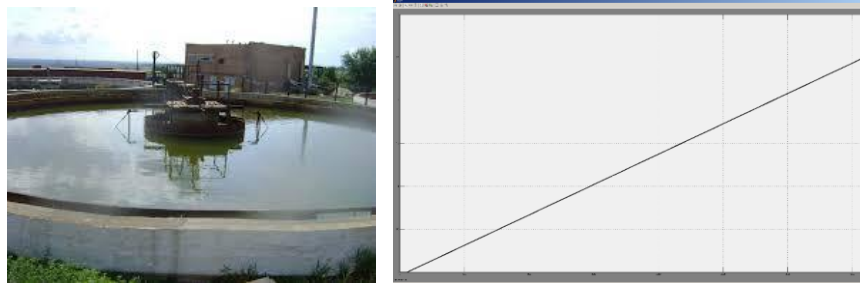


Рисунок 2.14 - Результат моделювання приймального резервуару

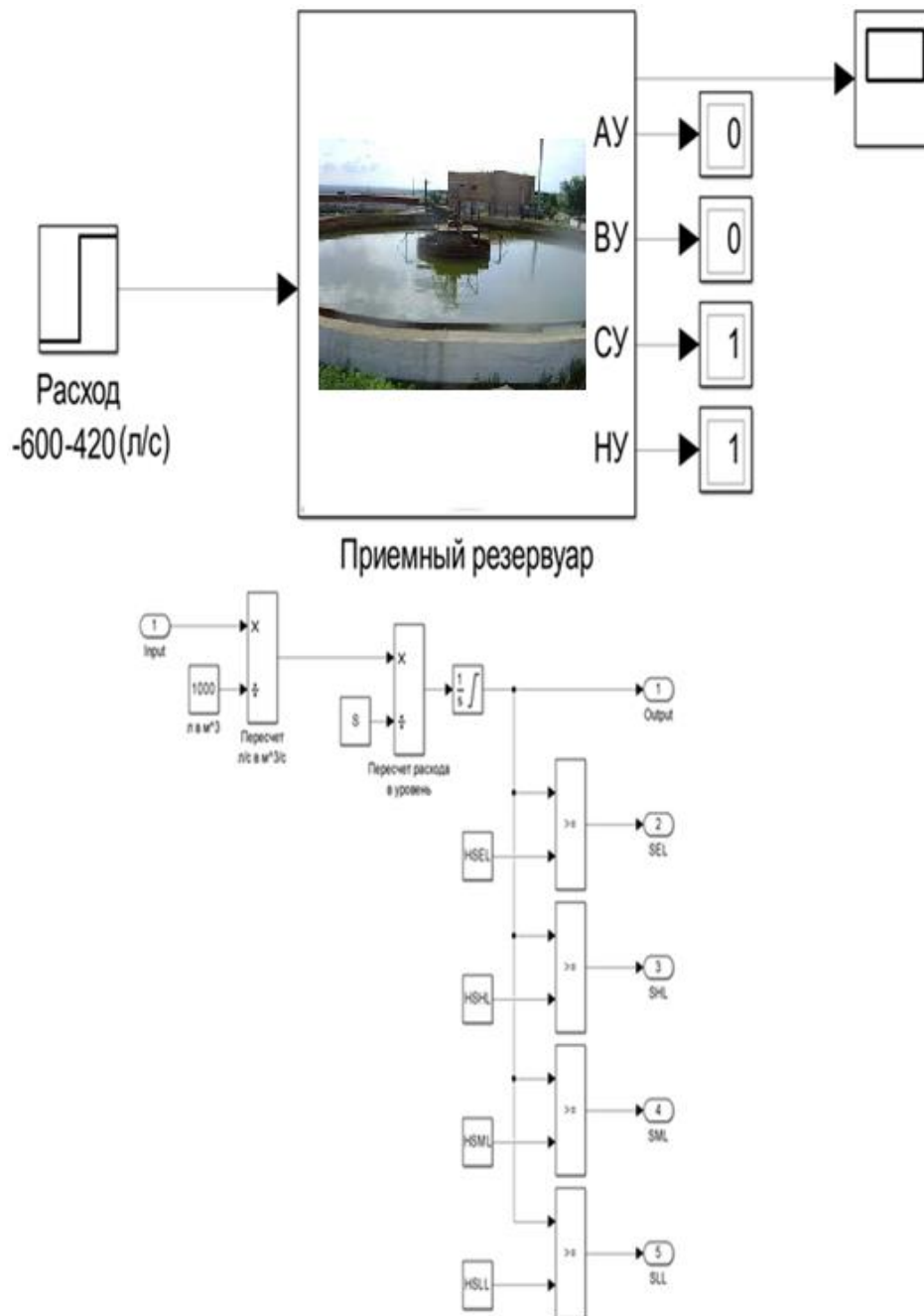


Рисунок 2.15 - Уточнена модель приймального резервуару

На підставі моделей приймального резервуару і насосних агрегатів розроблена модель насосної станції в графічному середовищі імітаційного моделювання Simulink, математичного пакету MATLAB, приведена на рис. 2.16.

Результат моделювання, отриманий при початковому рівні вод в приймальному резервуарі 5 м, постійній витраті води 250 л/с і включеному насосному агрегаті 1 на швидкості 1 450 об/хв приведені на рис. 2.17.

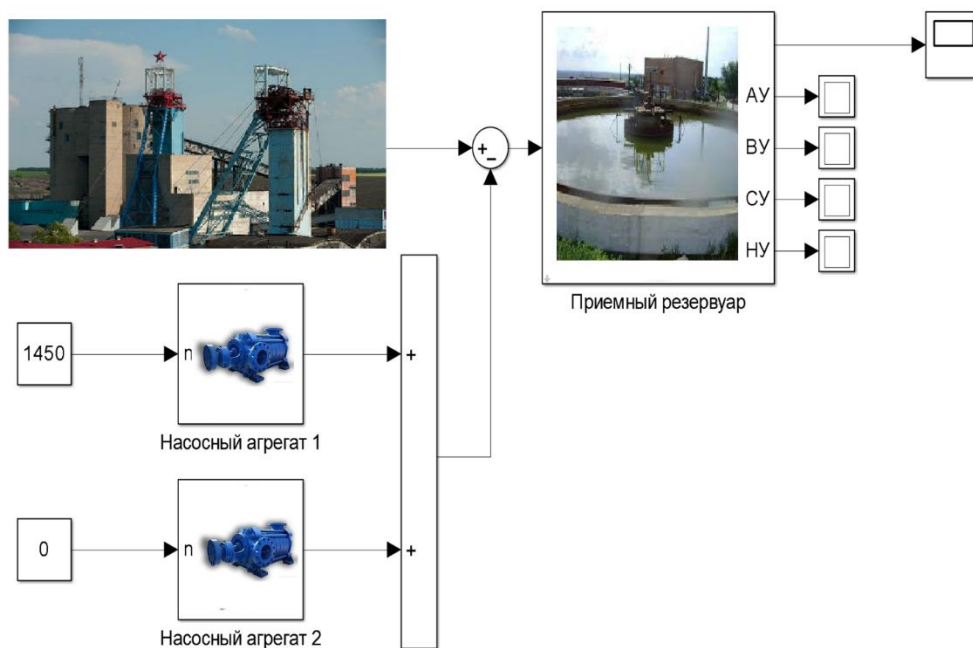


Рисунок 2.16 - Модель насосної станції

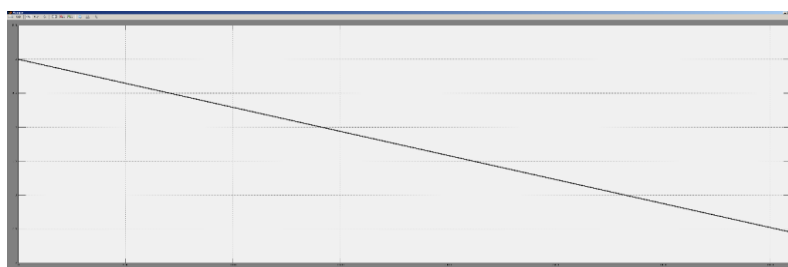


Рисунок 2.17 - Результат моделювання насосної станції

### 2.2.3 Критерії якості функціонування системи управління

Для насосної станції основним критерієм якості є мінімум енерговитрат, тобто мінімальне споживання електроенергії. Відомо, що мінімальне споживання електроенергії досягається тоді коли рівень в приймальному

резервуарі максимальний. Відповідно до технологічного процесу максимально допустимий рівень води в резервуарі відповідає датчику верхнього рівня. Таким чином, для мінімізації споживання електроенергії необхідно, щоб рівень води в резервуарі відповідав датчику верхнього рівня.

Витрата води впродовж дня є не рівномірною і частенько не дорівнює продуктивності насосних агрегатів. Для регулювання продуктивності насосних агрегатів використовується частотне управління. Проте максимальне значення ККД насосних агрегатів досягається тільки в діапазоні швидкостей від 870 до 1450 об/хв, що забезпечує витрату від 180 до 300 л/с. Таким чином, якщо витрата води менше 180 л/с, або від 350 до 480 л/с і при цьому вони знаходяться на верхньому рівні відповідно до технологічного процесу відкачування повинне проходити до нижнього рівня.

Виходячи з вищесказаного мінімізація витрат електроенергії може бути досягнута при знаходженні рівня води максимальний час на верхньому рівні, при швидкості роботи насосних агрегатів в діапазоні від 870 до 1450 об/хв, з дотриманням усіх вимог технологічного процесу.

Порівняння якості функціонування об'єктів при різних системах управління здійснюватиметься по інтегралу рівня вод. Чим більше значення інтеграла, тим, отже, більший час насосні агрегати пропрацювали при більш високому рівні води.

Виходячи з вищесказаного мінімізація витрат електроенергії може бути досягнута при знаходженні рівня вод максимальний час на верхньому рівні, при швидкості роботи насосних агрегатів в діапазоні від 870 до 1450 об/хв, з дотриманням усіх вимог технологічного процесу.

Порівняння якості функціонування об'єктів при різних системах управління здійснюватиметься по інтегралу рівня води. Чим більше значення інтеграла, тим, отже, більший час насосні агрегати пропрацювали при більш високому рівні води.

### 2.3 Висновки по розділу

На основі детального аналізу комп'ютерної системи підприємства виконано аналіз властивостей комп'ютерної мережі підприємства, для дослідження запропоновані сучасні методи, які використовуються для дослідження комп'ютерних систем як мереж масового обслуговування з урахуванням особливостей їх структури і інформаційного середовища.

Для реалізації моделі об'єкту управління розроблені модель приймального резервуару і модель насосного агрегату. На підставі отриманих моделей розроблена модель насосної станції. Отримана модель дозволяє задавати витрату води, швидкість обертання насосів, контролювати рівень води в резервуарі за допомогою датчиків нижнього, середнього, верхнього і аварійного рівнів. Сформульовані критерії якості функціонування об'єкту управління. Для насосної станції основним критерієм якості є мінімум витрат електроенергії. Для забезпечення мінімального споживання електроенергії на НС води повинні знаходитися на верхньому рівні, а швидкість роботи насосних агрегатів повинна знаходитися в діапазоні від 870 до 1 450 об/хв. При цьому повинні дотримуватися усі вимоги технологічного процесу.



## **3 СИНТЕЗ КОМП'ЮТЕРНОЇ СИСТЕМИ**

### **3.1 Комп'ютерна мережа підприємства**

#### **3.1.1 Структура комплексу технічних засобів комп'ютерної мережі підприємства**

Підприємство управління шахтою розподіляється на кілька підрозділів. В інженерному підрозділі знаходиться персонал, який обслуговує насосну станцію. Також в інженерному відділі знаходиться сервер ІС. В технічному відділі знаходяться системні адміністратори, які слідкують та обслуговують комп'ютерну мережу. Також в них є віддалений доступ до усіх мережевих пристроїв та серверів, у іншій підмережі знаходяться АРМ операторів, які слідкують за станом насосних агрегатів. Також при цьому є запасна машина АРМ, у випадку, якщо відкаже основна. В сусідній підмережі знаходиться сервер нижнього рівня, який збирає дані стану насосних агрегатів від ПЛК та передає їх на АРМ операторів. Інші три відділи складаються з бухгалтерів, відділу кадрів та відділу фінансів, які нараховують зарплату працівникам та розраховують бюджет підприємства. При цьому є вихід у мережу Інтернет для підключення віддалених співробітників у корпоративну мережу.

Етапи роботи у підприємстві залежить від переданих даних із одного підрозділу у інший. Залежить від безперебійної роботи мережі з доступ до серверів підприємства.

Вузькі місця системи: навантаження каналу у місті обробки оператором подій.

Модель мережі відповідає клієнт-серверному і розподіленому середовищу. Розподілені системи легко масштабуються, поступово системно розширюється, окремі частини системи повністю автономні, дають можливість спільно використовувати дані та пристрої. Переваги: підвищена надійність, полегшене тепловідведення.

На підприємстві представлена деревоподібна структура (рис. 3.1). Тип мережі ширококомовна маленького розміру з можливостями розширення.

Пропускна здатність мережі 100 Мбіт/сек. Типи підключення комп'ютерної мережі оптичний кабель до офісу, а у локальній мережі вита пара. Сегментація мережі представлена у вигляді підмереж. Мережа розподіляється на 7 підмереж згідно виконуваним функціям співробітників. Маршрутизація у мережі статична.

В проектуванні треба врахувати, що дані з технологічного процесу зберігаються на окремому сервері.

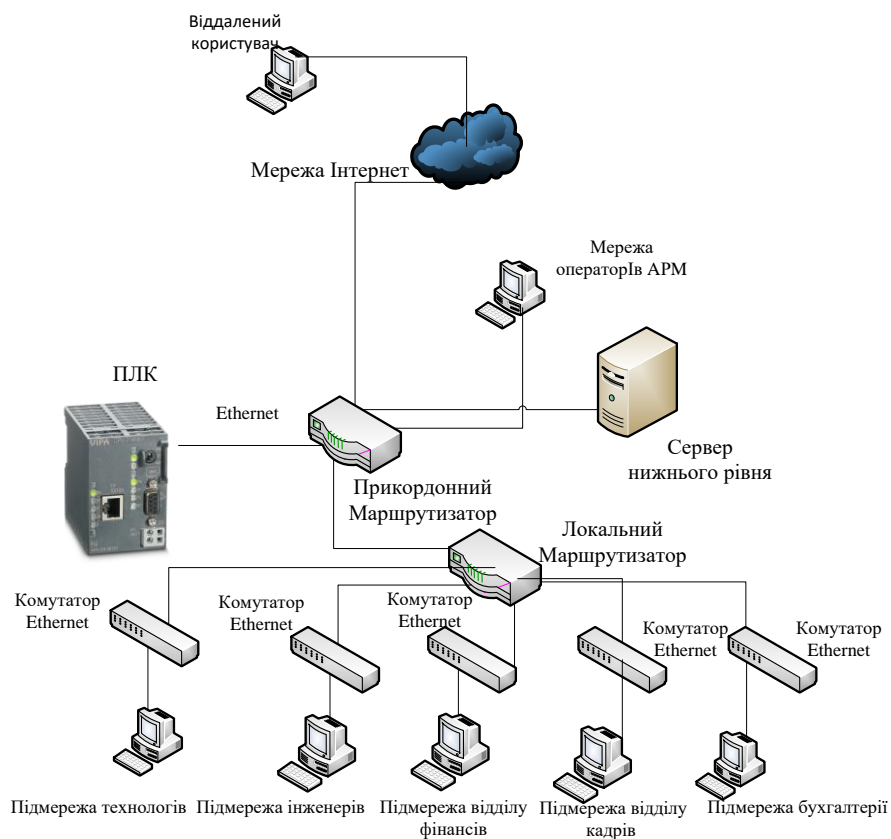


Рисунок 3.1 – Фізична топологія мережі

### 3.1.2 Особливості структурної схеми комп'ютерної мережі

Схема адресації була розроблена відповідно до ієрархічного принципу та фізичної топології мережі (рис. 3.1) проектування комп'ютерних мереж з діапазону приватних IP-адрес згідно з рекомендаціями RFC1918. У табл. 3.1 наведена схема адресації. У табл. 3.2 наведена адресація всіх пристроїв мережі. Відділи фінансів, бухгалтерії та відділу кадрів фізично під'єднані до єдиних мережних комутаторів і для їх сегментації застосовується метод VLAN на

комутаторах. У мережах VLAN використовується адресація кінцевих пристроїв за протоколом DHCP.

Таблиця 3.1-Схема адресації

Ім'я мережі	Кількість вузлів	Адреса мережі	Маска під мережі	Початкове значення	Кінцеве значення
LAN_ARM	6	192.168.0.0	255.255.255.248	192.168.0.1	192.168.0.6
LAN_Server	6	192.168.1.0	255.255.255.248	192.168.1.1	192.168.1.6
LAN_Tecnology	6	192.168.2.0	255.255.255.248	192.168.2.1	192.168.2.6
LAN_Engineers	6	192.168.3.0	255.255.255.248	192.168.3.1	192.168.3.6
LAN_Finance	30	192.168.4.0	255.255.255.224	192.168.4.1	192.168.4.30
LAN_Kadry	30	192.168.4.32	255.255.255.224	192.168.4.33	192.168.4.62
LAN_Buhgalters	30	192.168.4.64	255.255.255.224	192.168.4.65	192.168.4.94

Таблиця 3.2-Адресація пристроїв

Пристрій	Інтерфейс	IP-адреса	Маска	Шлюз	LAN
Прикордонний маршрутизатор	S 0/3/0	209.104.19.1	/30	-	209.104.19.0
	S 0/3/1	209.165.202.1	/30	-	209.165.202.0
	G 0/0	192.168.1.1	/29	-	192.168.1.0
	G 0/1	192.168.0.1	/29	-	192.168.0.0
Маршрутизатор локальної мережі	S 0/0/0	209.104.19.2	/30	-	209.104.19.0
	G 0/0	192.168.2.1	/29	-	192.168.2.0
	G 0/1	192.168.3.1	/29	-	192.168.3.0
	G 0/2.10	192.168.4.1	/27	-	192.168.4.0
	G 0/2.20	192.168.4.33	/27	-	192.168.4.32
	G 0/2.30	192.168.4.65	/27	-	192.168.4.64

### 3.1.3 Обґрунтування і вибір обладнання

Замовником було прийнято рішення використовувати обладнання компанії HP, Artline та Cisco. Нижче в табл. 3.3...3.6 приведені повні характеристики мережевого обладнання.

ПК Artline Business B27 v16 буде працювати в якості сервера нижнього рівня. Його технічні характеристики повністю відповідають вимогам, призначеним для його задач.

Таблиця 3.3 – Характеристики ПК Artline Business B27 v16

Встановлена операційна система	Windows 10 64bit
Процесор	Чотирьохядерний Intel Core i3-8100 ( 3.4 ГГц)
Технічні характеристики	Моноблок
Тип відеокарти й обсяг відеопам'яті	Intel UHD Graphics 630
Обсяг оперативної пам'яті	8 ГБ
Потужність БЖ	400 Вт
Попередньо встановлене ПЗ	Windows 10 64bit
Дисплей	Екран 27" (1920x1080) Full HD
Ціна	10000 грн

Маршрутизатор Cisco 2911-SEC буде пересилати трафік між підмережами організації. Основним акцент вибору даного пристрою був спрямований на те, щоб роутер міг підтримувати криптостійкі протоколи, такі як L2TP; IPsec, SSHv2 та шифрування DES, 3DES, AES-128. Також в ньому є функція налаштування списків контролю доступу (ACL). Маршрутизатор зображений на рис. 3.2.



Рисунок 3.2 – Маршрутизатор Cisco 2911-SEC

Таблиця 3.4 – Характеристики маршрутизатора Cisco 2911-SEC

Підтримка протоколів	L2TP; IPsec; PPPoE; DHCP
Функції безпеки	Шифрування DES, 3DES, AES-128
Інші функції	Продуктивність у постачанні 50Mbps з сервісами Розширення продуктивності за допомогою ліцензії Performance до 100Mbps Підтримка ACL, SSHv2 та інших механізмів безпеки
Додаткові характеристики	4 ГБ Flash; 4 ГБ DDR3 DRAM
Габарити та вага	4.455 x 36.957 x 29.464 см, 3.5 кг
Ціна	64 500 грн

Комутатор Cisco Catalyst 2960-X 24 буде поєднувати кінцеві пристрої в сегменти локальної мережі. В ньому є функція підтримки VLAN, що дозволяє

розділяти пристрої на окремі віртуальні локальні підмережі за для того, щоб обмежувати ширококомовний трафік та сегментувати мережу. Комутатор зображений на рис. 3.3.



Рисунок 3.3 – Коммутатор Cisco Catalyst 2960-X 24

Таблиця 3.5 – Характеристики коммутатора Cisco Catalyst 2960-X 24

Швидкість передачі даних (макс)	0.1 Гбит/с
Повнодуплексний режим	Підтримує
Кількість портів	24
Оперативна пам'ять	64 МБ
Флеш-пам'ять	32 МБ
Технічні особливості	VLAN, auto MDI/MDI-X, IGMP
Ціна	32 648 грн



Рисунок 3.4 – Сервер HP ProLiant DL360 Gen9

Таблиця 3.6 – Характеристики сервера HP ProLiant DL360 Gen9

Процесор	Восьмиядерний Intel Xeon E5-2620 v4 (2.1 - 3 ГГц)
Чипсет	Intel Xeon
Обсяг оперативної пам'яті	16 ГБ
Тип оперативної пам'яті	1 x 16 ГБ DDR4-DIMM, Registered (16 слотов) (ECC)
Жорсткий диск	2 x 300 ГБ, 12G SAS 10K, 2.5 " Всього слотів: 8 накопичувачів SAS / SATA / SSD малого форм-фактора
Інтерфейс HDD	SAS
Ціна	67 000 грн

### 3.1.4 Побудова моделі в середовищі Cisco Packet Tracer

На рис. 3.5 показана логічна топологія мережі.

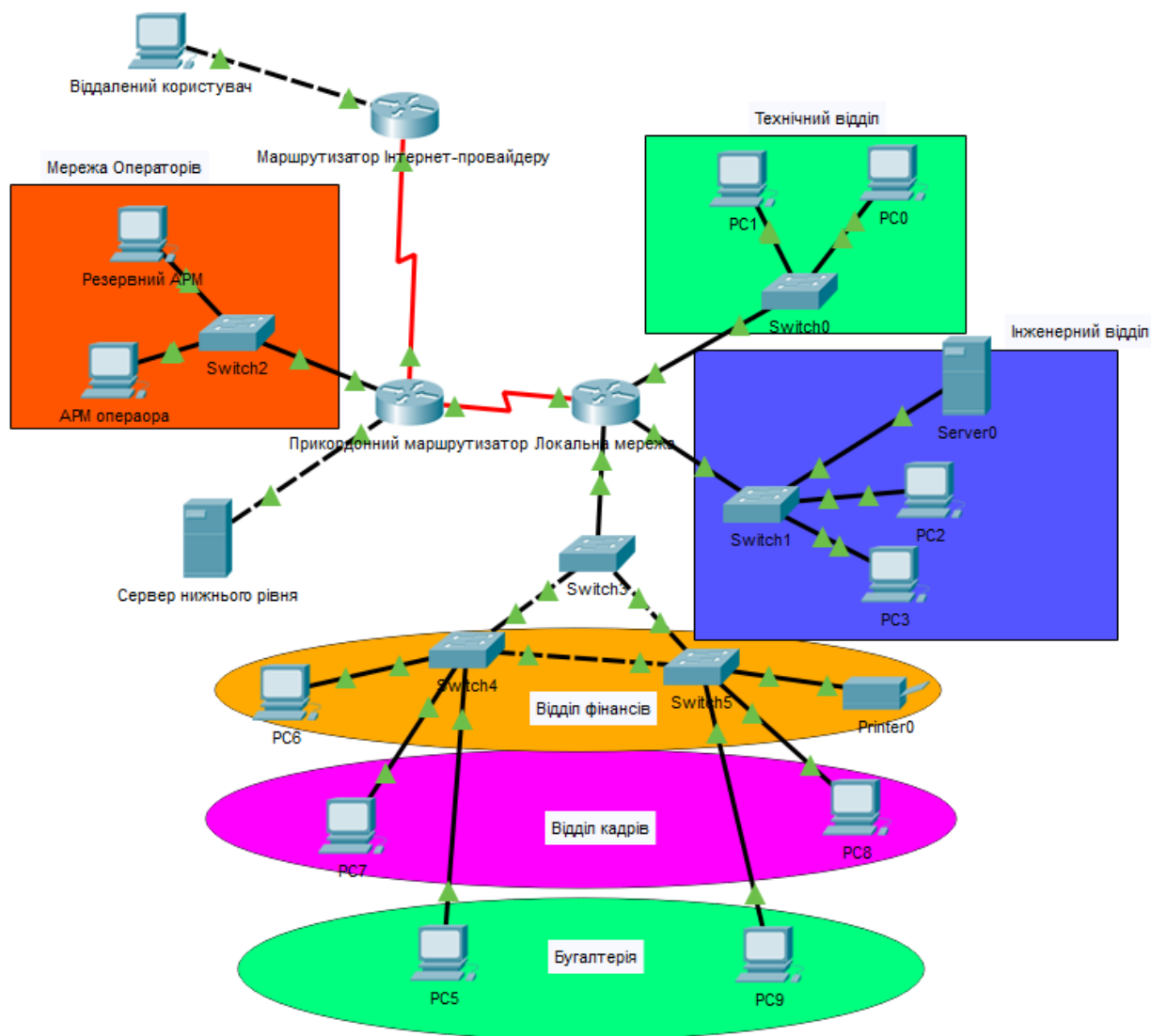


Рисунок 3.5 – Логічна топологія мережі

Для зв'язку з провайдером прикордонний розтер підключається до маршрутизатора провайдера за допомогою Serial- кабелю. Також до прикордонного роутера за допомогою кабелю GigabitEthernet приєднані дві мережі: в одній знаходиться сервер нижнього рівня, а в другій-мережа операторів. Маршрутизатор локальної мережі приєднаний до прикордонного роутера за допомогою Serial кабелю. До маршрутизатора локальної мережі за допомогою кабелю GigabitEthernet підключається комутатор відділів фінансів, бухгалтерії та відділу кадрів. На цьому комутаторі налаштований транковий порт, який пересилає vlan до інтерфейсу маршрутизатора, а той за допомогою технології router on a stick виконує маршрутизацію між віртуальними

локальними мережами. Також до роутера локальної мережі підключаються дві мережі: інженерний відділ та технічний відділ. Пересилання трафіку між маршрутизаторами відбувається за рахунок протоколу динамічної маршрутизації EIGRP.

### 3.1.3.1 Налаштування основних параметрів пристроїв

Перше, що необхідно при базовому налаштуванні мережі - це задання ім'я пристроям за правилом \_тип пристрою, тобто

```
Local-Router:
Router1>en
Router1#conf t
Router 1 #hostname Local-Router
```

Для безпечного доступу до пристроїв необхідно задати на всіх пристроях пароль до консолі і vty cisco, а також задати пароль до привілейованого режиму class та зашифрувати всі паролі, що зберігаються у відкритому виді. Також налаштовуємо банер, що буде відображатись при підключенні до маршрутизатора:

```
Local-Router (config)#line con 0
Local-Router (config-line)#password cisco
Local-Router (config-line)#login
Local-Router (config-line)#line vty 0 15
Local-Router (config-line)#password cisco
Local-Router (config-line)#login
Local-Router (config)#enable secret class
Local-Router (config)#service password-encryption
Local-Router (config)#banner motd
```

Отримати доступ до консолі маршрутизатора можна також за допомогою протокола Telnet, що передає дані у відкритому вигляді, або SSH захищеного з'єднання

При налаштуванні створимо користувача 123\_16ck\_1Savrickiy з паролем cisco. Для шифрування даних створюємо ключ RSA довжиною 1024 біт.

```
Local-Router (config)#username 123_16ck_1Savrickiy password cisco
Local-Router (config)#crypto key generate rsa 1024
Local-Router (config)#line vty 0 15
Local-Router (config-line)#login local
```

```
Local-Router (config-line)#transport input ssh
Local-Router (config-line)#exit
```

Налаштуємо IPv4-адреси відповідно до таблиці 2, наприклад, на

```
Local-Router (config)#int g0/1
Local-Router (config-if)#ip address 218.48.32.1 255.255.252.0
Local-Router (config-if)#no shutdown
Local-Router (config-if)#exit
```

На DCE-інтерфейсах маршрутизаторів встановлюємо значення тактової частоти - 128000

```
Local-Router (config)#int s0/1/0
Local-Router (config-if)#ip add 209.104.19.10 255.255.255.252
Local-Router (config-if)#clock rate 128000
```

### **3.1.3.2 Налаштування мереж VLAN, маршрутизації між VLAN і протоколу VTP**

Віртуальні локальні мережі (VLAN) - це окремі групи вузлів, трафік від яких на канальному рівні (тобто на основі адрес канального рівня) повністю ізольований від трафіку інших груп. Обмеження стосується також і широкомовного трафіку. В межах такої групи робота комутаторів здійснюється звичайним шляхом. Окремі комп'ютери можуть належати одночасно до кількох віртуальних локальних мереж. Так, приміром, можна налаштовувати сервери загального користування. Порти різних віртуальних мереж ніколи не з'єднуються між собою. У такому випадку зв'язок між мережами може здійснюватися лише через маршрутизатори.

Для створення аналогічної мережі без технології VLAN було б потрібно використати відповідну кількість окремих комутаторів, а для зміни логічної структури мережі довелося б також міняти її фізичну структуру.

При побудові мережі на двох чи більше комутаторах, вони з'єднуються між собою кабелем. Порти, через які здійснюється таке з'єднання, мають належати до відповідних віртуальних локальних мереж.

VLAN мають такі переваги:



-гнучкість впровадження. VLAN є ефективним способом угруповання мережевих користувачів в віртуальні робочі групи, незважаючи на їх фізичне розміщення в мережі;

-VLAN забезпечують можливість контролю широкомовних повідомлень, що збільшує смугу пропускання, доступну для користувача;

-VLAN дозволяють підвищити безпеку мережі, визначивши за допомогою фільтрів, напаштованих на комутаторі або маршрутизаторі, політику взаємодії користувачів з різних віртуальних мереж;

-спрощення керування мережею - VLAN робить керування мережею простішим, оскільки користувачі зі схожими вимогами до мережі будуть в одному VLAN. Коли додається новий комутатор в мережу, то вся політика безпеки, що застосована до конкретного VLAN, буде скопійована на нового користувача, що додається у VLAN;

-полегшення розподілу ресурсів - VLAN об'єднує користувачів, що мають подібні вимоги до мережі. Наприклад, є нелогічним об'єднати користувачів, що працюють з потоковим відео і користувачів, що працюють з великими файлами.

Налаштування VLAN на комутаторі Local-Sw1:

Створення VLAN та призначення їм імен:

```
Local-Sw1(config)#vlan 10
Local-Sw1 (config-vlan)#name buhgalters
Local-Sw1 (config-vlan)#vlan 20
Local-Sw1 (config-vlan)#name finance
Local-Sw1 (config-vlan)#vlan 30
Local-Sw1 (config-vlan)#name otdel_kadrov
Local-Sw1 (config-vlan)#exit
```

Призначення портів комутатора відповідним VLAN:

```
Local-Sw1 (config)#interface range f0/7-17
Local-Sw1 (config-if)#switchport mode access
Local-Sw1 (config-if)#switchport access vlan 10
Local-Sw1 (config-if)#interface range f0/18-22
Local-Sw1 (config-if)#switchport mode access
Local-Sw1 (config-if)#switchport access vlan 20
Local-Sw1 (config-if)#interface range f0/23-24
Local-Sw1 (config-if)#switchport mode access
```

```
Local-Sw1 (config-if)#switchport access vlan 30
Local-Sw1 (config-if)#exit
```

Налаштування транкового каналу:

```
Local-Sw1 (config)#interface f0/2
Local-Sw1 (config-if)#switchport mode trunk
Local-Sw1 (config-if)#exit
```

Налаштування маршрутизації між VLAN на маршрутизаторі:

```
Local-Router (config)#int g0/0.10
Local-Router(config-if)#encapsulation dot1q 10
Local-Router (config-if)# ip address 192.168.4.1 255.255.255.224
Local-Router (config-if)#int g0/0.20
Local-Router (config-if)#encapsulation dot1q 20
Local-Router (config-if)# ip address 192.168.4.33 255.255.255.224
Local-Router (config-if)#int g0/0.30
Local-Router (config-if)#encapsulation dot1q 30
Local-Router (config-if)# ip address 192.168.4.65 255.255.255.224
```

### 3.1.3.3 Налаштування протоколу STP

Spanning Tree Protocol (STP) - канальний протокол. Основним завданням STP є усунення петель в топології довільній мережі Ethernet, в якій є один або більше мережевих мостів, пов'язаних надлишковими сполуками. STP вирішує цю задачу, автоматично блокуючи з'єднання, які в даний момент для повної зв'язності комутаторів є надлишковими.

Необхідність усунення топологічних петель в мережі Ethernet впливає з того, що їх наявність в реальній мережі Ethernet з комутатором з високою ймовірністю призводить до нескінченних повторів передачі одних і тих же кадрів Ethernet одним і більш комутатором, чому пропускна здатність мережі виявляється майже повністю зайнята цими марними повторами ; в цих умовах, хоча формально мережу може продовжувати працювати, на практиці її продуктивність стає настільки низькою, що може виглядати як повна відмова мережі.

```
Local-Sw1 (config)# spanning-tree vlan 10, 20, 30 root primary
```

Вивід команди show-spanning tree vlan 10, зображена на рис. 3.6.

```

Local-Sw1#show spanning-tree vlan 10
VLAN0010
  Spanning tree enabled protocol rstp
  Root ID      Priority      24586
                Address      00D0.9725.19A7
                This bridge is the root
                Hello Time  2 sec  Max Age 20 sec  Forward
Delay 15 sec

  Bridge ID    Priority      24586 (priority 24576 sys-id-ext
10)
                Address      00D0.9725.19A7
                Hello Time  2 sec  Max Age 20 sec  Forward
Delay 15 sec
                Aging Time  20

Interface      Role Sts Cost          Prio.Nbr Type
-----
Fa0/1          Desg FWD 19           128.1   P2p
Fa0/2          Desg FWD 19           128.2   P2p
Gi0/1          Desg FWD 4           128.25  P2p

```

Рисунок 3.6 – Вивід команди

### 3.1.3.4 Налаштування маршрутизації

Маршрутизація — процес визначення маршруту прямування інформації між мережами. Маршрутизатор приймає рішення, що базується на IP-адресі отримувача пакету. Для того, щоб переслати пакет далі, всі пристрої на шляху слідування використовують IP-адресу отримувача. Для прийняття правильного рішення маршрутизатор має знати напрямки і маршрути до віддалених мереж.

Є два типи маршрутизації:

- статична маршрутизація — маршрути задаються вручну адміністратором.
- динамічна маршрутизація — маршрути обчислюються автоматично за допомогою протоколів динамічної маршрутизації— RIP, OSPF, E1GRP, BGP, HSRP та ін, які отримують інформацію про топологію і стан каналів зв'язку від інших маршрутизаторів у мережі.

Оскільки статичні маршрути конфігуруються вручну, будь-які зміни мережної топології вимагають участі адміністратора для додавання і видалення статичних маршрутів відповідно до змін. У великих мережах підтримка таблиць маршрутизації вручну може вимагати величезних витрат часу адміністратора. У невеликих мережах це робити легше. Статична маршрутизація не має можливості масштабування, яку має динамічна маршрутизація через додаткові вимоги до налаштування і втручання адміністратора. Але і у великих мережах часто конфігуруються статичні маршрути для спеціальних цілей у комбінації з протоколами динамічної маршрутизації, оскільки статична маршрутизація є стабільнішою і вимагає мінімум апаратних ресурсів маршрутизатора для обслуговування таблиці. Динамічні маршрути виставляються іншим чином. Після того, як адміністратор активізував і налаштував динамічну маршрутизацію за одним з протоколів, інформація про маршрути оновлюється автоматично в процесі маршрутизації після кожного отримання з мережі нової інформації про маршрути. Маршрутизатори обмінюються повідомленнями про зміни у топології мережі в процесі динамічної маршрутизації.

Відповідно до завдання у мережі використовується статична маршрутизація.

Вмикаємо EIGRP на маршрутизаторі, задаємо його унікальний ідентифікатор:

```
Border-Router (config)#ip route 192.168.4.0 255.255.252.0 209.104.19.2
Border-Router (config-router)#exit
Local-Router (config)#ip route 192.168.0.0 255.255.255.224 209.104.19.1
Налаштовуємо на ньому статичний маршрут та ручне підсумовування,
щоб протокол маршрутизації підсумовував тільки підмережі організації:
Border-Router (config-router)#ip route 0.0.0.0 0.0.0.0 209.165.202.2
```

### 3.1.3.5 Реалізація DHCP та NAT

DHCP (Dynamic Host Configuration Protocol — протокол динамічної конфігурації вузла) — це мережний протокол, що дозволяє комп'ютерам автоматично одержувати IP-адресу й інші параметри, необхідні для роботи в

комп'ютерна мережа|мережі TCP/IP. Для цього комп'ютер звертається до спеціального сервера (піл назвою сервера DHCP). Мережевий адміністратор може задати діапазон адрес, що розподіляють серед комп'ютерів. Це дозволяє уникнути ручного налаштування комп'ютерів мережі. Зменшує кількість помилок. Протокол DHCP використовується в більшості великих мереж TCP/IP.

DHCP є розширенням протоколу BOOTP, що використовувався раніше для забезпечення бездискової робочої станції|бездискових робочих станцій IP-адресами при їхньому завантаженні. DHCP зберігає зворотну сумісність з BOOTP.

Стандарт протоколу DHCP був прийнятий у жовтні 1993 року. Остання версія протоколу березень 1997 року. Нова версія DHCP, призначена для використання в середовищі IPv6, зветься DHCPv6.

Налаштування DHCP для мереж VLAN:

Виключаємо з пулу адреси, що не повинні роздаватися вузлам мережі:

```
Local-Router(config)#ip dhcp excluded-address 192.168.4.1 192.168.4.10
```

```
Local-Router (config)# ip dhcp excluded-address 192.168.4.33 192.168.4.43
```

```
Local-Router (config)# ip dhcp excluded-address 192.168.4.65 192.168.4.75
```

Створюємо DHCP-пули для підмереж VLAN, задаємо для цих мереж

та шлюз за замовчуванням:

```
Local-Router (config)#ip dhcp pool pool vlan10
```

```
Local-Router (dhcp-config)#network 192.168.4.0 255.255.255.224
```

```
Local-Router (dhcp-config)#default-router 192.168.4.1
```

```
Local-Router (dhcp-config)#ip dhcp pool pool vlan20
```

```
Local-Router (dhcp-config)#network 192.168.4.32 255.255.255.224
```

```
Local-Router (dhcp-config)#default-router 192.168.4.33
```

```
Local-Router (dhcp-config)#ip dhcp pool pool vlan30
```

```
Local-Router (dhcp-config)#network 192.168.4.64 255.255.255.224
```

```
Local-Router (dhcp-config)#default-router 192.168.4.65
```

NAT - це механізм в мережах TCP/IP, що дозволяє перетворювати IP-адреси транзитних пакетів. Завдяки NAT можна, використовуючи одну або кілька зовнішніх IP-адрес, виданих провайдером, підключити до мережі практично будь-яку кількість комп'ютерів. Більшість маршрутизаторів дозволяють виконувати трансляцію адрес, завдяки чому їх можна

використовувати для підключення невеликих мереж до інтернету, використовуючи одну зовнішню IP-адресу.

Налаштування NAT:

```
Border-Router (config)#ip nat inside source list 10 interface serial 0/3/1 overload
```

```
Border-Router (config)#ip route 0.0.0.0 0.0.0.0 209.165.202.2
```

```
Border-Router (config)#access-list 10 permit 192.168.0.0 0.0.255.255
```

Статистика NAT-трансляцій зображена на рис. 3.7

NAT Table for Прикордонний маршрутизатор				
Protocol	Inside Global	Inside Local	Outside Local	Outside Global
icmp	209.165.202.1:1028	192.168.2.3:1	10.122.34.10:1	10.122.34.10:1028
icmp	209.165.202.1:1024	192.168.3.2:1	10.122.34.10:1	10.122.34.10:1024
icmp	209.165.202.1:1026	192.168.3.4:1	10.122.34.10:1	10.122.34.10:1026
icmp	209.165.202.1:1027	192.168.4.3:1	10.122.34.10:1	10.122.34.10:1027
icmp	209.165.202.1:1	192.168.4.35:1	10.122.34.10:1	10.122.34.10:1
icmp	209.165.202.1:1025	192.168.4.67:1	10.122.34.10:1	10.122.34.10:1025

Рисунок 3.7 – Трансляції NAT

### 3.1.4 Розрахунок показників ефективності найбільшої під мережі

Вихідний трафік маршрутизується в лінію з пропускнуою здатністю 100 Мбіт/с.

Для того, щоб маршрутизатор не був перенасичений, швидкість надходження пакетів не повинна перевищувати швидкості їх відправлення.

Таким чином, загальне навантаження не повинно перевищувати

$$\mu_{\text{вих}} = 100\,000\,000 / (600 * 8) = 20\,834 \text{ пакетів/с}$$

Оскільки кожне джерело виробляє в середньому 125 пакетів/с, то ми обмежені приєднанням до маршрутизатора максимум:

$$N = 20834 / 125 = 167 \text{ джерел.}$$

Що задовольняє нашу мережу на 90 ПК.

Кожен з 90 ПК посилає потік заявок з інтенсивністю 125 кадрів/с.

Інтенсивність вихідного трафіку:

$$\lambda = 90 \cdot 125 = 11250 \text{ (пакетів/с)}$$

Коефіцієнт затримки:

$$\rho = \frac{\lambda}{\mu_{\text{вих}}} = \frac{11250}{20834} = 0,54$$

Коефіцієнт зайнятості маршрутизатора:

$$\frac{\rho}{1 - \rho} = \frac{0,54}{1 - 0,54} = 1,17$$

Середня затримка кадру, пов'язана з чергою M/M/1, дорівнює:

$$T = \frac{1}{(\mu - \lambda)} = \frac{1}{20834 - 11250} = 1,04 \text{ мкс}$$

Це значення менше необхідного значення  $\leq 5$  мс, що задовольняє вимогам.

Середня довжина черги:

$$\mathcal{L}_{\text{чер}} = \frac{\rho^2}{1 - \rho} = \frac{0,54^2}{1 - 0,54} = 0,63$$

Середній час перебування пакета в черзі

$$T_{\text{оч}} = \frac{\mathcal{L}_{\text{чер}}}{\lambda} = \frac{0,63}{11250} = 0,56 \text{ мкс}$$

Пропускна здатність каналу:

$$\lambda = \frac{\text{пропускна здатність}}{\text{довжина кадру}} = \frac{b}{l}$$

$$b = \lambda * l = 11250 * 600 * 8 = 54000000 \text{ біт/с} = 54 \text{ Мбіт/с}$$

Що задовольняє пропускну здатності вихідного каналу в 100 Мбіт/с.

## 3.2 Система управління водовідливом

### 3.2.1 Модель системи управління з безперервним датчиком рівня

Модель будь-якої системи управління включає моделі об'єкту управління і регулятора, при цьому в більшості випадків регулятор функціонує за значенням помилки, формованому як різниця між значенням уставки і зворотного зв'язку, званим у такому разі негативним зворотним зв'язком. Виходячи з цього, модель системи управління, що розробляється, повинна включати модель насосної станції і регулятора. Як регулятор буде використаний пропорційний регулятор,

оскільки сам об'єкт має інтегруючі властивості. Крім того як сигнал зворотного зв'язку буде вибраний рівень води в приймальному резервуарі. Надалі необхідно буде перейти від безперервного рівня до дискретного рівня, що формується датчиками аварійного, верхнього, середнього і нижнього рівня.

У системі управління присутні два насосні агрегати. Один агрегат повинен управлятися безперервно в діапазоні швидкостей 0,870...1 450 об/мін, а другий релейних в діапазоні швидкостей 0...1 450 об/хв. Оскільки безперервний регулятор може формувати дію, що управляє, тільки в безперервному діапазоні 870...1 450 об/мін, необхідно розробити елементи моделі, які дозволять виконувати включення і виключення насосних агрегатів.

Розглянемо модель релейного управління насосним агрегатом. Ця модель повинна залежно від рівня води виконувати включення і виключення насосного агрегату. Тобто якщо рівень води досягає рівня 8 м, насосний агрегат має бути включений, а якщо рівень вод опускається нижче 5 м, насосний агрегат має бути вимкнений. На підставі цього розроблена відповідна модель, приведена на рис. 3.8.

Модель релейного управління насосним агрегатом приймає вхідний сигнал рівень води, функція "EnablePump" реалізує формування сигналів включення і виключення насосного агрегату. У разі включення насосного агрегату на вихід подається максимальна швидкість "nMaximum" відповідна 1 450 об/мін, у разі виключення подається швидкість 0 об/хв.



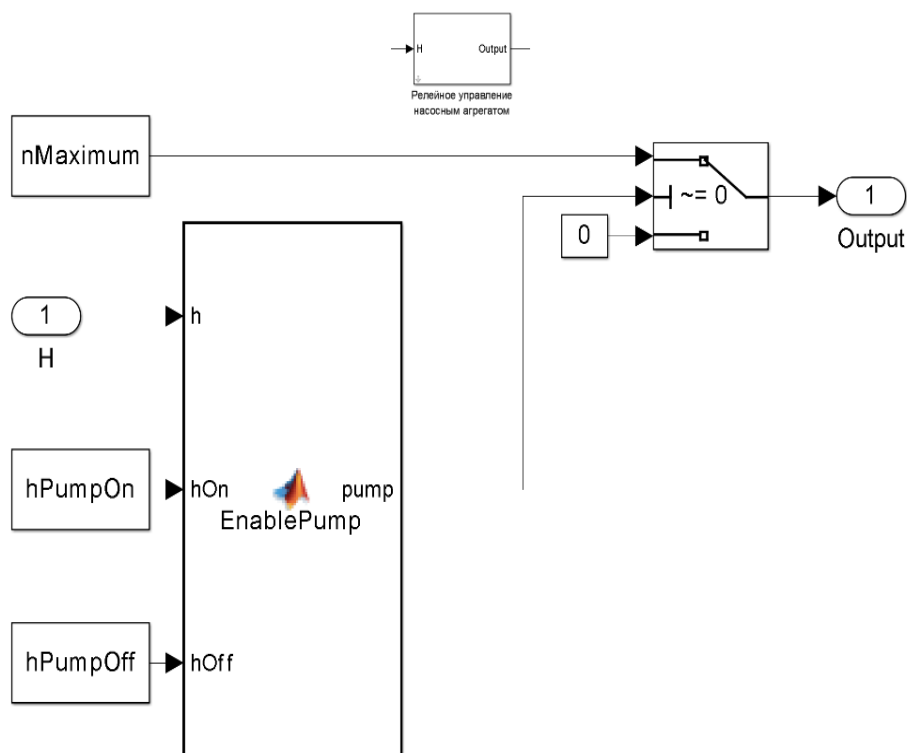


Рисунок 3.8 – Модель релейного управління насосним агрегатом

Функція "EnablePump" реалізована на мові MATLAB:

```
function pump = EnablePump(h, hOn, hOff)

% h - Уровень воды
% hOn - Уровень включения насосного агрегата
% hOff - Уровень выключения насосного агрегата
% pump - Сигнал включения/выключения насосного агрегата

persistent statePump;
persistent time;
persistent hPrevious;
persistent hState;
if isempty(statePump)
запуске
    statePump = 0;
    time = 0;
    hPrevious = 0;
    hState = 0.0;
end

if statePump == 0
    timeMaximum = 60.0 / 0.1;
    0.1 c
    if time >= timeMaximum
        if hState < 0 && h >= hOn
            statePump = 1;
        else
            statePump = 0;
        end
        hPrevious = 0;
        hState = 0;
        time = 0;
    else
        if hPrevious == 0
            hPrevious = h;
        end
        hState = hState + (hPrevious - h);
        time = time + 1;
    end
end
else
    % Локальные статические переменные
    % Состояние насосного агрегата
    % Счетчик времени
    % Уровень воды в предыдущем цикле
    % Разность уровней воды
    % Инициализация переменных при первом

    % Насосный агрегат выключен
    % Количество циклов = 60 с / время цикла

    % Время вышло
    % Подъем воды

    % Опускание воды

    % Нет предыдущего значения

    % Расчет разностей уровней воды
    % Увеличение счетчика времени

    % Насосный агрегат включен
```

```

    if h <= hOff                                % Уровень воды ниже уровня выключения
        statePump = 0;
    end
end
pump = statePump;                               % Управление насосным агрегатом

```

Функція "EnablePump" визначає досягнення води заданого рівня включення / виключення насоса. Якщо насос вимкнений "if statePump == 0", відбувається запуск лічильника часу "time = 0;". При кожному виклику функції виконується розрахунок різниці між поточним значенням рівня води і попереднім значенням "(hPrevious - h)". Отримана різниця підсумовується з різницями, отриманими в усіх попередніх циклах "hState = hState + (hPrevious - h);". Якщо після закінчення часу рівного 60 секунд "if time >= timeMaximum" різниця є негативною, відбувається зменшення рівня вод, якщо позитивною відбувається збільшення рівня вод. Якщо відбувається збільшення рівня вод і рівень вище заданого відбувається включення насосного агрегату "if hState < 0 && h >= hOn". При зниженні рівня води нижче заданого рівня "if h <= hOff" відбувається виключення насосного агрегату.

Розглянемо модель безперервного управління насосним агрегатом (рис. 3.9). Модель приймає як вхідний сигнал рівень води, функція "EnablePump" реалізує формування сигналів включення і виключення насосного агрегату. У разі включення насосного агрегату на вихід подається значення швидкості в діапазоні 870...1 450 об/мін у разі виключення 0 об/хв.

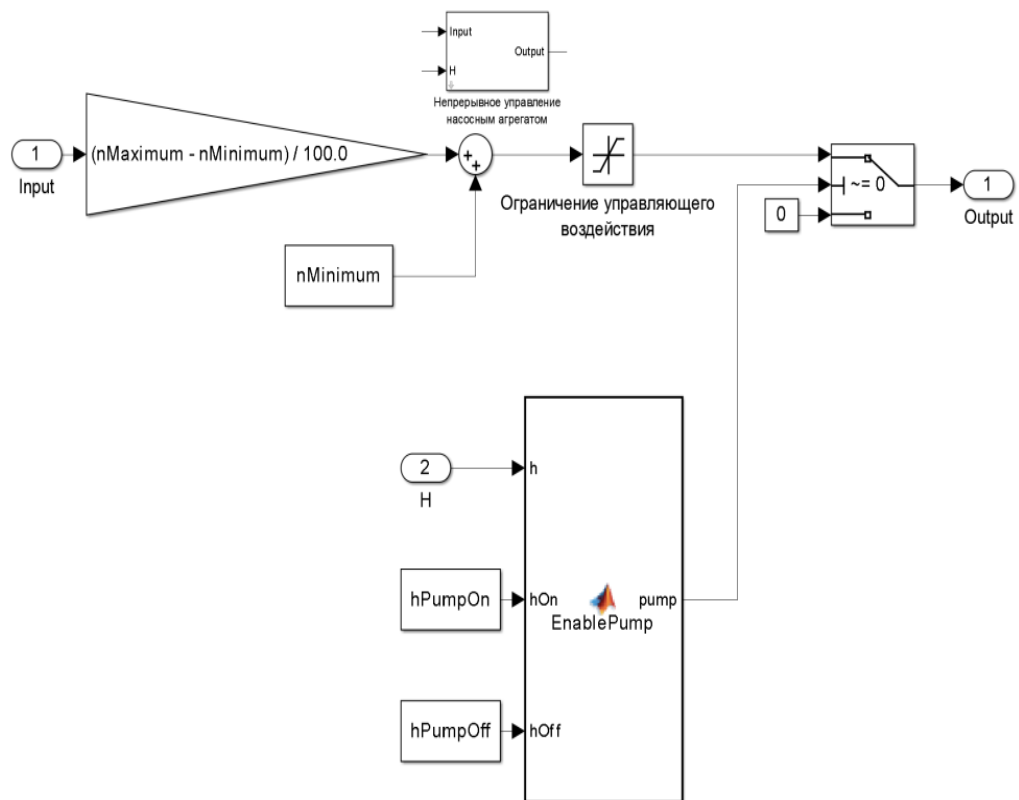


Рисунок 3.9 – Модель безперервного управління насосним агрегатом

Як вхідне значення моделі "Input" використовується нормоване значення сигналу управління від 0 до 100 %. Це значення перераховується в значення швидкості від 870 до 1 450 об/мін, після чого обмежується і подається на вихід моделі.

На базі моделей безперервного і релейного управління насосними агрегатами і раніше розглянутої структури, розроблена система управління насосною станцією в графічному середовищі імітаційного моделювання Simulink математичного пакету MATLAB (рис. 3.10).

На вхід моделі задається уставка рівня вод для підтримки насосним агрегатом з безперервним управлінням. Пропорційний регулятор розраховує значення сигналу, що управляє, яке обмежується в діапазоні від 0 до 100 %. На підставі сигналу, що управляє, і рівня вод модель безперервного управління насосним агрегатом виконує включення/виключення насосного агрегату і завдання уставки (870...1 450 об/мін) його швидкості обертання. Модель релейного управління насосним агрегатом виконує аналогічні функції, як

уставки використовується швидкість обертання 1 450 обо/хв. Результат роботи моделі при витраті вод 350 л/с приведений на рис. 3.11.

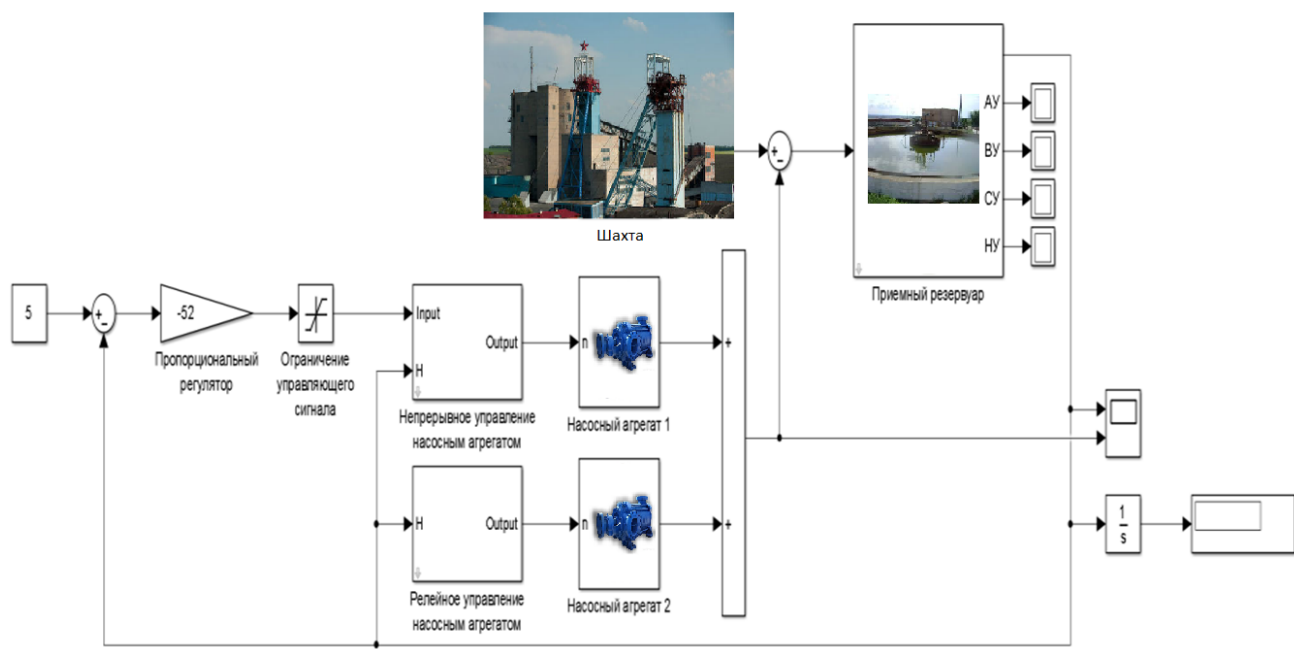


Рисунок 3.10 – Модель системи управління

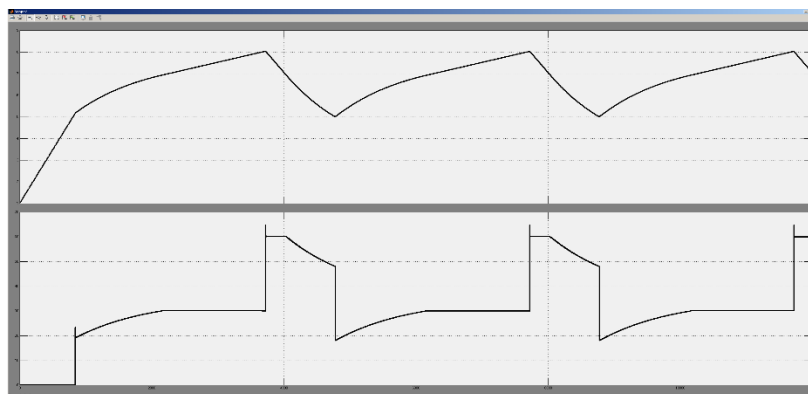


Рисунок 3.11 – Результат моделювання системи управління

Як видно за результатами моделювання насосний агрегат 1 включається через 60 с після досягнення рівнем вод 5 м Швидкістю насосного агрегату 1 управляє пропорційний регулятор і, отже, витрата. Із збільшенням значення помилки витрата зростає до тих пір, поки не досягне максимуму. Після того, як витрата досягне максимуму рівень вод повільно підвищується до рівня 8 м По досягненню рівня 8 м включається насосний агрегат 2. Після того, як рівень вод

опуститься до 8 м насосний агрегат 2 вимикається і залишається включеним тільки насосний агрегат 1.

З метою оцінки впливу коефіцієнта посилення на якість роботи системи управління виконано ряд моделювань результати, яких приведені в табл. 3.7.

Таблиця 3.7 - Результати моделювання

Витрата вод, л/с	Інтеграл рівня води / 104				
	Коефіцієнт посилення				
	-18	-35	-52	-70	-88
150	4,319	4,319	4,319	4,319	4,319
250	7,772	7,310	6,869	6,612	6,452
300	7,582	7,499	7,726	7,275	6,995
350	7,636	7,645	7,830	7,704	7,632
400	7,624	7,624	7,655	7,635	7,614

Оцінка виконувалася за інтегральним критерієм. Чим більше значення інтеграла рівня вод тим, отже, більший інтервал часу рівень вод був підвищений, і як наслідок цього тим менше були витрати електричної енергії. Як видно з результатів дослідження при витраті вод відповідному функціонуванню тільки насосного агрегату з частотним управлінням (150, 250, 300 л/с) найкращі результати досягаються при коефіцієнті посилення - 18, а при роботі двох насосних агрегатів при коефіцієнті посилення - 52. Таким чином, для досягнення зменшення витрат електроенергії досить використовувати два різні коефіцієнти посилення, які можуть задаватися виходячи з добового графіку.

При роботі двох насосних агрегатів, насосний агрегат з частотним управлінням функціонує на підставі дії формованого, що управляє, пропорційним регулятором. При цьому очевидно збільшується час знаходження на нижчому рівні в районі 5 м У зв'язку з цим необхідно перевірити, як зміняться витрати електричної енергії при максимально швидкому зменшенні рівня вод в резервуарі від 8 до 5 м, з метою виключення насосного агрегату 2.

З метою проведення дослідження в модель безперервного управління насосним агрегатом необхідно додати вхід визначення включення насосного агрегату з релейним управлінням і при включенні насосного агрегату 2 задавати

уставку швидкості насосного агрегату 1 рівною максимальною. Уточнена модель безперервного управління насосним агрегатом приведена на рис. 3.12.

На підставі уточненої моделі безперервного управління насосним агрегатом розроблена уточнена модель системи управління приведена на рис. 3.13.

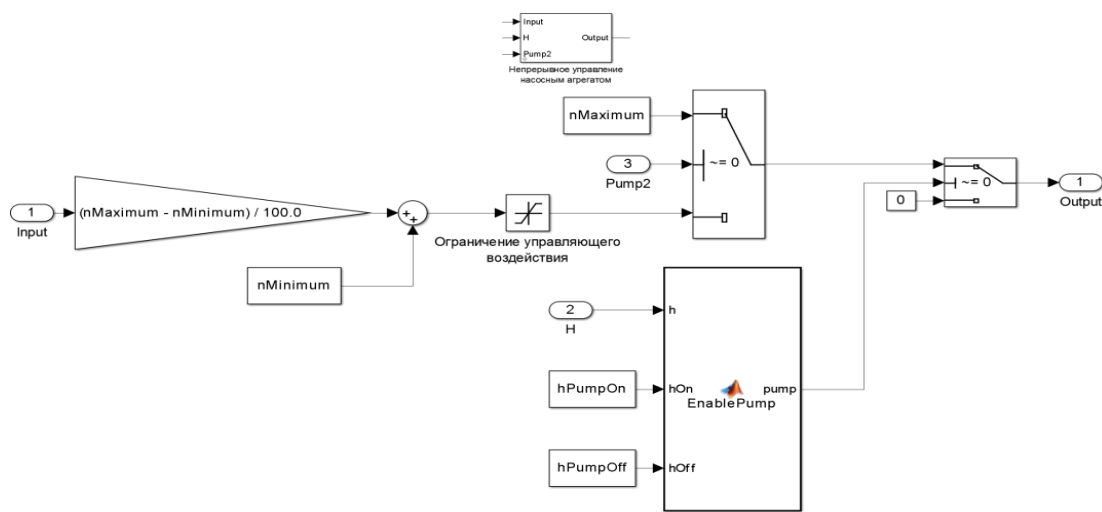


Рисунок 3.12 – нена модель безперервного управління насосним агрегатом

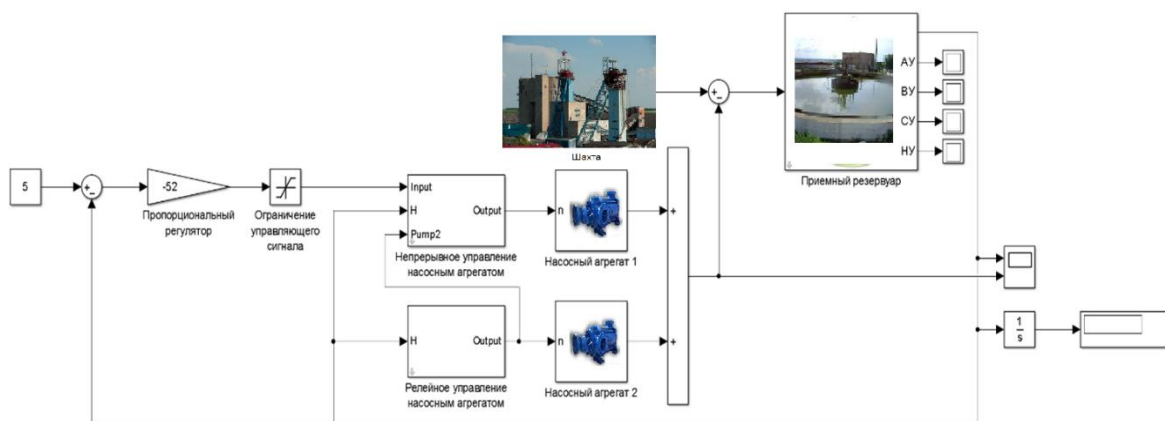


Рисунок 3.13 – Уточнена модель системи управління

З метою оцінки впливу коефіцієнта посилення на якість роботи системи управління з уточненою структурою виконано ряд моделювань результати, яких приведені в табл. 3.8.

Таблиця 3.8 - Результати моделювання

Витрата вод, л/з	Інтеграл рівня вод / 104				
	Коефіцієнт посилення				
	-18	-35	-52	-70	-88
350	7,619	7,766	7,861	7,770	7,677
400	7,700	7,694	7,704	7,748	7,734

Порівнявши результати моделювання моделей при роботі двох насосних агрегатів можна зробити висновок, що зменшення часу роботи двох насосів приводить до збільшення часу знаходження рівня вод. При цьому необхідно відмітити, що при швидкості обертання асинхронного електродвигуна наближеної до максимальної ККД, як самого електродвигуна, так і насоса так само наближено до максимального. На підставі цього можна зробити висновок, що для зменшення витрат електричній енергії при досягненні води верхнього рівня обидва насоси повинні функціонувати на максимальній швидкості.

### 3.2.2 Модель системи управління з дискретним датчиком рівня

У раніше використаних моделях вод визначався по безперервному датчику, проте на насосних станціях встановлюються дискретні датчики аварійного, верхнього, середнього і нижнього рівня на рівні 2, 5, 8 і 9 м відповідно. Як показали результати дослідження, уточнена модель системи управління забезпечує кращі показники зменшення витрат на електричну енергію. Виходячи з цього, безперервне визначення рівня вод необхідно здійснювати тільки коли працює насосний агрегат з безперервним управлінням, при підйомі вод від датчика середнього рівня до датчика верхнього рівня. Виходячи з цього, необхідно виконати непряму оцінку швидкості зміни рівня вод в резервуарі на середньому рівні. Для цього додамо над датчиком середнього рівня датчик швидкості, який розміщуватиметься вище на 0,1 м.

Для реалізації поставленого завдання модель релейного управління насосним агрегатом була переведена на роботу по датчиках рівня. Уточнена модель релейного управління насосним агрегатом приведена на рис. 3.14.

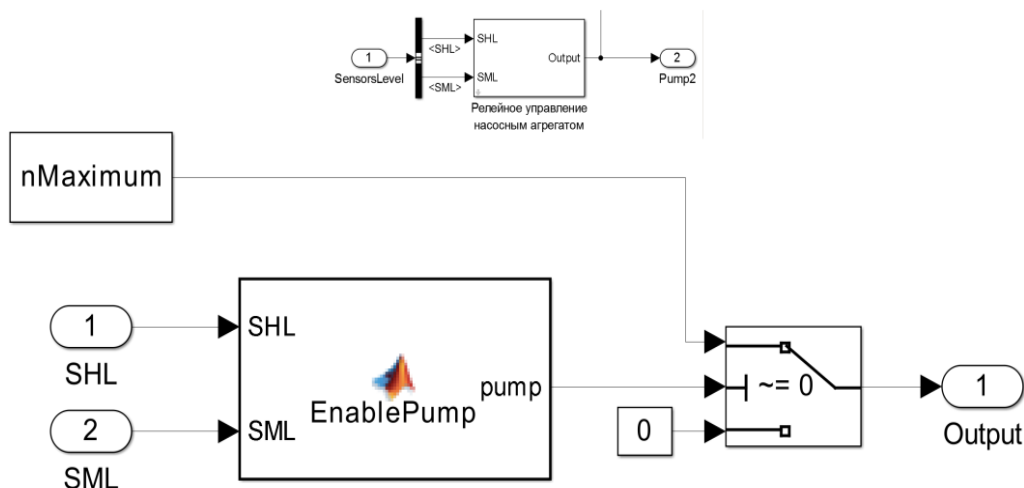


Рисунок 3.14 – Уточнена модель релейного управління насосним агрегатом

Функція "EnablePump" для уточненої моделі реалізована на мові MATLAB:

```
function pump = EnablePump(SHL, SML)

% SHL - Датчик верхнего уровня
% SML - Датчик среднего уровня
% pump - Сигнал включения/выключения насосного агрегата

persistent statePump;
if isempty(statePump)
    запуске
        statePump = 0;
    end

if statePump == 0
    if SHL == 1
        statePump = 1;
    end
else
    if SML == 0
        statePump = 0;
    end
end

pump = statePump;
```

% Локальные статические переменные  
 % Состояние насосного агрегата  
 % Инициализация переменных при первом  
  
 % Насосный агрегат выключен  
 % Воды достигли верхнего уровня  
 % Включение насосного агрегата  
  
 % Насосный агрегат включен  
 % Воды опустились ниже среднего уровня  
 % Выключение насосного агрегата  
  
 % Управление насосным агрегатом

Уточнена модель безперервного управління насосним агрегатом приведена на рис. 3.15.



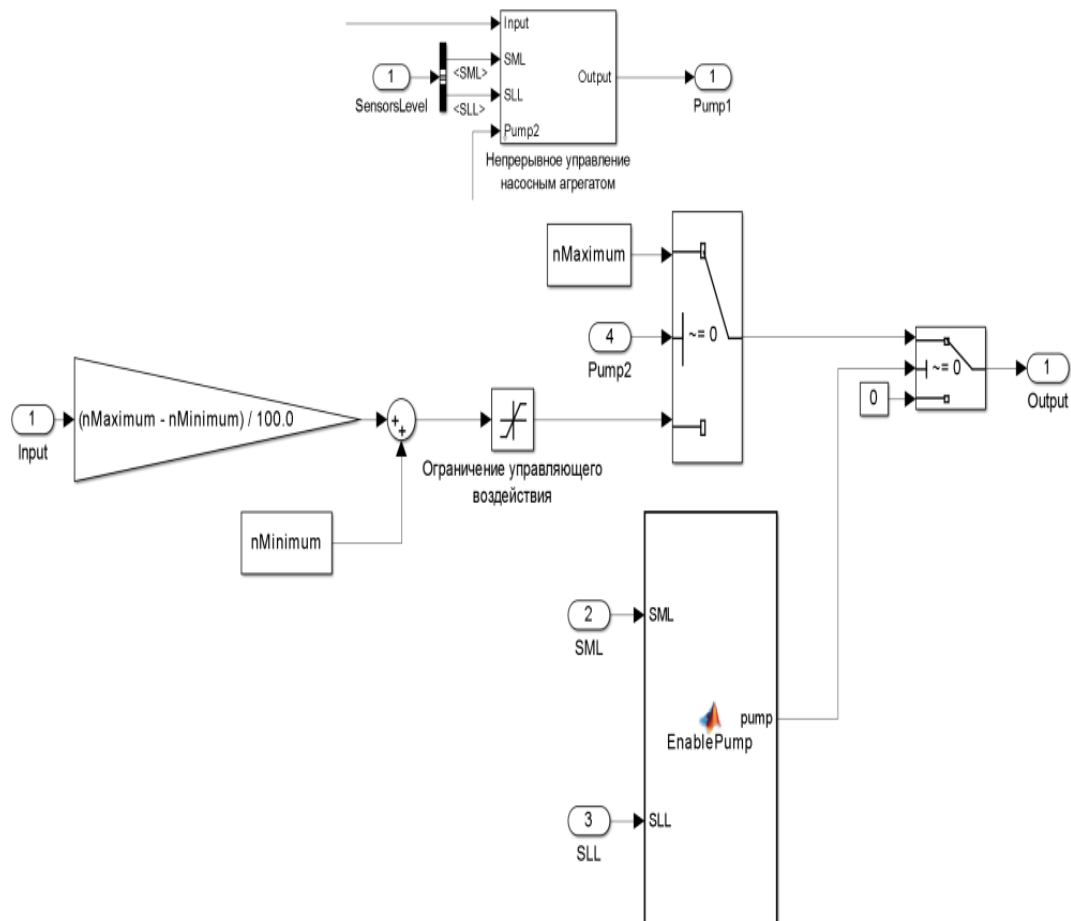


Рисунок 3.15 – Уточнена модель безперервного управління насосним агрегатом

Функція "EnablePump" для уточненої моделі реалізована на мові MATLAB:

```
function pump = EnablePump(SML, SLL)

% SML - Датчик среднего уровня
% SLL - Датчик нижнего уровня
% pump - Сигнал включения/выключения насосного агрегата

persistent statePump;
if isempty(statePump)
    % Локальные статические переменные
    % Состояние насосного агрегата
    % Инициализация переменных при
    % первом запуске
    statePump = 0;
end

if statePump == 0
    % Насосный агрегат выключен
    % Воды достигли среднего уровня
    % Включение насосного агрегата
    if SML == 1
        statePump = 1;
    end
else
    % Насосный агрегат включен
    % Воды опустились ниже нижнего
    % уровня
    % Выключение насосного агрегата
    if SLL == 0
        statePump = 0;
    end
end

pump = statePump;

% Управление насосным агрегатом
```

Датчик рівня реалізований як функція "SensorH" приймає як вхідні параметри рівень розміщення датчика верхнього "HSHL" і середнього рівня "HSML", а так само датчика швидкості "HSSL". Крім того у функцію передаються стани цих датчиків "SHL", "SML", "SSL". Функція "SensorH" реалізована на мові MATLAB :

```
function h = SensorH(HSHL, HSSL, HSML, SHL, SSL, SML)

% HSHL - Уровень размещения датчика верхнего уровня
% HSSL - Уровень размещения датчика скорости
% HSML - Уровень размещения датчика среднего уровня
% SHL - Состояние датчика верхнего уровня
% SSL - Состояние датчика скорости
% SML - Состояние датчика нижнего уровня

% Локальные статические переменные
persistent initialize; % Инициализация
persistent time; % Счетчик времени
persistent hState; % Уровень воды
persistent hStateUpdate; % Приращение уровня воды
if isempty(initialize) % Инициализация переменных при первом запуске
    initialize = 0;
    time = 0;
    hState = 0;
    hStateUpdate = 0;
end

if SSL == 0 % Води ниже датчика скорости
    if SML == 0 % Води ниже датчика среднего уровня
        initialize = 0;
        time = 0;
        hState = 0;
        hStateUpdate = 0;
    elseif initialize == 0 % Води выше датчика среднего уровня
        initialize = 1; % Запуск инициализации
    else % Увеличение счетчика времени
        time = time + 1;
    end
else % Води выше датчика скорости
    if initialize == 1 % Инициализация
        hState = HSSL; % Уровень воды соответствует уровню датчика скорости
        hStateUpdate = (HSSL - HSML) / time; % Расчет приращения
        initialize = 0;
    elseif SHL == 1 % Води выше датчика верхнего уровня
        hState = HSHL; % Уровень воды соответствует уровню датчика верхнего уровня
        hStateUpdate = 0;
    % Система включилась когда води были выше датчика скорости, но ниже датчика верхнего уровня
    elseif hState == 0 % Принять уровень води по середине между датчиками среднего и верхнего уровня
        hState = (HSHL - HSML) / 2.0 + HSML;
    elseif hState ~= 0 % Води выше датчика среднего уровня, но ниже датчика верхнего уровня
        hState = hState + hStateUpdate; % Расчет уровня води
        % Уровень води выше уровня размещения датчика верхнего уровня
        if hState > HSHL
            hState = HSHL; % Уровень води соответствует уровню датчика верхнего уровня
        end
    else % Води были выше датчика верхнего уровня, а теперь опустились ниже его
        hState = 0;
    end
end
end

h = hState; % Уровень води
```

Якщо рівень вод нижче за датчик швидкості "if SSL == 0" і нижче за датчик середнього рівня виконується "if SML == 0" скидання усіх значень. По

досягненню води датчика середнього рівня "elseif initialize == 0" виконується установка прапора ініціалізації "initialize = 1;" а на кожному наступному циклі збільшення лічильника часу "time = time + 1;".

По досягненню води датчика швидкості виконується ініціалізація "if initialize == 1", під час якої як поточне значення рівня вод задається рівень розміщення датчика швидкості "hState = HSSL;" а як приріст рівня вод "hStateUpdate = (HSSL - HSML) / time;" швидкість приросту, що розраховується як різниця рівнів датчика середнього рівня і датчика швидкості ділена на лічильник часу. На наступному циклі "elseif hState ~= 0" відбувається збільшення рівня "hState = hState + hStateUpdate;".

Якщо рівень вод вище за датчик верхнього рівня "if hState > HSHL", то рівень вод задається рівним рівню розміщення цього датчика "hState = HSHL;".

Розроблена модель пропорційного регулятора з використанням датчика рівня приведена на рис. 3.16.

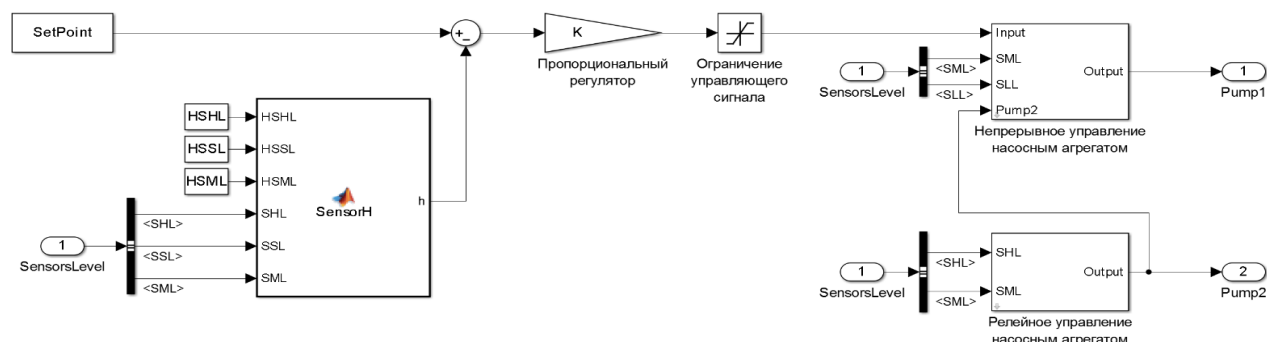


Рисунок 3.16 – Модель регулятора

На вхід моделі подається стан датчиків верхнього, середнього, нижнього рівня і датчика швидкості. Значення рівня, використовуване як сигнал зворотного зв'язку, розраховує функція "SensorH". Значення уставки задається через параметр "SetPoint", а коефіцієнт посилення через параметр "K". Значення рівнів розміщення датчиків задаються через параметри "HSHL", "HSSL", "HSML".

Модель системи управління із застосуванням розробленого регулятора приведена на рис. 3.17.

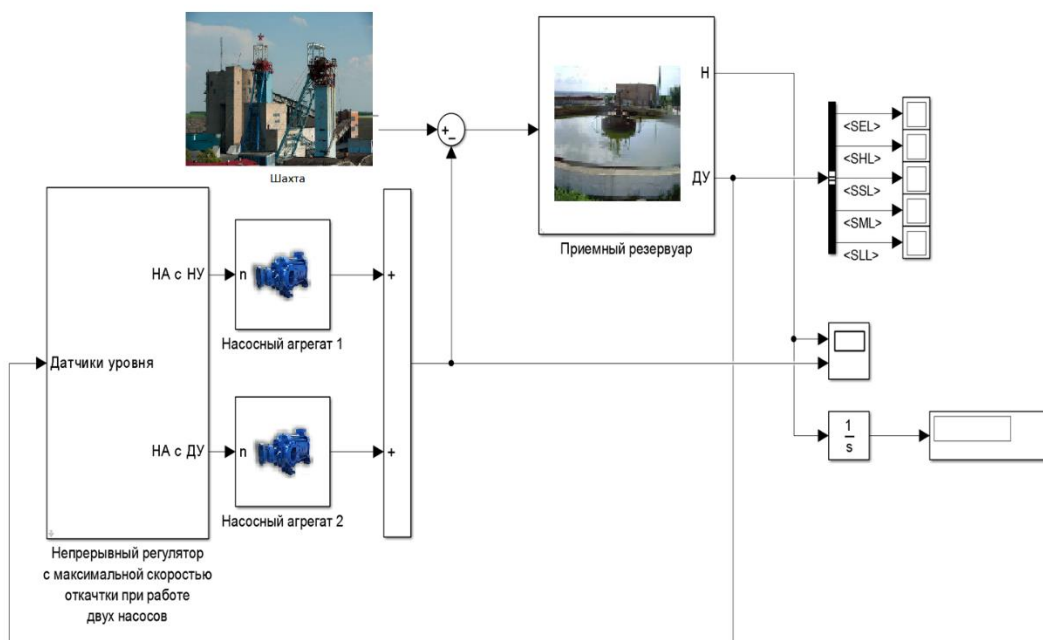


Рисунок 3.17 – Модель системи управління

Результати роботи моделі при витраті вод 350 л/с приведені на рис. 3.18.



Рисунок 3.18 – Результат моделювання системи управління

### 3.2.3 Оцінка якості функціонування системи управління

З метою оцінки якості функціонування системи управління, виконано ряд моделювань результати, яких приведені в табл. 3.9.

Таблиця 3.9 - Результати моделювання

Витрата вод, л/с	Інтеграл рівня вод / 104				
	Коефіцієнт посилення				
	-18	-35	-52	-70	-88
150	4,294	4,294	4,294	4,294	4,294
250	7,577	6,352	6,138	6,070	6,018
300	7,612	7,311	6,840	6,584	6,431
350	7,587	7,799	7,749	7,634	7,574
400	7,590	7,665	7,686	7,707	7,696

Як видно з результатів моделювання якість функціонування системи управління стала незначна гірше, що пов'язано з переходом від безперервного датчика до аналітичного датчика з використанням дискретних інформаційних сигналів. Проте навіть у такому вигляді ця система справляється з поставленим завданням і може бути використана для зменшення витрат електричної енергії на насосних станціях.

### 3.4 Висновки по розділу

В даному розділі була спроектована та налаштована комп'ютерна мережа управління шахтою. Модель мережі відповідає клієнт-серверному і розподіленому середовищу. В якості мережевих пристроїв було обране обладнання компанії Cisco Systems, так як воно відрізняється своєю надійністю та якістю. Основний акцент був зроблений на тому, щоб зробити систему відмовостійкою, безпечною та продуктивною. За для цього було використано ряд мережевих протоколів. Також був розрахований показник ефективності найбільшого сегменту мережі підприємства при максимальному навантаженні. Розрахунки показали, що система повністю відповідає пропускній здатності вихідного каналу.

Синтезована система управління насосною станцією на базі безперервного пропорційного регулятора. При її реалізації використані спеціально розроблені моделі безперервного і релейного управління насосними агрегатами. Для оцінки можливості, підвищення ефективності функціонування системи управління, використаний алгоритм що дозволяє, виконувати відкачування вод з верхнього рівня з максимальною продуктивністю насосних агрегатів. Дослідження

показали, що це рішення приводить до загального збільшення часу коли води знаходяться на високому рівні, що у свою чергу призводить до зменшення витрат електричній енергії. Реалізований перехід від безперервного датчика рівня до аналітичного датчика, який виконує розрахунок рівня вод при роботі насосного агрегату з частотним управлінням між датчиками середнього і верхнього рівня. Встановлено, що при витраті вод рівному продуктивності одного насосного агрегату або меншою оптимальний коефіцієнт посилення рівний - 18, а при вищій витраті - 52. При цьому коефіцієнти посилення можуть змінюватися відповідно до сезонного ритму подачі води з шахти.

## 4 ЕКСПЕРИМЕНТАЛЬНИЙ РОЗДІЛ

### 4.1 Комп'ютерна мережа підприємства

#### 4.1.1 Розробка математичної моделі мережі

Відповідно до структурної схеми комп'ютерної мережі та її імітаційної моделі розроблено структуру математичної моделі комп'ютерної мережі як замкнутої системи масового обслуговування (рис. 4.1) [3].

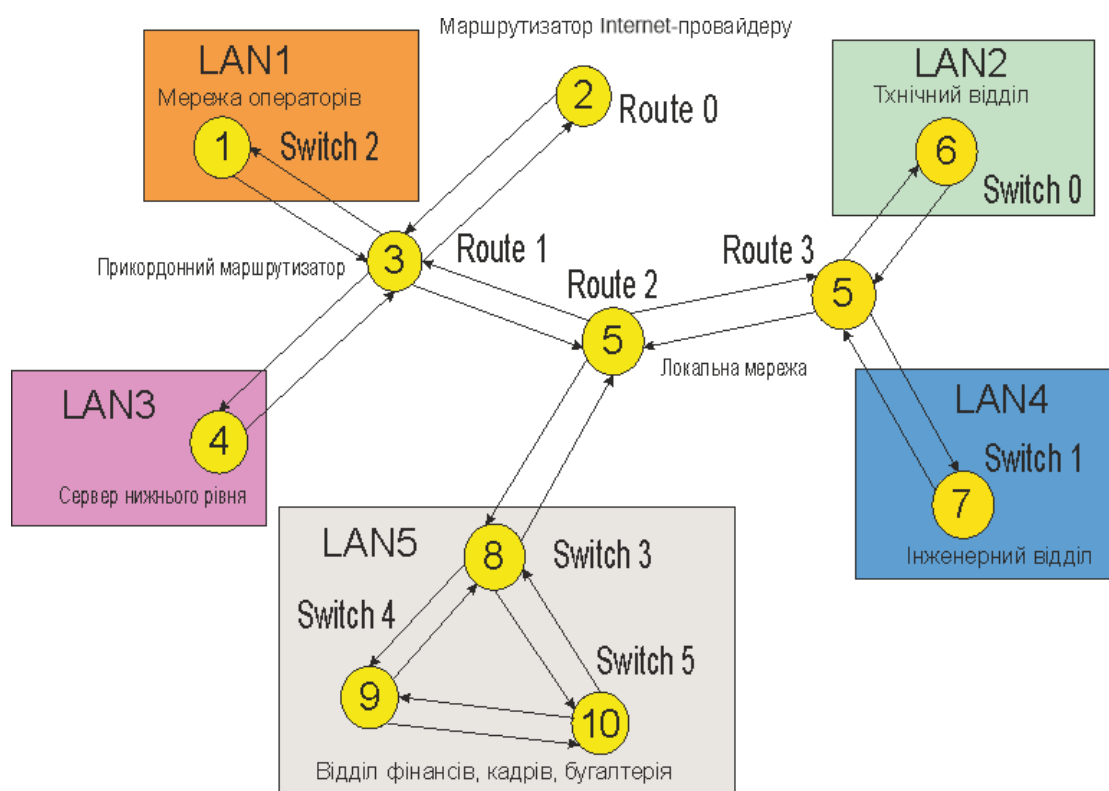


Рисунок 4.1 – Структура математичної моделі комп'ютерної мережі

В структурі моделі комп'ютерної мережі вузли 1, 4, 6, 7 – це комутатори, що обслуговують локальні мережі.

Вузли 2, 3, 5, 8, 9, 10 – маршрутизатори.

Зв'язки між елементами структури – це вірогідність передачі пакета від одного до другого вузла. Кожен вузол – це система масового обслуговування.

Вірогідність того, що вузол зв'язується сам з собою дорівнює нулю.

Вірогідності того, що вузол 1 зв'яжеться з вузлом 3 рівна 1 (100 %).

Вірогідності того, що вузол 2 зв'яжеться з вузлом 3 рівна 1.





### 4.1.2 Розрахунок параметрів мережі по її моделі

Далі методом Гауса розраховується матриця стовпчик з передаточними коефіцієнтами

$$e := \begin{pmatrix} 1 \\ 1.5 \\ 5 \\ 1 \\ 3.75 \\ 0.75 \\ 0.75 \\ 1.25 \\ 0.417 \\ 0.417 \end{pmatrix}$$

Задаємо матрицю  $m$ , коефіцієнти якої означають кількість конвеєрів обробки пакетів в кожному із вузлів системи масового обслуговування.

$$m_1 := \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Для розрахунків приймаємо, що в кожному пристрої знаходиться лише один конвеєр обробки пакетів.

Матриця  $B$  це матриця яка визначає з якою вірогідністю у відповідному вузлі (строчки) буде знаходитися у чеканні обробки пакетів (номер стовпчика).

	0	1	2	3	4
0	0.844	0.135	0.019	$2.268 \cdot 10^{-3}$	0
1	0.766	0.186	0.04	$6.928 \cdot 10^{-3}$	$7.25 \cdot 10^{-4}$
2	0.219	0.249	0.248	0.194	0.09
3	0.844	0.135	0.019	$2.124 \cdot 10^{-3}$	$1.432 \cdot 10^{-4}$
4	0.414	0.287	0.18	0.091	0.028
5	0.883	0.105	0.011	$9.113 \cdot 10^{-4}$	$4.531 \cdot 10^{-5}$
6	0.883	0.105	0.011	$9.113 \cdot 10^{-4}$	$4.531 \cdot 10^{-5}$
7	0.805	0.162	0.029	$4.079 \cdot 10^{-3}$	$3.496 \cdot 10^{-4}$
8	0.935	0.061	$3.534 \cdot 10^{-3}$	$1.601 \cdot 10^{-4}$	$4.33 \cdot 10^{-6}$
9	0.935	0.065	$3.699 \cdot 10^{-3}$	$1.644 \cdot 10^{-4}$	$4.33 \cdot 10^{-6}$

Відповідно до алгоритму Бузена розраховуються середні значення по кожному із вузлів мережі. Інтенсивність вхідного потоку пакетів у кожному вузлі:

	0
0	0.025
1	0.037
2	0.123
3	0.025
$\lambda =$ 4	0.092
5	0.018
6	0.018
7	0.031
8	0.01
9	0.01

Середнє число пакетів що чекають на обробку в кожному вузлі:

	0
0	0.18
1	0.291
2	1.686
3	0.18
$L =$ 4	1.033
5	0.13
6	0.13
7	0.233
8	0.069
9	0.073

Середній час обробки пакета в вузлі:

	0
0	7.32
1	7.888
2	13.731
3	7.326
$t =$ 4	11.217
5	7.065
6	7.065
7	7.6
8	6.739
9	7.134

#### 4.1.2.1 Параметри роботи мережі без впливу шкідливого програмного забезпечення

Робота комп'ютерної мережі в «нормальному» режимі характеризується наступними параметрами.

Кількість пакетів, які циркулюють у мережі дорівнює 6. Час обробки пакетів у всіх вузлах мережі однаків і складає 6 часових одиниць (для досліджуваної мережі 1 часова одиниця дорівнює 1 мілісекунді). Кількість конвеєрів обробки пакетів у кожному вузлі мережі дорівнює 1.

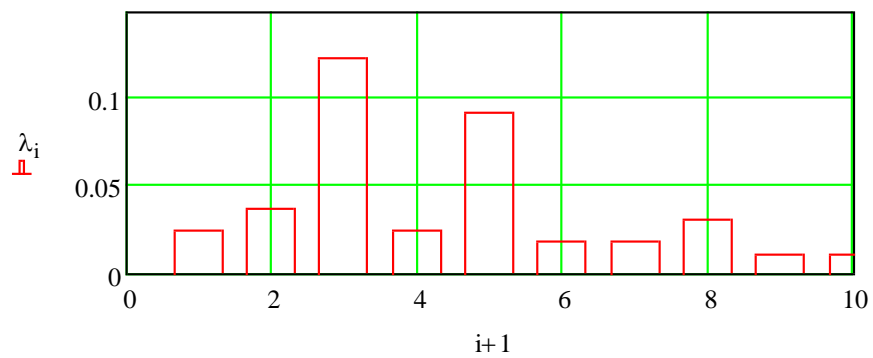


Рисунок 4.2 – Інтенсивність потоку, що входить у вузол

За таких вихідних даних отримано графіки, що показують усереднені характеристики кожного з вузлів.

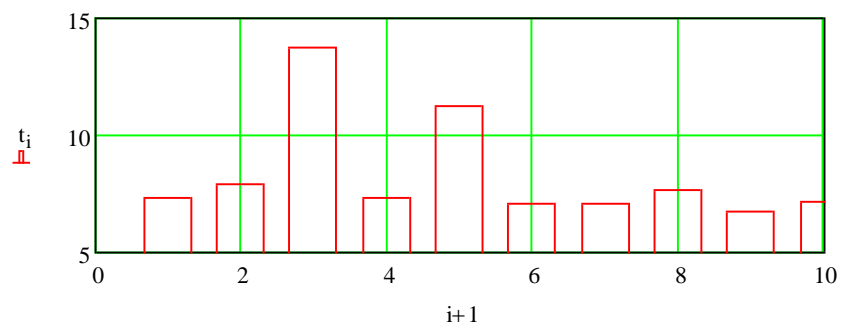


Рисунок 4.3 – Середній час перебування пакета у вузлі

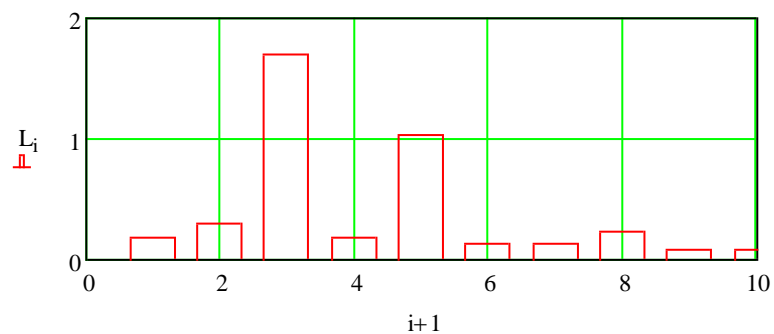


Рисунок 4.4 – Середня кількість пакетів які знаходяться у вузлі

Як бачимо, в цілому у всіх вузлах мережі, які є комутаторами усереднені параметри показують, що усі повідомлення обробляються швидко і без черги. Виключенням є вузли 3, 5, які відповідають за маршрутизацію всередині мережі.

Рис. 4.5 показує з якою вірогідністю у вузлах мережі буде черга.

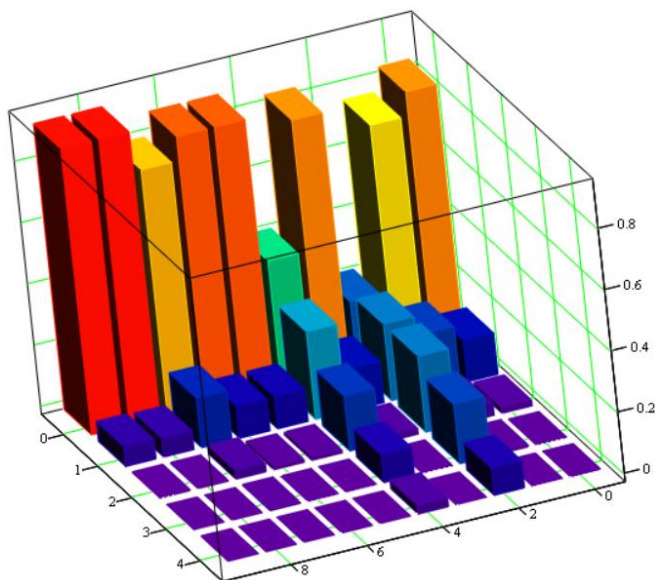


Рисунок 4.5 – Вірогідність черги у вузлах мережі при кількості пакетів 5

Можемо зробити попередній висновок про те, що при заданих параметрах маршрутизатори мережі є найбільш проблемним.

#### 4.1.2.2 Параметри роботи мережі під впливом вірусних програм

Вірусне ПЗ, як і будь-які інші програми, вимагають певного обсягу ресурсів ЕОМ, на якому вони виконуються, атакож можуть генерувати додатковий трафік в мережі. В залежності від виду вірусів, створювана ними навантаження на ЕОМ і мережу може сильно відрізнятись. Так, наприклад, класичні віруси в загальному випадку не створюють навантаження на мережу зовсім. Деякі ж мережеві віруси можуть здійснювати таку кількість мережевих запитів, що обчислювальна мережа може і зовсім перестати функціонувати [3].

Для моделювання подібної ситуації задаємо кількість запитів у мережі збільшуємо вдвічі,  $N=10$ .

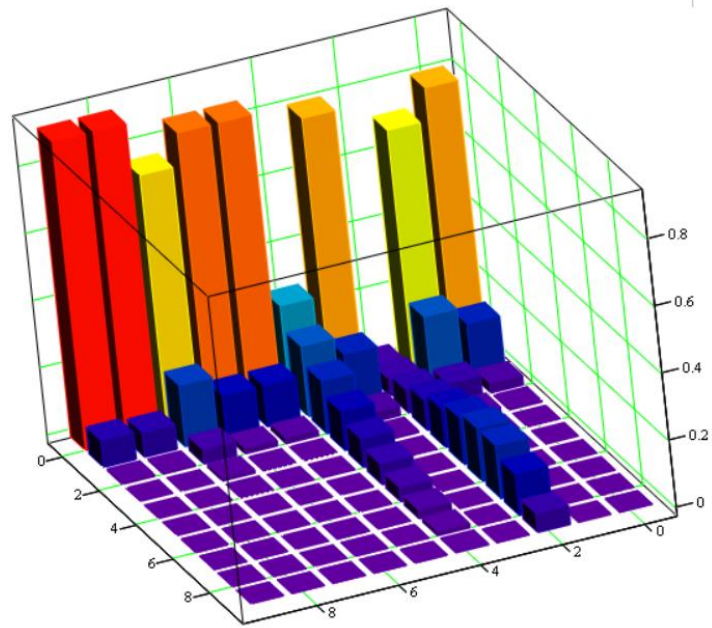


Рисунок 4.6 – Вірогідність черги у вузлах якщо в мережі циркулює 10 пакетів

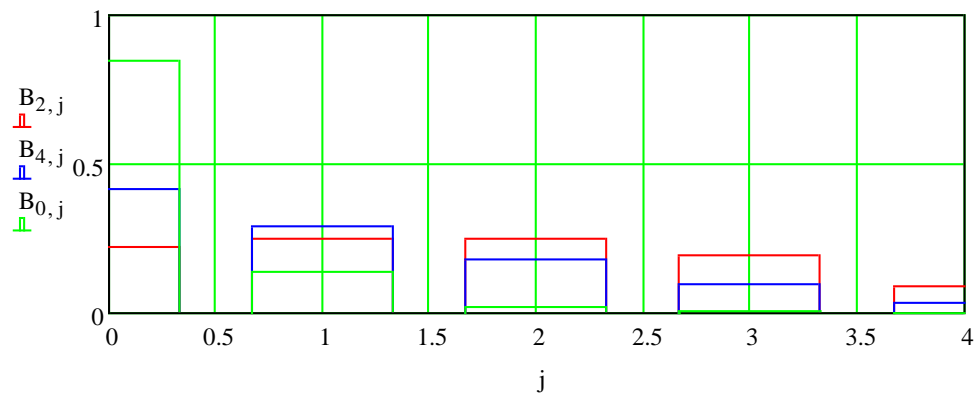


Рисунок 4.7 – Вірогідність черги у вузлах якщо в мережі циркулює 5 пакетів

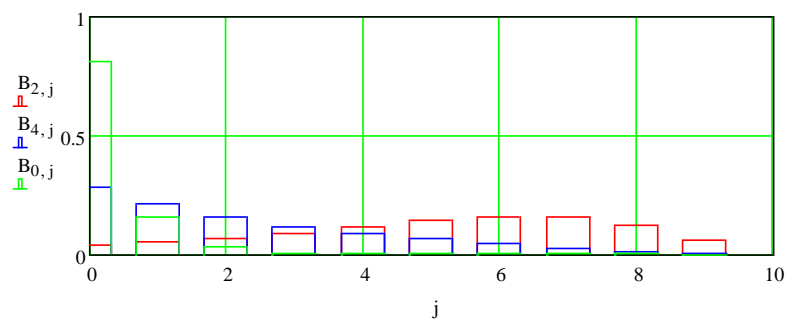


Рисунок 4.8 – Вірогідність черги у вузлах якщо в мережі циркулює 10 пакетів

Результати моделювання показують, що збільшення вдвічі кількості пакетів, що циркулюють в мережі приводить до того, що тільки для вузлів комутаторів № 3, 5 зростає вірогідність того, що в черзі уже будуть знаходитися 2 пакета і більше.

#### 4.1.3 Робота мережі із скоригованими характеристиками проблемних вузлів

Корекція характеристик вузлів № 3, 5 проводиться за рахунок підвищення швидкості (зменшення вдвічі часу) обробки пакетів.

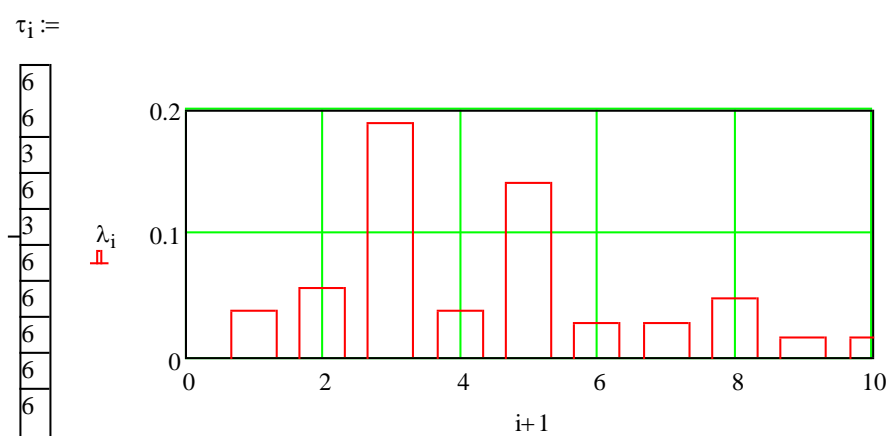


Рисунок 4.9 – Інтенсивність потоку, що входить у вузол

Відповідно до змін розраховані як усереднені так і ймовірнісні характеристики.

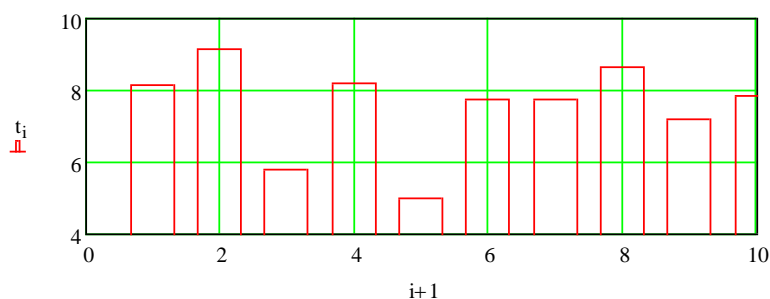


Рисунок 4.10 – Середній час перебування пакета у вузлі

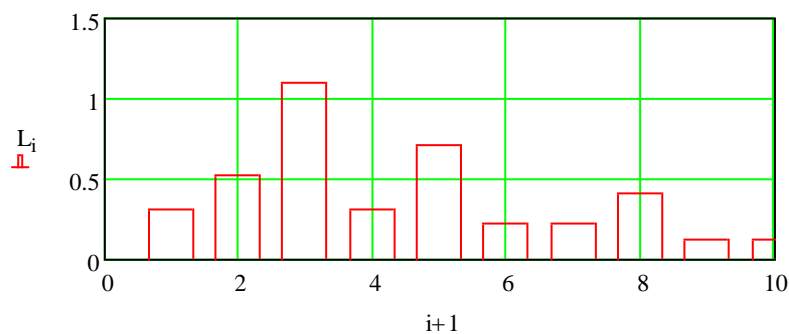


Рисунок 4.11 – Середня кількість пакетів які знаходяться у вузлі

Вірогідність того, що в вузлах мережі може виникнути черга значно знизилася (рис. 4.12)

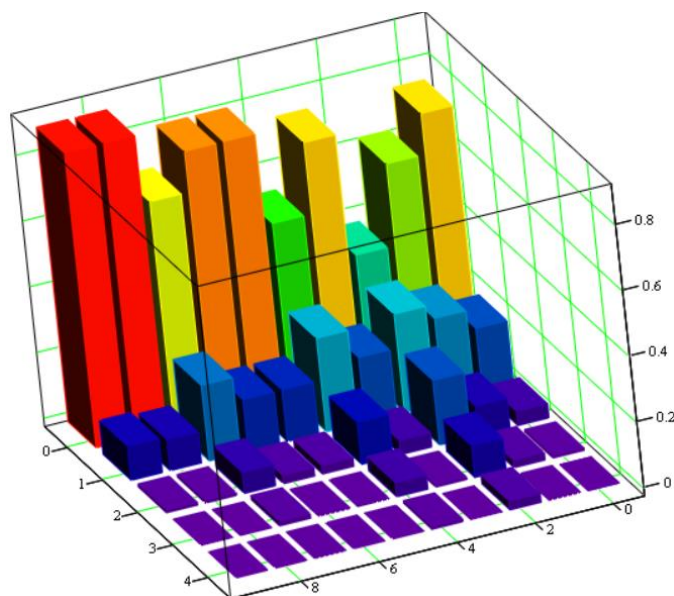


Рисунок 4.12 – Вірогідність черги у вузлах якщо в мережі циркулює 5 пакетів

Якщо в мережі із скоригованими характеристиками буде циркулювати 10 пакетів тоді вірогідність того, що в вузлах не буде черги зменшується для роутерів (рис. 13), але для роутера №3 всеодно є верогідність черги

Підвищення швидкості обробки пакетів у вузлах, які показали найменшу стійкість до перевантаження дозволили певним чином покращити характеристики мережі. Для поліпшення стану для роутеру №3 можна запропонувати застосовувати, ще більш продуктивне оладнання. Корекція

характеристик вузла № 3 проводиться за рахунок ще більшого підвищення швидкості (зменшення часу) обробки пакетів.

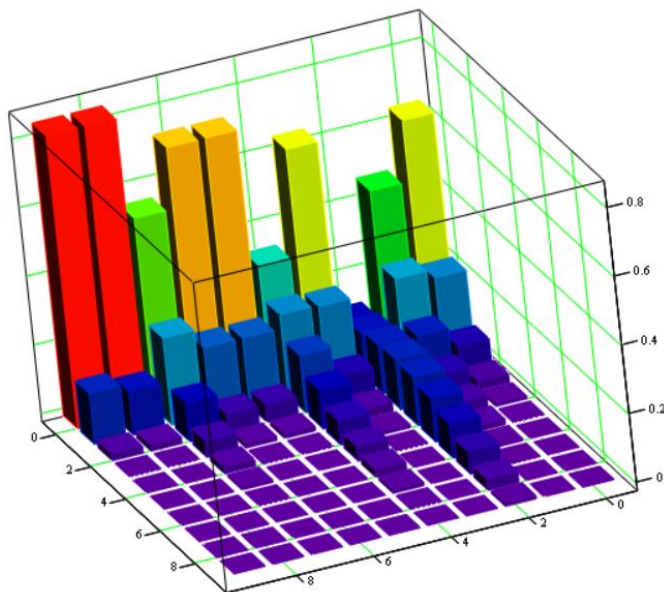


Рисунок 4.13 – Вірогідність черги у вузлах якщо в мережі циркулює 10 пакетів

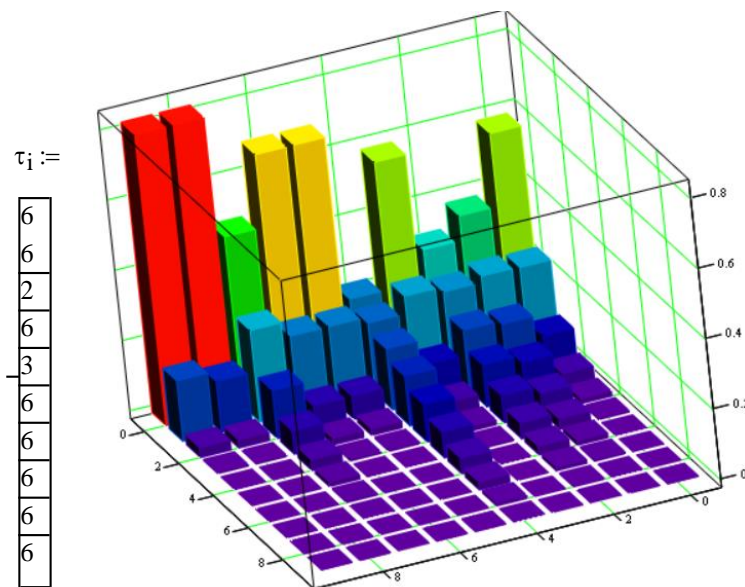


Рисунок 4.13 – Вірогідність черги у вузлах якщо в мережі циркулює 10 пакетів  
З поліпшеними характеристикам вузла № 3

Згідно з аналізом даних [3, 5], отриманих в результаті вивчення стану мережі під впливом тільки шкідливих програм, стохастичний характер маршрутної матриці, яка описує мережу, може стати причиною нелінійного зростання основних характеристик в деяких вузлах мережі, незважаючи на те, що



завантаження вузлів мережі зростатиме лінійно. Таке явище може призводити до збоїв в роботі комп'ютерної мережі.

Аналіз характеристик розглянутої мережі під впливом тільки АПО показує, що внаслідок запуску антивірусної програми в вузлах мережі з досить високою середньою завантаженістю конвеєрів, основні характеристики можуть погіршитися в рази.

За результатами порівняння характеристик обчислювальної мережі у всіх розглянутих станах зроблено висновок про те, що найбільш негативно на характеристиках мережі позначається атака шкідливими програмами.

## **4.2 Система управління водовідливом**

### **4.2.1 Цифрова модель системи управління**

Відповідно до завдання необхідно розробити програмне забезпечення системи управління і людино-машинний інтерфейс в SCADA системі zenon Supervisor 7.10. Програмне забезпечення має бути розроблене так, щоб як об'єкт управління міг використовуватися реальний об'єкт або його програмне модель. Виходячи з цього, модель системи управління має бути представлена у вигляді двох програмних модулів: регулятора і об'єкту управління.

Для перетворення моделей математичного пакету MATLAB в текст програми необхідно, щоб усі елементи цих моделей були цифровими. Таким чином, спочатку необхідно замінити усі безперервні елементи моделі системи управління на цифрові. Далі виконати розподіл моделі на модель регулятора і модель об'єкту управління. Після чого перетворити цифрові моделі в програмні модулі, які будуть використані при розробці програмного забезпечення системи управління в SCADA системі zenon.

У системі управління є присутній дві моделі, що включають безперервні елементи, насосних агрегатів і приймального резервуару. Оскільки перехідні процеси, що проходять в моделях насосних агрегатів, не чинять впливу на роботу системи в цілому, оскільки їх постійні часу значно менше постійних часу

приймального резервуару їх динамікою можна нехтувати. Спрощена модель насосного агрегату приведена на рис. 4.14.

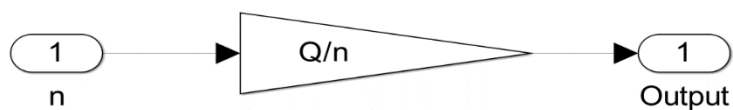


Рисунок 4.14 – Модель насосного агрегату

До складу моделі приймального резервуару входить один безперервний елемент інтегратор, який необхідно замінити його цифровим аналогом, після чого модель набере вигляду, як показано на рис. 4.15.

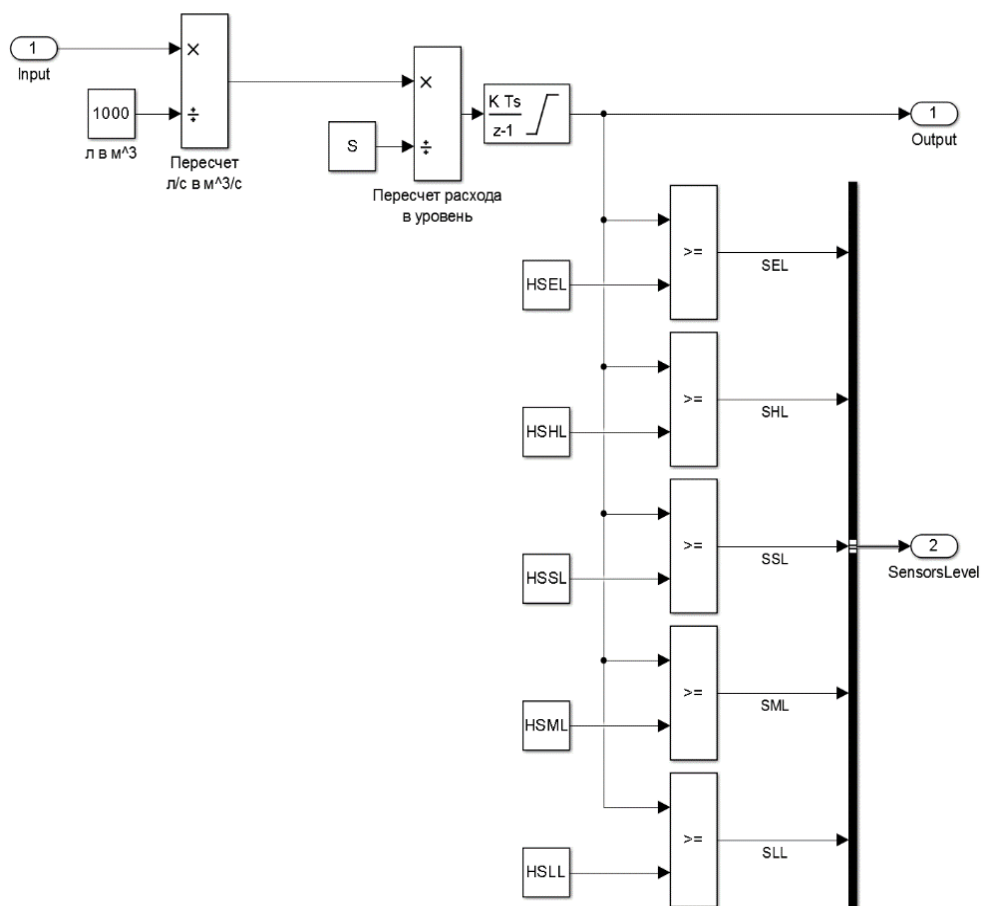


Рисунок 4.15 – Модель приймального резервуару

Далі виділимо модель об'єкту управління, в окрему підсистему, внаслідок чого вона набере вигляду як показано на рис. 4.16. У моделі є присутній два входи відповідні швидкості насосного агрегату 1 "bi\_n1" і швидкості насосного агрегату 2 "bi\_n2". Так само модель має один комплексний вихід відповідний датчикам рівня "bo\_Sensors".

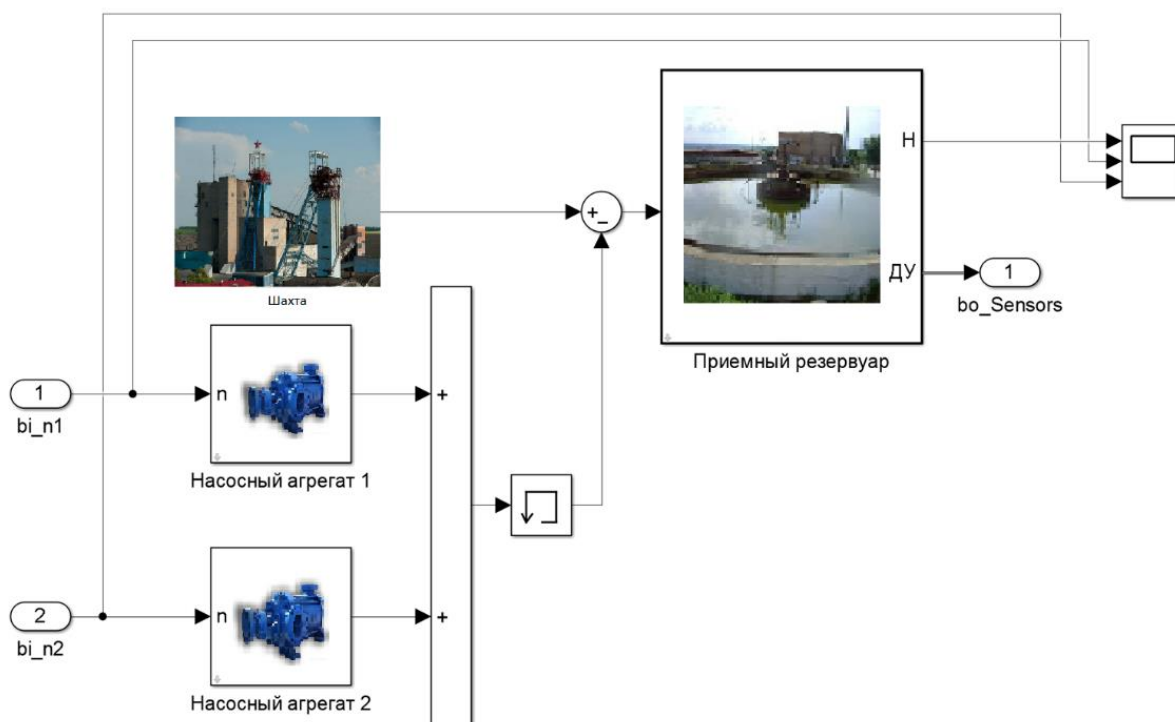


Рисунок 4.16 – Модель об'єкту управління

Модель регулятора початково синтезувалася в цифровій формі і оформлена у вигляді окремої підсистеми, тому не вимагає додаткових перетворень.

Таким чином, модель об'єкту управління і модель регулятора розділені в окремі підсистеми, а їх безперервні елементи замінені на цифрові. Отримана таким чином модель системи управління насосної станції приведена на рис. 4.17.

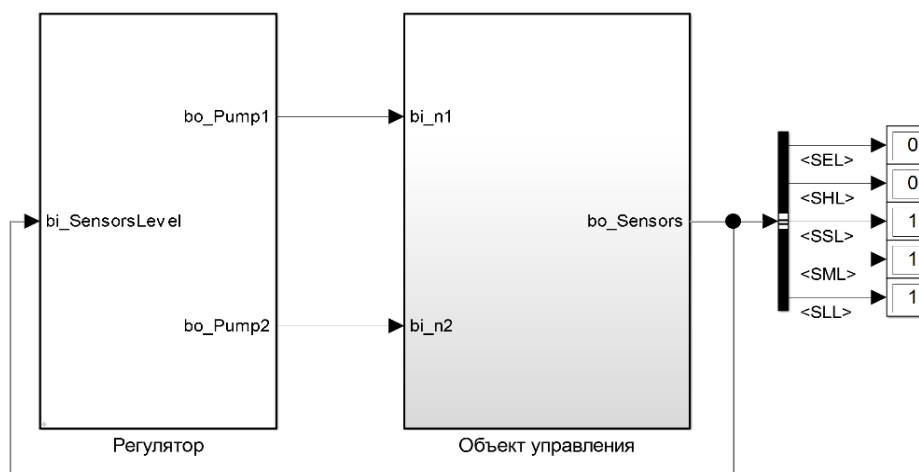


Рисунок 4.7 – Модель системи управління

Отримані моделі регулятора і об'єкту управління за допомогою стандартного засобу MATLAB для генерації програмного коду були перетворені в текст програми на мові програмування Structured Text приведений у відповідному додатку.

#### 4.2.2 Розробка програмного забезпечення системи управління

Відповідно до завдання необхідно розробити програмне забезпечення що реалізовує об'єкт управління і програмне забезпечення що реалізовує регулятора. Спочатку буде розроблено програмне забезпечення що реалізовує об'єкт управління. Для цього в SCADA системі zenon доданий драйвер "S 7 TCP - IP driver" для підключення до програмованого логічного контролера "VIPA 214-1BA03" за допомогою інтерфейсу Ethernet, а так само відповідні йому змінні (рис. 4.19).

Програмне забезпечення об'єкту управління функціонуватиме на програмному програмованому логічному контролері zenon Logic що реалізовує в режимі симуляції контролер VIPA.

State	Name	Identification	M...	Driver	Data type
	Filter text	Filter text		*VIPA_200	Filter text
Data type : REAL (2 Elements)					
	Pump1Speed	Скорость насосного агрегата 1		S7TCP32 - VIPA_200	REAL
	Pump2Speed	Скорость насосного агрегата 2		S7TCP32 - VIPA_200	REAL
Data type : BOOL (5 Elements)					
	SensorEL	Датчик аварийного уровня		S7TCP32 - VIPA_200	BOOL
	SensorHL	Датчик верхнего уровня		S7TCP32 - VIPA_200	BOOL
	SensorLL	Датчик нижнего уровня		S7TCP32 - VIPA_200	BOOL
	SensorML	Датчик среднего уровня		S7TCP32 - VIPA_200	BOOL
	SensorSL	Датчик скорости		S7TCP32 - VIPA_200	BOOL

Рисунок 4.19 – Змінні програмованого логічного контролера

На базі отриманого тексту програми з математичного пакету MATLAB розроблений функціональний блок "Object". Текст програми якого написаний на мові програмування Structured text (представлено у відповідному додатку):

```

IF l_State >= LREAL#10.0 THEN
    l_State := LREAL#10.0;
ELSIF l_State <= LREAL#0.0 THEN
    l_State := LREAL#0.0;
END_IF;

bo_SEL := l_State >= 9.0;
bo_SHL := l_State >= 8.0;
bo_SSL := l_State >= 5.1;
bo_SML := l_State >= 5.0;
bo_SLL := l_State >= 2.0;

l_State := (((bi_Q - l_Memory) / LREAL#1000.0) / LREAL#7.0685834705770345) * LREAL#0.1) + l_State;
IF l_State >= 10.0 THEN
    l_State := 10.0;
ELSIF l_State <= 0.0 THEN
    l_State := 0.0;
END_IF;

l_Memory := (LREAL#0.020689655172413793 * bi_n1) + (LREAL#0.020689655172413793 * bi_n2);

```

Спочатку виконується перевірка стану інтегратора збереженого в попередньому циклі. Якщо значення більше 10,0, йому привласнюється значення рівне 10,0, якщо воно менше 0,0 те йому привласнюється значення рівне 0,0. Далі йде оцінка рівня вод якщо значення більше або рівне 9,0 спрацьовує датчик аварійного рівня, якщо воно більше або рівне 8,0 спрацьовує датчик верхнього рівня, якщо воно більше або рівне 5,1 спрацьовує датчик швидкості, якщо воно більше або рівне 5,0 спрацьовує датчик середнього рівня і якщо воно більше або рівне 2,0 спрацьовує датчик нижнього рівня. Далі йде розрахунок рівня води в приймальному резервуарі, після чого він знову обмежується. У кінці виконується розрахунок витрати формованого насосними агрегатами.

Виклик функціонального блоку виконується в основному тілі програми, приведеному на рис. 4.20.

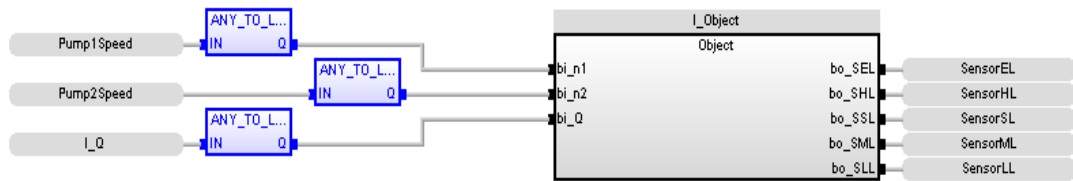


Рисунок 4.20 – Виклик функціонального блоку "Object"

На вхід блоку подаються значення швидкостей насосних агрегатів і витрата води. З вихід блоку виходить стани датчиків аварійного, верхнього, середнього, нижнього рівня і датчика швидкості.

Відповідно до завдання програмне забезпечення контролера повинне функціонувати в SCADA системі zenon на програмному програмованому логічному контролері zenon Logic. Для цього в SCADA системі zenon створений проект zenon Logic "Controller". Для зв'язку контролера з SCADA системою додані змінні, приведені на рис. 4.21.

На базі отриманого тексту програми з математичного пакету MATLAB розроблений функціональний блок "Controller". Текст програми якого написаний на мові програмування Structured text і приведений у відповідному додатку.

Виклик функціонального блоку виконується в основному тілі програми, приведені на рис. 4.22.

State	Name	Identification	M...	Driver	Data type
Filter text	Filter text	Filter text	Filter text	*Controller	Filter text
Data type : REAL (2 Elements)					
	Controller/Global/Pump1Speed	Скорость насосного агрегата 2		STRATONNG - zenon Logic: Controller	REAL
	Controller/Global/Pump2Speed	Скорость насосного агрегата 1		STRATONNG - zenon Logic: Controller	REAL
Data type : BOOL (5 Elements)					
	Controller/Global/SensorEL	Датчик аварийного уровня		STRATONNG - zenon Logic: Controller	BOOL
	Controller/Global/SensorHL	Датчик верхнего уровня		STRATONNG - zenon Logic: Controller	BOOL
	Controller/Global/SensorLL	Датчик нижнего уровня		STRATONNG - zenon Logic: Controller	BOOL
	Controller/Global/SensorML	Датчик среднего уровня		STRATONNG - zenon Logic: Controller	BOOL
	Controller/Global/SensorSL	Датчик скорости		STRATONNG - zenon Logic: Controller	BOOL

Рисунок 4.21 – Змінні програмного програмованого логічного контролера

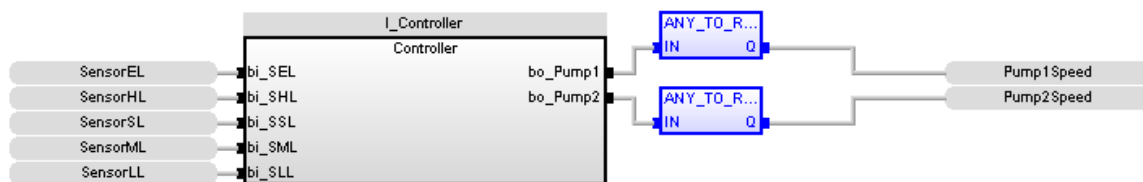


Рисунок 4.22 – Виклик функціонального блоку “Controller”

На вхід блоку подаються стани датчиків аварійного, верхнього, середнього, нижнього рівня і датчика швидкості. З вихід блоку виходить швидкість насосних агрегатів.

### 4.2.3 Розробка програмного забезпечення людино-машинного інтерфейсу

Програмне забезпечення людино-машинного інтерфейсу (наведене у відповідному додатку), що розробляється, повинне містити чотири вікна: технологічного процесу, тренду зміни швидкостей роботи асинхронних електродвигунів насосних агрегатів, хронологічного списку подій і інформаційного списку тривог. Виходячи з того, що людино-машинний інтерфейс повинен відображувати декілька зображень, він повинен складатися з двох шаблонів на одному повинні відображуватися кнопки перемикання зображень, а на другому самі зображення.

Оскільки людино-машинний інтерфейс повинен відображатися на моніторі з дозволом 1 366 x 768 пкс, розмір шаблону основного зображення "MainScreenFrame" вибраний рівним 1 366 x 700 пкс (рис. 4.23). На основному зображенні відображуватимуться зображення технологічного процесу, тренду і список. Розмір додаткового шаблону "ButtonScreeFrame" вибраний рівним 1 366 x 68 пкс, на нім відображуватимуться кнопки перемикання зображень.

Далі розробимо зображення кнопок. На нім мають бути розміщені кнопки перемикання зображень, кнопка перезавантаження людино-машинного інтерфейсу і кнопка припинення роботи людино-машинного інтерфейсу. Розроблене зображення приведене на рис. 4.24.



Рисунок 4.23 – Шаблон зображення

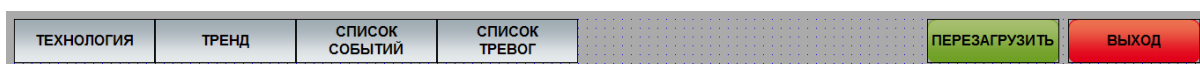


Рисунок 4.24 – Зображення кнопок

На зображенні технологічного процесу повинні відображатися насосна станція, датчики рівня вод, стан насосів і швидкість роботи насосів. Розроблене зображення технологічного процесу приведенне на рис. 4.25.

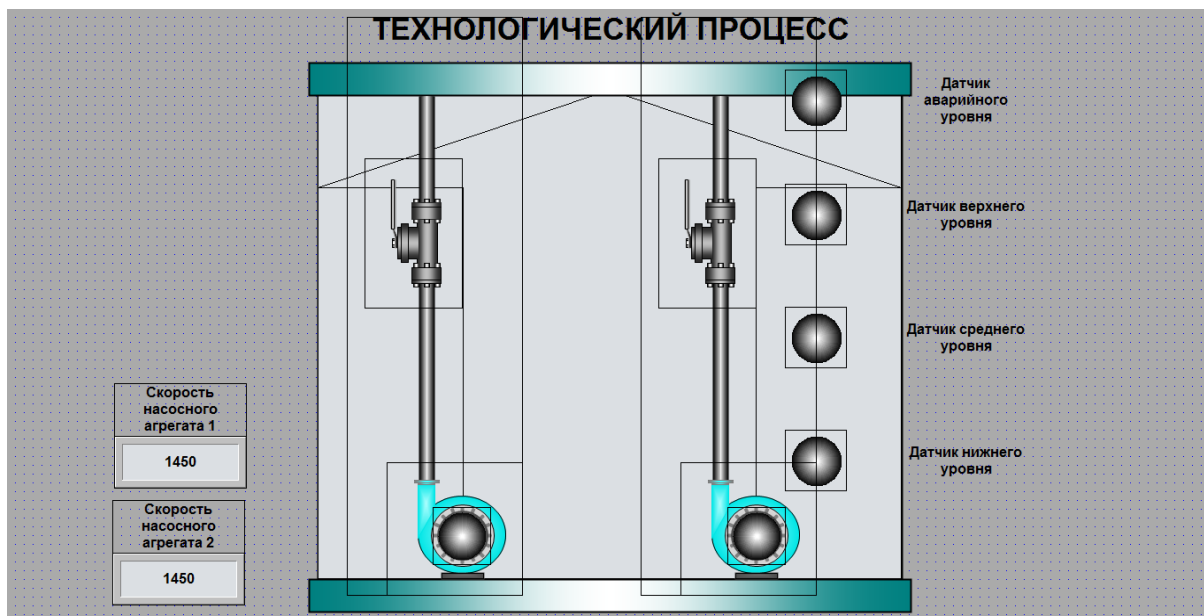


Рисунок 4.25 – Зображення технологічного процесу

Для реалізації зміни рівня вод використаний елементу стовпчикової діаграми "Bar", який має п'ять станів: води нижчі за датчик нижнього рівня, води



вищі за датчик нижнього рівня, води вищі за датчик середнього рівня, води вищі за датчик аварійного рівня.

Для відображення тренду використано зображення типу "Extended trend". Тренд налагоджений таким чином, що б відображувати швидкість асинхронних електродвигунів насосних агрегатів. Діапазон відображення рівний по осі Y від 0 до 1 500 об/мін, а по осі X 1 година астрономічного часу. Тренд насосного агрегату 1 відображується червоним кольором, а насосного агрегату 2 зеленим кольором. Розроблене зображення приведенне на рис. 4.26.

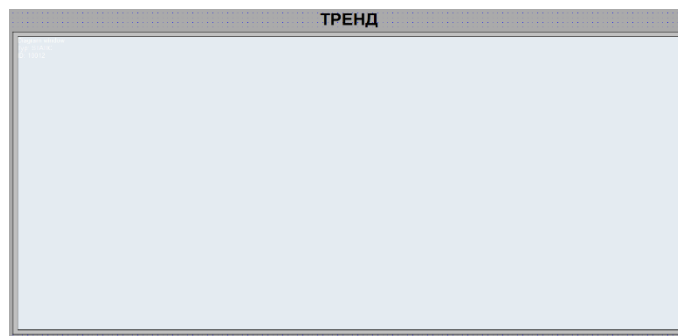


Рисунок 4.126 – Зображення тренду

Для відображення списку подій використано зображення типу "Chronologic event list". Відповідно до вимог в хронологічний список подій записується інформація про включення/виключенні насосних агрегатів і про спрацьовування датчика аварійного рівня. Розроблене зображення приведенне на рис. 4.27.

Для відображення списку тривог використано зображення типу " Alarm message list". Відповідно до вимог в інформаційний список тривог записується інформація про спрацьовування датчика аварійного рівня. Розроблене зображення приведенне на рис. 4.28.

Для зв'язку зображення з даними програмного програмованого логічного контролера управління, що реалізовує об'єкт, і програмного програмованого логічного контролера що реалізовує регулятора використані змінні, список яких приведені на рис. 4.29.

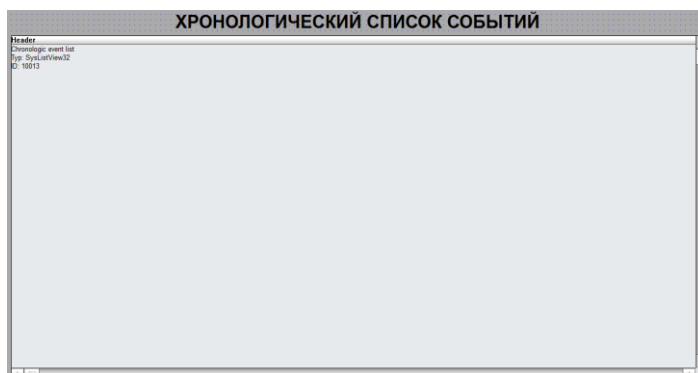


Рисунок 4.28 – Зображення інформаційних тривог

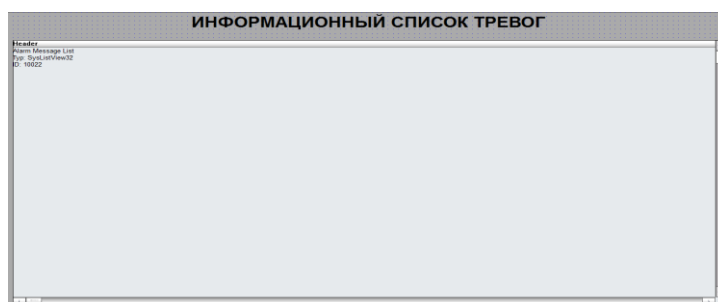


Рисунок 4.29 – Зображення хронологічного списку подій

State	Name	Идентификатор	Единиц...	Net address	Data block	Offset	Bit num...	Alignm...	Symbol...	#	Driver	Data type	Decimals	Start offset
F...	Filter text	Filter text	Filter...	Filter text	Filter text	Filter text	Filter...	Filter...	Filter...	Filter...	Filter text	Filter text	Filter...	Filter text
	Border			0	0	0	0	0			Intern - Driver for int...	BOOL	0	0
	Controller/Global/Pump1Sp...	Скорость насосног...		0	0	0	0	0			STRATONMG - zenon ...	REAL	0	0
	Controller/Global/Pump2Sp...	Скорость насосног...		0	0	0	0	0			STRATONMG - zenon ...	REAL	0	0
	Controller/Global/SensorEL	Датчик аварийног...		0	0	0	0	0			STRATONMG - zenon ...	BOOL	0	0
	Controller/Global/SensorHL	Датчик верхнего ...		0	0	0	0	0			STRATONMG - zenon ...	BOOL	0	0
	Controller/Global/SensorLL	Датчик нижнего у...		0	0	0	0	0			STRATONMG - zenon ...	BOOL	0	0
	Controller/Global/SensorML	Датчик среднего ...		0	0	0	0	0			STRATONMG - zenon ...	BOOL	0	0
	Controller/Global/SensorSL	Датчик скорости		0	0	0	0	0			STRATONMG - zenon ...	BOOL	0	0
	Level	Уровень стоков		0	0	0	0	0			MATHDR32 - Driver Fo...	LREAL	0	0
	Pump1Speed	Скорость насосног...		0	0	0	0	0			57TCP32 - VIPA_200	REAL	0	0
	Pump2Speed	Скорость насосног...		0	0	0	0	0			57TCP32 - VIPA_200	REAL	0	0
	SensorEL	Датчик аварийног...		0	0	0	0	0			57TCP32 - VIPA_200	BOOL	0	0
	SensorHL	Датчик верхнего ...		0	0	0	0	0			57TCP32 - VIPA_200	BOOL	0	0
	SensorLL	Датчик нижнего у...		0	0	0	0	0			57TCP32 - VIPA_200	BOOL	0	0
	SensorML	Датчик среднего ...		0	0	0	0	0			57TCP32 - VIPA_200	BOOL	0	0
	SensorSL	Датчик скорости		0	0	0	0	0			57TCP32 - VIPA_200	BOOL	0	0

Рисунок 4.30 - Список змінних

#### 4.2.4 Перевірка функціонування програмного забезпечення

Перевірка функціонування розробленого програмного забезпечення здійснюється в режимі програмної симуляції, коли замість програмованого логічного контролера використовується програмний програмований логічний контролер з моделлю об'єкту управління.

В процесі перевірки було встановлено, що усі індикатори технологічного процесу відображують правильну інформацію, зміна рівня вод призводить до

спрацьовування відповідних датчиків, рівень вод відображується коректно (рис. 4.31).

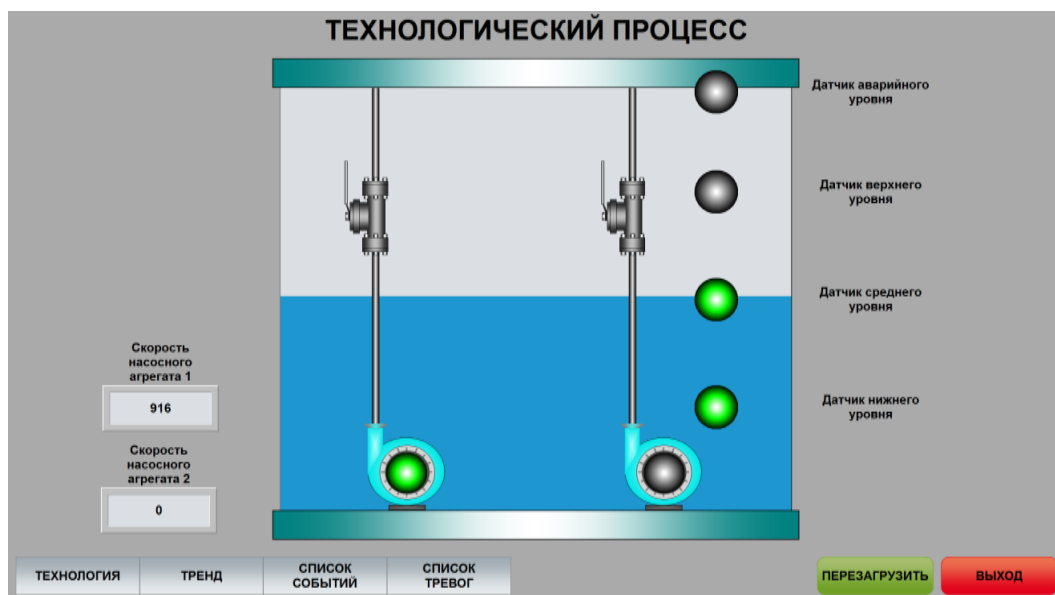


Рисунок 4.31 – Зображення технологічного процесу

Перевірка правильності функціонування програмного забезпечення системи управління, і зображення тренду здійснювалося спільно. Регулятор працює відповідно до алгоритму управління, а зображення відображує швидкість обертання насосних агрегатів (рис. 4.32).

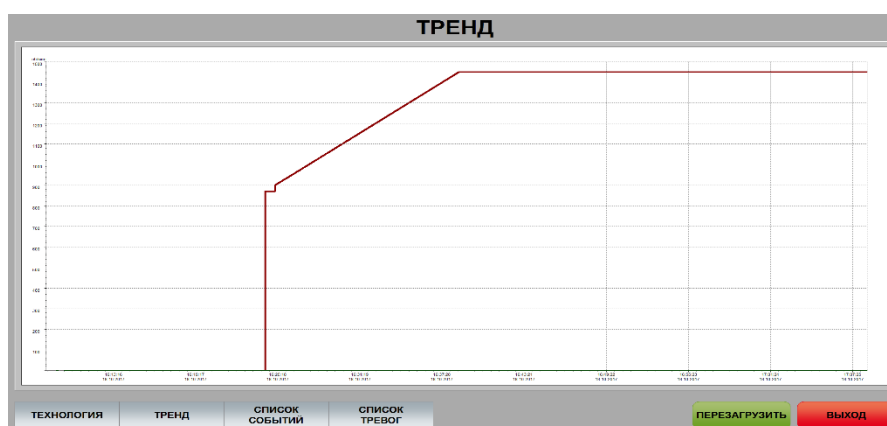


Рисунок 4.32 – Зображення тренду

На зображенні хронологічного списку подій відображується включення/виключення насосних агрегатів і спрацьовує датчика аварійного

рівня. Зображення інформаційного списку тривог відображує попередження про спрацьовування датчика аварійного рівня. Розроблене програмне забезпечення забезпечує виконання усіх поставлених завдань.

### **4.3 Висновки по розділу**

Розроблена модель дозволяє визначити усереднені значення інтенсивності вхідного потоку, часу перебування пакета у вузлах, середню кількість пакетів у вузлах. Що відноситься до класу безрозмірних характеристик. Аналіз ймовірностей черги у вузлах мережі показав що найбільша ймовірність черги виникає у вузлах №3, 5. При збільшенні кількості пакетів, що циркулюють у мережі вдвічі у вузлах №3 і 5 висока ймовірність виникнення черги з 2 – 8 пакетів. Ця ситуація можлива коли в мережі функціонує шкідливе програмне забезпечення. Для підвищення стійкості мережі до перевантаження необхідно замінити роутер, який є вузлами №3, 5 моделі на більш потужний. Час обробки пакетів у цього пристрою повинен бути в 2 рази менший.

На базі безперервної моделі системи управління розроблена цифрова модель системи управління. З моделі виділені модель об'єкту управління і модель регулятора. За допомогою засобів математичного пакету MATLAB отримані програмні моделі, що реалізують об'єкту управління і регулятор на мові програмування Structured Text. На базі програмних модулів розроблено програмне забезпечення системи управління для програмного програмованого логічного контролера zenon Logic що входить до складу SCADA системи zenon. У SCADA системі zenon розроблений людино-машинний інтерфейс включаючий технологічний процес, що відображує, тренд зміни швидкостей насосних агрегатів і хронологічний список подій реєструючий включення і виключення насосних агрегатів, і інформаційний список застережливий тривог про досягнення води аварійного рівня. Виконана перевірка функціонування програмного забезпечення. Програмне забезпечення функціонує відповідно до заданих алгоритмів і забезпечує виконання поставлених завдань.

## ВИСНОВКИ

На основі детального аналізу комп'ютерної системи підприємства виконане дослідження властивостей комп'ютерної мережі підприємства.

Для виконання поставлених завдань дослідження використані сучасні методи, які використовуються для дослідження комп'ютерних систем як мереж масового обслуговування з урахуванням особливостей їх структури і інформаційного середовища.

1. Аналіз комп'ютерної мережі підприємства показав, що мережа є ключовим елементом комп'ютерної системи з точки зору забезпечення ефективного виробничого процесу. Але нема даних про можливість виконувати мережею функції при інформаційному перенавантаженні.
2. Вирішення завдань досліджень можливе при застосуванні методів теорії масового обслуговування. Класичні підходи до рішення завдань доволі трудомісткі, тому використаний алгоритм Бузена.
3. Розроблена математична модель комп'ютерної мережі як замкнутої системи масового обслуговування. Обґрунтовано обрані параметри перехідної матриці мережі.
4. Розроблена модель дозволяє визначити усереднені значення інтенсивності вхідного потоку, часу перебування пакета у вузлах, середню кількість пакетів у вузлах. Що відноситься до класу безрозмірних характеристик.
5. Аналіз ймовірностей черги у вузлах мережі показав що найбільша ймовірність черги виникає у вузлах №3, 5.
6. При збільшенні кількості пакетів, що циркулюють у мережі вдвічі у вузлах №3, 5 висока ймовірність виникнення черги з 2 – 8 пакетів. Ця ситуація можлива коли в мережі функціонує шкідливе програмне забезпечення.

7. Для підвищення стійкості мережі до перевантаження необхідно замінити роутер, який є вузлами №3, 5 моделі на більш потужний. Час обробки пакетів у цього пристрою повинен бути в 2 рази менший.

У структуру системи водовідведення входять насосні станції, які виконують перекачування шахтних вод. У роботі цих станцій основним споживачем електроенергії є насоси.

Забезпечення підвищення ефективності використання електроенергії можливо при максимізації часу роботи насосних агрегатів, коли вода знаходиться на рівні відповідному датчику верхнього рівня. Проте при цьому частота живлення насосів не повинна опускатися нижче 30 Гц, для забезпечення достатнього коефіцієнта корисної дії.

Виходячи з цього, основним завданням що стоїть перед системою управління насосною станцією являється відкачування шахтних вод таким чином, що б забезпечити мінімальні енерговитрати з дотриманням вимог технологічного процесу.

Для реалізації моделі об'єкту управління розроблені модель приймального резервуару і модель насосного агрегату. На підставі отриманих моделей розроблена модель насосної станції. Отримана модель дозволяє задавати витрату води, швидкість обертання насосів, контролювати рівень води в резервуарі за допомогою датчиків нижнього, середнього, верхнього і аварійного рівнів.

Сформульовані критерії якості функціонування об'єкту управління. Для насосної станції основним критерієм якості є мінімум витрат електроенергії. Для забезпечення мінімального споживання електроенергії на НС води повинні знаходитися на верхньому рівні, а швидкість роботи насосних агрегатів повинна знаходитися в діапазоні від 870 до 1 450 об/хв. При цьому повинні дотримуватися усі вимоги технологічного процесу.

Синтезована система управління станцією на базі безперервного пропорційного регулятора. При її реалізації використані спеціально розроблені моделі безперервного і релейного управління насосними агрегатами.

Для оцінки можливості, підвищення ефективності функціонування системи управління, використаний алгоритм що дозволяє, виконувати відкачування шахтних вод з верхнього рівня з максимальною продуктивністю насосних агрегатів. Дослідження показали, що це рішення приводить до загального збільшення часу коли води знаходяться на високому рівні, що у свою чергу призводить до зменшення витрат електричної енергії.

Реалізований перехід від безперервного датчика рівня до аналітичного датчика, який виконує розрахунок рівня води при роботі насосного агрегату з частотним управлінням між датчиками середнього і верхнього рівня.

Встановлено, що при витраті води рівному продуктивності одного насосного агрегату або меншою оптимальний коефіцієнт посилення рівний - 18, а при вищій витраті - 52. При цьому коефіцієнти посилення можуть змінюватися відповідно до сезонного графіку вступу шахтних вод.

На базі безперервної моделі системи управління розроблена цифрова модель системи управління. З моделі виділені модель об'єкту управління і модель регулятора. За допомогою засобів математичного пакету MATLAB отримані програмні моделі, що реалізують об'єкту управління і регулятор на мові програмування Structured Text.

На базі програмних модулів розроблено програмне забезпечення системи управління для програмного програмованого логічного контролера zenon Logic що входить до складу SCADA системи zenon.

У SCADA системі zenon розроблений людино-машинний інтерфейс включаючий технологічний процес, що відображує, тренд зміни швидкостей насосних агрегатів і хронологічний список подій реєструючий включення і виключення насосних агрегатів, і інформаційний список тривоги застережливий про досягнення шахтних вод аварійного рівня.

Виконана перевірка функціонування програмного забезпечення. Програмне забезпечення функціонує відповідно до заданих алгоритмів і забезпечує виконання поставлених завдань.

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Журавська І. М. Проектування та монтаж локальних комп'ютерних мереж :[навчальний посібник] / І. М. Журавська. – Миколаїв : Видавництво ЧДУ ім. Петра Могили, 2016. – 396 с.
2. Жуков, І. А. Комп'ютерні мережі та технології : навч. посіб./І. А. Жуков, В. О. Гуменюк, І. Є. Альтман. – К. : НАУ, 2004. – 276 с.
3. Груздева Л.М., Монахов Ю.М., Монахов М.Ю. Оценка сетевых характеристик компьютерных сетей в условиях информационного вредоносного воздействия: учебное пособие. / Владим. гос. ун-т. – Владимир: Изд-во Владим. гос. ун-та, 2010. – с.
4. Литвинов А. Л. Теорія систем масового обслуговування : навч. посібник / А. Л. Литвинов ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2018. – 141 с..
5. Globa L.S. Approaches and technologies of creating data-processing resources in the telecommunication environment – Электроника и связь. Проблемы электроники ч.2, Киев, 2005
6. Глоба Л.С. Математичні основи побудови інформаційно телекомунікаційних систем. - К.: Політехніка, 2003. -276с
7. ГОСТ 2.702-75. ЕСКД. Правила выполнения электрических схем. – М.: Госстандарт, 1995. – 115 с.
8. IP Калькулятор [Электронный ресурс] – Режим доступа : URL : <http://ip-calculator.ru/>. – Загол. з екрана.
9. VLSM Calculator – калькулятор подсетей с маской переменной длины [Электронный ресурс]. – Режим доступа:URL:<http://www.vlsm-calc.net/>. – Загол. з екрана.
10. Воробьёва Н.И., Корнейчук В.И., Савчук Е.В. Надёжность компьютерных систем. – К.: «Корнійчук», 2002. – 144 с.
11. Мережеве обладнання [Электронный ресурс] – Режим доступа : URL: [https://elmir.ua/routers/router\\_zyxel\\_sbg5500-a.html](https://elmir.ua/routers/router_zyxel_sbg5500-a.html). – Загол. з екрану.



12. Классификация угроз информационной безопасности (Электрон. ресурс)/Способ  
доступу: URL: [http://www.cnews.ru/reviews/free/oldcom/security/elvis\\_class.shtml](http://www.cnews.ru/reviews/free/oldcom/security/elvis_class.shtml) – Загол. з екрана..
13. Правила з технічного захисту інформації для приміщень банків, у яких обробляються електронні банківські документи (Електрон. ресурс) /  
Способ доступу: URL: <http://www.txnet.com/ekranuvanna-servernih-primisen> – Загол. з екрана.
14. Новиков Ю.В., Карпенко Д.Г. Аппаратура локальных сетей: функции, выбор, разработка / Под общей редакцией Ю.В. Новикова. – М.: Эком, 1998. – 288 с.
15. Кулаков Ю.А., Луцкий Г.М. Локальные сети. – К.: Юниор, 1998. – 336 с.
16. Кулаков Ю.А., Омелянский С.В. Компьютерные сети. Выбор, установка, использование и администрирование. – К: Юниор, 1999. – 544 с.
17. Спортак М, Паппас Ф., Рензинг Э. Компьютерные сети. Книга 1. Энциклопедия пользователя: Пер. с англ. – М.: Диасофт, 1998. – 432 с.
18. Баня Е.Н. Компьютерные сети. – К.: Світ, 1999. – 112 с.
19. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб.: Питер, 2001. – 172 с.
20. Джеймс Челлис Основы построения сетей: Учебное пособие для специалистов MCSE 1.0. – СПб.: Питер, 1997. – 326 с.
21. Технологии разработки программного обеспечения: / Учебник С. Орлов. – СПб.: Питер, 2002. – 464 с.
22. Microsoft Corporation. Принципы проектирования и разработки программного обеспечения. Учебный курс MSCD/ Пер. с англ. – М.: Издательско-торговый дом «Русская редакция», 2002. – 736 с.
23. Розробка програмного забезпечення комп'ютерних систем. Програмування [Текст]: навч. посібник / Л.І. Цвіркун, А.А. Євстігнєєва, Я.В. Панферова. – 2-ге вид., випр. – Д.: Національний гірничий університет, 2011. – 222 с.

24. Цвіркун Л.І. Глобальні комп'ютерні мережі. Програмування мовою РНР: навч. посібник / Л.І. Цвіркун, Р.В. Липовий, під заг. ред. Л.І. Цвіркуна. – Д.: Національний гірничий університет, 2013. – 239 с.
25. Горные машины для подземной добычи угля / П.А. Горбатов, Г.В. Петрушкин, Н.М. Лысенко и др. – Донецк: Донецький національний технічний університет, 2006. – 669 с.
26. Морозов В.И. Очистные комбайны. Справочник / В.И. Морозов, В.И. Чуденков, Н.В. Сурина. – М.: Высшее горное образование, 2006. – 650 с.
27. Зайков В. И. Эксплуатация горных машин и оборудования / В.И. Зайков, Г.П. Берлявский. – Москва: Высшее горное образование, 2006. – 257 с.
28. Машины и оборудование для шахт и рудников. Справочник / С.Х. Клорикьян, В.В. Старичнев, М.А. Сребный и др. – М.: Московский государственный горный университет, 2002. – 471 с.
29. Балабышко А.М. Гидропривод механизированных крепей / А.М. Балабышко В.П. Ружицкий, К.М. Первов. – М.: Высшее горное образование, 2002. – 138 с.
30. Бубликов А.В. Автоматическое управление добычным комбайном в режиме стабилизации скорости подачи / А.В. Бубликов. – Д.: Збірник наукових праць Національного гірничого університету. – 2013. - №40. С. 91-97
31. Ключев В.И. Теория электропривода: Учеб. для вузов. – 2-е изд. перераб. и доп. / В.И. Ключев. – М.: Энергоатомиздат, 2001. – 704 с.
32. Иванов А.А. Теория автоматического управления и регулирования / А.А. Иванов. – М.: изд-во «Недра», 1970. – с. 352.
33. Никулин Е.А. Основы теории автоматического управления. Частотные методы анализа и синтеза / Учеб. пособие для вузов. – СПб.: БХВ-Петербург, 2004. – 640 с.
34. Дорф Р. Современные системы управления / Р. Дорф, Р. Бишоп; пер. С англ. Б.И. Копылова. – М.: Лаборатория Базовых Знаний, 2012. – 832 с.

**ДОДАТОК А - ТЕКСТИ ПРОГРАМ НАЛАШТУВАННЯ МЕРЕЖІ  
КОМП'ЮТЕРНОЇ СИСТЕМИ**

**Міністерство освіти і науки України**  
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**  
**“ДНІПРОВСЬКА ПОЛІТЕХНІКА”**

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**  
**НАЛАШТУВАННЯ МЕРЕЖІ КОМП'ЮТЕРНОЇ СИСТЕМИ**

Текст програми

804.02070743.20005-01 12 01

Листів 7

2020

## АНОТАЦІЯ

Даний документ містить ПЗ налаштувань маршрутизаторів Cisco для структурної схеми моделі комп'ютерної системи.

Тексти програм реалізовані на мові конфігураційних скриптів для мережного обладнання Cisco.

Середовище розробки та налагодження скриптів – пакет моделювання - мереж Cisco Packet Tracer в середовищі операційної системи Windows 7.

**ЗМІСТ**

	стор.	
1. Router4_startup	4	4
2. Switch4_startup-config	5	4
3. Прикордонний маршрутизатор_startup-config	6	5
4.		5
4.		7

**Скрипт налаштування Router4\_startup**

```
!  
version 15.1  
no service timestamps log datetime msec  
no service timestamps debug datetime msec  
no service password-encryption  
hostname Router  
ip dhcp excluded-address 192.168.4.1  
ip dhcp excluded-address 192.168.4.33  
ip dhcp excluded-address 192.168.4.65  
ip dhcp pool vlan10  
network 192.168.4.0 255.255.255.224  
default-router 192.168.4.1  
ip dhcp pool vlan20  
network 192.168.4.32 255.255.255.224  
default-router 192.168.4.33  
ip dhcp pool vlan30  
network 192.168.4.64 255.255.255.224  
default-router 192.168.4.65  
no ip cef  
no ipv6 cef  
license udi pid CISCO2911/K9 sn FTX1524BHOT-  
spanning-tree mode pvst  
interface GigabitEthernet0/0  
ip address 192.168.2.1 255.255.255.248  
duplex auto  
speed auto  
interface GigabitEthernet0/1  
ip address 192.168.3.1 255.255.255.248  
duplex auto  
speed auto  
interface GigabitEthernet0/2  
no ip address  
duplex auto  
speed auto  
interface GigabitEthernet0/2.10  
encapsulation dot1Q 10  
ip address 192.168.4.1 255.255.255.224  
interface GigabitEthernet0/2.20  
encapsulation dot1Q 20  
ip address 192.168.4.33 255.255.255.224  
interface GigabitEthernet0/2.30  
encapsulation dot1Q 30  
ip address 192.168.4.65 255.255.255.224  
interface Serial0/0/0  
ip address 209.104.19.2 255.255.255.252  
interface Serial0/0/1  
no ip address  
clock rate 2000000  
shutdown  
interface Vlan1  
no ip address
```

```
shutdown
router eigrp 200
 network 192.168.2.0 0.0.0.7
 network 192.168.3.0 0.0.0.7
 network 209.104.19.0 0.0.0.3
 network 192.168.4.0 0.0.0.127
ip classless
ip flow-export version 9
no cdp run
line con 0
line aux 0
line vty 0 4
 login
end
```

### **Скрипт налаштування Switch4\_startup-config**

```
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
hostname Switch
spanning-tree mode pvst
spanning-tree extend system-id
interface FastEthernet0/1
interface FastEthernet0/2
 switchport access vlan 10
 switchport mode access
interface FastEthernet0/3
 switchport access vlan 20
 switchport mode access
interface FastEthernet0/4
 switchport access vlan 30
 switchport mode access
interface FastEthernet0/5
interface FastEthernet0/6
interface FastEthernet0/7
interface FastEthernet0/8
interface FastEthernet0/9
interface FastEthernet0/10
interface FastEthernet0/11
interface FastEthernet0/12
interface FastEthernet0/13
interface FastEthernet0/14
interface FastEthernet0/15
interface FastEthernet0/16
interface FastEthernet0/17
interface FastEthernet0/18
interface FastEthernet0/19
interface FastEthernet0/20
interface FastEthernet0/21
interface FastEthernet0/22
interface FastEthernet0/23
```



```

interface FastEthernet0/24
interface GigabitEthernet0/1
interface GigabitEthernet0/2
interface Vlan1
  no ip address
  shutdown
line con 0
line vty 0 4
login
line vty 5 15
login
end

```

### **Скрипт налаштування Прикордонний маршрутизатор\_startup-config**

```

version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
hostname Router
ip cef
no ipv6 cef
license udi pid CISCO2901/K9 sn FTX15242A76-
spanning-tree mode pvst
interface GigabitEthernet0/0
  ip address 192.168.1.1 255.255.255.248
  ip nat inside
  duplex auto
  speed auto
interface GigabitEthernet0/1
  ip address 192.168.0.1 255.255.255.248
  ip nat inside
  duplex auto
  speed auto
interface Serial0/3/0
  ip address 209.104.19.1 255.255.255.252
  ip nat inside
  clock rate 2000000
interface Serial0/3/1
  ip address 209.165.202.1 255.255.255.252
  ip nat outside
  clock rate 2000000
interface Vlan1
  no ip address
  shutdown
router eigrp 200
  redistribute static
  network 192.168.0.0 0.0.0.7
  network 192.168.1.0 0.0.0.7
  network 209.104.19.0 0.0.0.3
ip nat inside source list 10 interface Serial0/3/1 overload
ip classless
ip route 0.0.0.0 0.0.0.0 209.165.202.2

```

```
ip flow-export version 9
access-list 10 permit 192.168.0.0 0.0.255.255
line con 0
line aux 0
line vty 0 4
login
end
```

**ДОДАТОК Б - ТЕКСТИ ПРОГРАМ РЕАЛІЗАЦІЇ МАТЕМАТИЧНОЇ  
МОДЕЛІ КОМП'ЮТЕРНОЇ МЕРЕЖІ ЯК ЗАМКНУТОЇ СИСТЕМИ  
МАСОВОГО ОБСЛУГОВУВАННЯ**

**Міністерство освіти і науки України**  
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**  
**“ДНІПРОВСЬКА ПОЛІТЕХНІКА”**

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**  
**ПРОГРАМА МАТЕМАТИЧНОЇ МОДЕЛІ МЕРЕЖІ**  
**КОМП'ЮТЕРНОЇ СИСТЕМИ**

Текст програми

804.02070743.20005-01 12 01

Листів 18

**2020**

## АНОТАЦІЯ

Даний документ містить ПЗ реалізації математичної моделі комп'ютерної мережі.

Тексти програм реалізовані в середовищі Matchcfdв середовищі операційної системи Windows 10..

Програма реалізує рекурентний метод Бузена для розрахунку параметрів комп'ютерної мережі як замкнутої системи масового обслуговування.

**ЗМІСТ**

			стор.
1.	Перелік використаних змінних	4	
2.	Текст програми	6	6
2.	Результати розрахунку	16	7

## 1 Перелік використаних змінних, та переклад коментарів

$N_n$  – кількість вузлів мережі.

$\tau$  – час обробки одного пакета у вузлі.

$P_r$  – матриця перехідних ймовірностей.

$e$  – матриця перехідних коефіцієнтів.

$m$  – кількість конвеєрів у вузлах.

$N$  – кількість пакетів що циркулюють в мережі.

$B$  – матриця ймовірностей черги у вузлах.

$\lambda$  – середня інтенсивність запитів на вході у вузол.

$L$  – середня черга пакетів у вузлі.

$t$  – середній час перебування пакета у вузлі.

Average time spent in the node - Середній час перебування у вузлі.

Calculation of function A - Розрахунок функції A.

Calculation of the matrix of constants T - Обчислення матриці констант T.

Calculate the constants for the second and subsequent nodes - Розраховуємо константи для другого і наступних вузлів.

Calculation of the intensity of query processing in network nodes - Розрахунок інтенсивності обробки запитів у вузлах мережі.

Calculation of auxiliary coefficients - Розрахунок допоміжних коефіцієнтів.

Calculation of probabilities of receipt in the last node of applications – j - Розрахунок ймовірностей надходження в останньому вузлі заявок – j.

Determination of transmission coefficients - Визначення коефіцієнтів передачі.

Matrix of transfer coefficients - Матриця передаточних коефіцієнтів.

Number of nodes in the network - Кількість вузлів у мережі.

Transfer matrix - Передаточна матрица.

The number of packets circulating in the network - Кількість пакетів, які циркулюють в мережі.

The number of pipelines in each node - Кількість конвеєрів в кожному вузлі.

The intensity of the input stream - Інтенсивність вхідного потоку.

The average number of packets per node - Середня кількість пакетів в вузлах.

Packet processing time at the node - Час обробки пакета у вузлі.



## 2 Текст програми

Number of nodes in the network 10

$Nn := 9$

$i := 0..Nn$      $j := 0..Nn$

### Packet processing time at the node

$\tau_i :=$

6
6
6
6
6
6
6
6
6
6
6

$\tau =$

	0
0	6
1	6
2	6
3	6
4	6
5	6
6	6
7	6
8	6
9	6

Calculation of the intensity of request processing in network nodes

$$\mu_i := \frac{1}{\tau_i}$$

Transfer matrix

$\mu =$

	0
0	0.167
1	0.167
2	0.167
3	0.167
4	0.167
5	0.167
6	0.167
7	0.167
8	0.167
9	0.167

Route matrix

$Pr :=$

0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0.2	0.3	0	0.2	0.3	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0.4	0	0	0.2	0.2	0.2	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0.6	0	0	0	0.2	0.2
0	0	0	0	0	0	0	0.6	0	0.4
0	0	0	0	0	0	0	0.6	0.4	0

$$\text{SumPr}_1 := \sum_{j=0}^{Nn} Pr_{i,j}$$

$\text{SumPr} =$

	0
0	1
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1

Determination of transmission coefficients

$$P := P_r^T$$

$$P =$$

	0	1	2	3	4	5	6	7	8	9
0	0	0	0.2	0	0	0	0	0	0	0
1	0	0	0.3	0	0	0	0	0	0	0
2	1	1	0	1	0.4	0	0	0	0	0
3	0	0	0.2	0	0	0	0	0	0	0
4	0	0	0.3	0	0	1	1	0.6	0	0
5	0	0	0	0	0.2	0	0	0	0	0
6	0	0	0	0	0.2	0	0	0	0	0
7	0	0	0	0	0.2	0	0	0	0.6	0.6
8	0	0	0	0	0	0	0	0.2	0	0.4
9	0	0	0	0	0	0	0	0.2	0.4	0

$$D :=$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$P1 := P - D$$

$$P1 =$$

	0	1	2	3	4	5	6	7	8	9
0	-1	0	0.2	0	0	0	0	0	0	0
1	0	-1	0.3	0	0	0	0	0	0	0
2	1	1	-1	1	0.4	0	0	0	0	0
3	0	0	0.2	-1	0	0	0	0	0	0
4	0	0	0.3	0	-1	1	1	0.6	0	0
5	0	0	0	0	0.2	-1	0	0	0	0
6	0	0	0	0	0.2	0	-1	0	0	0
7	0	0	0	0	0.2	0	0	-1	0.6	0.6
8	0	0	0	0	0	0	0	0.2	-1	0.4
9	0	0	0	0	0	0	0	0.2	0.4	-1

$$j := 1..Nn \quad i := 0..Nn$$

$$P2_{(j-1),i} := P1_{0,i} + P1_{j,i}$$

$$P2 =$$

	0	1	2	3	4	5	6	7	8	9
0	-1	-1	0.5	0	0	0	0	0	0	0
1	0	1	-0.8	1	0.4	0	0	0	0	0
2	-1	0	0.4	-1	0	0	0	0	0	0
3	-1	0	0.5	0	-1	1	1	0.6	0	0
4	-1	0	0.2	0	0.2	-1	0	0	0	0
5	-1	0	0.2	0	0.2	0	-1	0	0	0
6	-1	0	0.2	0	0.2	0	0	-1	0.6	0.6
7	-1	0	0.2	0	0	0	0	0.2	-1	0.4
8	-1	0	0.2	0	0	0	0	0.2	0.4	-1

$$j := 0..Nn - 1 \quad i := 0..Nn - 1 \quad PP2_{j,i} := P2_{j,i+1}$$

$$PP2 = \begin{pmatrix} -1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -0.8 & 1 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & -1 & 1 & 1 & 0.6 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0.2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0.2 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0.2 & 0 & 0 & -1 & 0.6 & 0.6 & 0 \\ 0 & 0.2 & 0 & 0 & 0 & 0 & 0.2 & -1 & 0.4 & 0 \\ 0 & 0.2 & 0 & 0 & 0 & 0 & 0.2 & 0.4 & -1 & 0 \end{pmatrix}$$

$$Q_{j,0} := P2_{j,0}$$

$$Q = \begin{pmatrix} -1 \\ 0 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix}$$

$$E := \text{lsolve}(PP2, Q)$$

$$E = \begin{pmatrix} -1.5 \\ -5 \\ -1 \\ -3.75 \\ -0.75 \\ -0.75 \\ -1.25 \\ -0.417 \\ -0.417 \end{pmatrix}$$

Matrix of transfer coefficients  $N \times 1$

$$e := \begin{pmatrix} 1 \\ 1.5 \\ 5 \\ 1 \\ 3.75 \\ 0.75 \\ 0.75 \\ 1.25 \\ 0.417 \\ 0.417 \end{pmatrix}$$

**The number of packets circulating in the network**

$$N := 5$$

$$i := 0..Nn \quad j := 0..N-1$$

The number of pipelines in each node

$m :=$

1
1
1
1
1
1
1
1
1
1

$m =$

	0
0	1
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1

Calculation of function A

$$A_{i,j} := \begin{cases} j! & \text{if } m_1 \geq N-1 \\ 1 & \text{if } m_1 = 1 \\ j! & \text{if } 1 < m_1 < N-1 \wedge j \leq m_1 \\ m_1! \cdot \binom{j}{m_1}^{j-m_1} & \text{if } 1 < m_1 < N-1 \wedge j > m_1 \end{cases}$$

A =

	0	1	2	3	4
0	1	1	1	1	1
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	1	1	1
5	1	1	1	1	1
6	1	1	1	1	1
7	1	1	1	1	1
8	1	1	1	1	1
9	1	1	1	1	1

$$X_1 := \frac{e_1}{\mu_1}$$

X =

	0
0	6
1	9
2	30
3	6
4	22.5
5	4.5
6	4.5
7	7.5
8	2.502
9	2.502

Calculation of the matrix of constants T

$$T_{i,j} := \frac{(X_1)^j}{A_{i,j}} \quad T_{i,0} := 1$$

	0	1	2	3	4
0	1	6	36	216	$1.296 \cdot 10^3$
1	1	9	81	729	$6.561 \cdot 10^3$
2	1	30	900	$2.7 \cdot 10^4$	$8.1 \cdot 10^5$
3	1	6	36	216	$1.296 \cdot 10^3$
4	1	22.5	506.25	$1.139 \cdot 10^4$	$2.563 \cdot 10^5$
5	1	4.5	20.25	91.125	410.063
6	1	4.5	20.25	91.125	410.063
7	1	7.5	56.25	421.875	$3.164 \cdot 10^3$
8	1	2.502	6.26	15.663	39.188
9	1	2.502	6.26	15.663	39.188

Calculate the constants for the second and subsequent nodes

$$i := 1..Nn \quad k := 0..N-1$$

$$G_{0,j} := T_{0,j}$$

$$G_{i,k} := \sum_{j=0}^k (T_{i,j} \cdot G_{i-1,k-j})$$

	0	1	2	3	4
0	1	6	36	216	$1.296 \cdot 10^3$
1	1	15	171	$1.755 \cdot 10^3$	$1.709 \cdot 10^4$
2	1	45	$1.521 \cdot 10^3$	$4.739 \cdot 10^4$	$1.439 \cdot 10^6$
3	1	51	$1.827 \cdot 10^3$	$5.835 \cdot 10^4$	$1.789 \cdot 10^6$
4	1	73.5	$3.481 \cdot 10^3$	$1.367 \cdot 10^5$	$4.864 \cdot 10^6$
5	1	78	$3.832 \cdot 10^3$	$1.539 \cdot 10^5$	$5.556 \cdot 10^6$
6	1	82.5	$4.203 \cdot 10^3$	$1.728 \cdot 10^5$	$6.334 \cdot 10^6$
7	1	90	$4.878 \cdot 10^3$	$2.094 \cdot 10^5$	$7.904 \cdot 10^6$
8	1	92.502	$5.109 \cdot 10^3$	$2.222 \cdot 10^5$	$8.46 \cdot 10^6$
9	1	95.004	$5.347 \cdot 10^3$	$2.356 \cdot 10^5$	$9.05 \cdot 10^6$

Calculation of probabilities of receipt in the last node of applications - j

$$B_{Nn,j} := \frac{T_{Nn,j}}{G_{Nn,N-1}} \cdot G_{Nn,N-1-j}$$

$$B_{Nn,0} := 1 - B_{Nn,1}$$

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0.935	0.065	$3.699 \cdot 10^{-3}$	$1.644 \cdot 10^{-4}$	$4.33 \cdot 10^{-6}$

Calculation of auxiliary coefficients

$$G_n = \begin{pmatrix} 1 & 89.004 & 4.777 \times 10^3 & 2.035 \times 10^5 & 7.636 \times 10^6 \\ 1 & 86.004 & 4.492 \times 10^3 & 1.874 \times 10^5 & 6.93 \times 10^6 \\ 1 & 65.004 & 2.497 \times 10^3 & 7.515 \times 10^4 & 1.983 \times 10^6 \\ 1 & 89.004 & 4.777 \times 10^3 & 2.035 \times 10^5 & 7.636 \times 10^6 \\ 1 & 72.504 & 3.21 \times 10^3 & 1.153 \times 10^5 & 3.75 \times 10^6 \\ 1 & 90.504 & 4.92 \times 10^3 & 2.115 \times 10^5 & 7.99 \times 10^6 \\ 1 & 90.504 & 4.92 \times 10^3 & 2.115 \times 10^5 & 7.99 \times 10^6 \\ 1 & 87.504 & 4.635 \times 10^3 & 1.955 \times 10^5 & 7.283 \times 10^6 \end{pmatrix} \quad G_{n,0} := 1$$

$$B_{i,j} := \frac{T_{i,j}}{G_{Nn,N-1}} G_{n,N-1-j}$$

	0	1	2	3	4
0	0.844	0.135	0.019	$2.124 \cdot 10^{-3}$	$1.432 \cdot 10^{-4}$
1	0.766	0.186	0.04	$6.928 \cdot 10^{-3}$	$7.25 \cdot 10^{-4}$
2	0.219	0.249	0.248	0.194	0.09
3	0.844	0.135	0.019	$2.124 \cdot 10^{-3}$	$1.432 \cdot 10^{-4}$
4	0.414	0.287	0.18	0.091	0.028
5	0.883	0.105	0.011	$9.113 \cdot 10^{-4}$	$4.531 \cdot 10^{-5}$
6	0.883	0.105	0.011	$9.113 \cdot 10^{-4}$	$4.531 \cdot 10^{-5}$
7	0.805	0.162	0.029	$4.079 \cdot 10^{-3}$	$3.496 \cdot 10^{-4}$
8	0.935	0.061	$3.534 \cdot 10^{-3}$	$1.601 \cdot 10^{-4}$	$4.33 \cdot 10^{-6}$
9	0.935	0.065	$3.699 \cdot 10^{-3}$	$1.644 \cdot 10^{-4}$	$4.33 \cdot 10^{-6}$

$$i := 0..Nn$$

$$j := 0..N -$$

$$\text{Sum}B_i := \sum_j B_{i,j}$$

	0
0	1
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1.004

SumB =

$$\lambda_i := e_i \cdot \frac{G_{Nn-1, N-2}}{G_{Nn, N-1}}$$

$$L_i := \sum_{n=0}^{N-1} (n \cdot B_{i,n})$$

The intensity of the input stream

The average number of packets per node

	0
0	0.025
1	0.037
2	0.123
3	0.025
4	0.092
5	0.018
6	0.018
7	0.031
8	0.01
9	0.01

$\lambda =$

	0
0	0.18
1	0.291
2	1.686
3	0.18
4	1.033
5	0.13
6	0.13
7	0.233
8	0.069
9	0.073

$L =$

$$t_i := \frac{L_i}{\lambda_i}$$

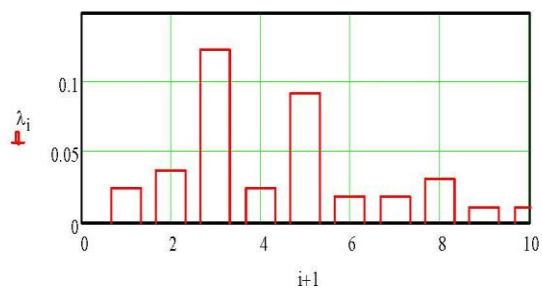
Average time spent in the node

	0
0	7.326
1	7.888
2	13.731
3	7.326
4	11.217
5	7.065
6	7.065
7	7.6
8	6.739
9	7.134

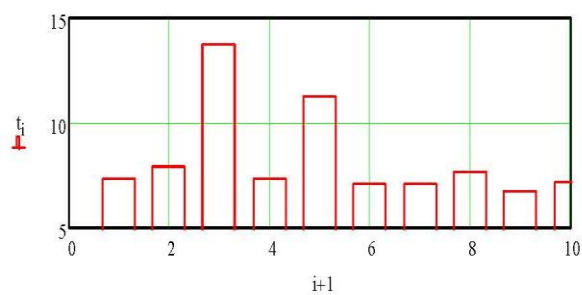
$t =$



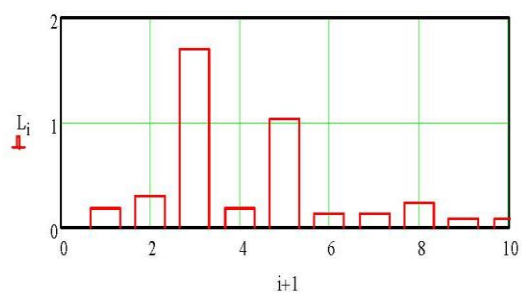
### 3 Результати розрахунку



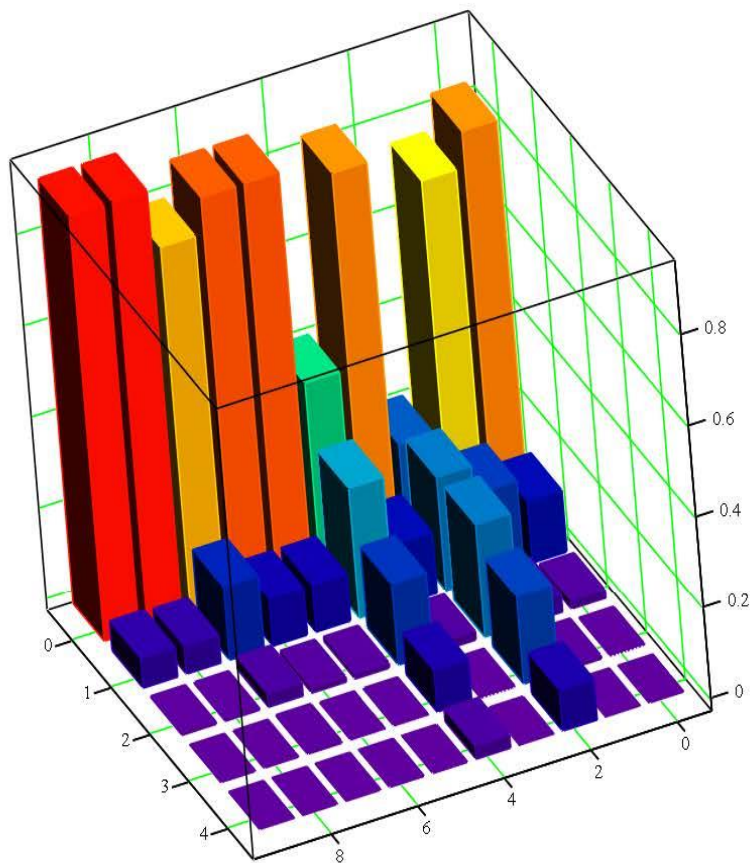
The intensity of the flow entering the node



The average residence time of a packet in a node

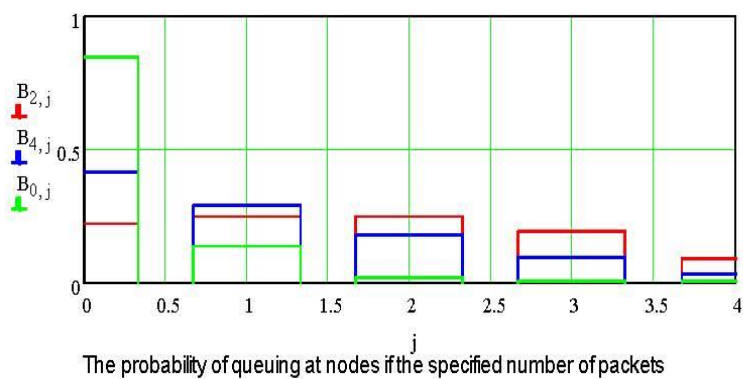


The average number of packets in a node



B

The probability of queuing at network nodes



## ДОДАТОК В – ПРОГРАМНІ МОДУЛІ СИСТЕМИ УПРАВЛІННЯ

## Додаток В.1 Модуль об'єкта управління

```

(*)
*
* File: Model_SystemKNS_P_Maximum_SensorH_Z.exp
*
* IEC 61131-3 Structured Text (ST) code generated for subsystem
"Model_SystemKNS_P_Maximum_SensorH_Z/Об'єкт управління"
*
* Model name                : Model_SystemKNS_P_Maximum_SensorH_Z
* Model version             : 1.332
* Model creator             : bukap
* Model last modified by   : bukap
* Model last modified on   : Wed Oct 18 13:00:41 2017
* Model sample time        : 0.1s
* Subsystem name           : Model_SystemKNS_P_Maximum_SensorH_Z/Об'єкт управління
* Subsystem sample time    : 0.1s
* Simulink PLC Coder version : 1.6 (R2013b) 08-Aug-2013
* ST code generated on     : Wed Oct 18 13:00:45 2017
*
* Target IDE selection     : 3S CoDeSys 2.3
* Test Bench included      : No
*
*)
FUNCTION_BLOCK sfun_target
VAR_INPUT
    ssMethodType: SINT;
    bi_n1: LREAL;
    bi_n2: LREAL;
END_VAR
VAR_OUTPUT
    b_SEL: BOOL;
    b_SHL: BOOL;
    SSL: BOOL;
    SML: BOOL;
    SLL: BOOL;
END_VAR
VAR
    DiscreteTimeIntegrator_DSTATE: LREAL;
    Memory_PreviousInput: LREAL;
END_VAR
VAR_TEMP
END_VAR
CASE ssMethodType OF
    SS_INITIALIZE:
        (* InitializeConditions for Memory: '<S1>/Memory' *)
        Memory_PreviousInput := 0.0;

        (* InitializeConditions for Atomic SubSystem: '<S1>/Приемный резервуар' *)
        (* InitializeConditions for DiscreteIntegrator: '<S4>/Discrete-Time Integrator' *)
        DiscreteTimeIntegrator_DSTATE := 1.0;
        (* End of InitializeConditions for SubSystem: '<S1>/Приемный резервуар' *)
    SS_STEP:
        (* Outputs for Atomic SubSystem: '<S1>/Приемный резервуар' *)

        (* DiscreteIntegrator: '<S4>/Discrete-Time Integrator' *)
        IF DiscreteTimeIntegrator_DSTATE >= 10.0 THEN
            DiscreteTimeIntegrator_DSTATE := 10.0;
        ELSIF DiscreteTimeIntegrator_DSTATE <= 0.0 THEN
            DiscreteTimeIntegrator_DSTATE := 0.0;
        END_IF;

        (* Output: '<Root>/SEL' incorporates:
        * Constant: '<S4>/Constant1'
        * DiscreteIntegrator: '<S4>/Discrete-Time Integrator'
        * RelationalOperator: '<S4>/Relational Operator' *)
        b_SEL := DiscreteTimeIntegrator_DSTATE >= 9.0;

        (* Output: '<Root>/SHL' incorporates:
        * Constant: '<S4>/Constant2'
        * DiscreteIntegrator: '<S4>/Discrete-Time Integrator'
        * RelationalOperator: '<S4>/Relational Operator1' *)
        b_SHL := DiscreteTimeIntegrator_DSTATE >= 8.0;

```

```

(* Output: '<Root>/SML' incorporates:
 * Constant: '<S4>/Constant3'
 * DiscreteIntegrator: '<S4>/Discrete-Time Integrator'
 * RelationalOperator: '<S4>/Relational Operator2' *)
SML := DiscreteTimeIntegrator_DSTATE >= 5.0;

(* Output: '<Root>/SLL' incorporates:
 * Constant: '<S4>/Constant4'
 * DiscreteIntegrator: '<S4>/Discrete-Time Integrator'
 * RelationalOperator: '<S4>/Relational Operator3' *)
SLL := DiscreteTimeIntegrator_DSTATE >= 2.0;

(* Output: '<Root>/SSL' incorporates:
 * Constant: '<S4>/Constant5'
 * DiscreteIntegrator: '<S4>/Discrete-Time Integrator'
 * RelationalOperator: '<S4>/Relational Operator4' *)
SSL := DiscreteTimeIntegrator_DSTATE >= 5.1;

(* Update for DiscreteIntegrator: '<S4>/Discrete-Time Integrator' incorporates:
 * Constant: '<S4>/Constant'
 * Constant: '<S4>/л в м^3'
 * Constant: '<S5>/Constant'
 * Memory: '<S1>/Memory'
 * Product: '<S4>/Пересчет л//с в м^3//с'
 * Product: '<S4>/Пересчет расхода в уровень'
 * Sum: '<S1>/Sum' *)
DiscreteTimeIntegrator_DSTATE := (((35.0 - Memory_PreviousInput) / 1000.0) /
7.0685834705770345) * 0.1) + DiscreteTimeIntegrator_DSTATE;
IF DiscreteTimeIntegrator_DSTATE >= 10.0 THEN
    DiscreteTimeIntegrator_DSTATE := 10.0;
ELSIF DiscreteTimeIntegrator_DSTATE <= 0.0 THEN
    DiscreteTimeIntegrator_DSTATE := 0.0;
END_IF;
(* End of Update for DiscreteIntegrator: '<S4>/Discrete-Time Integrator' *)
(* End of Outputs for SubSystem: '<S1>/Приемный резервуар' *)

(* Update for Memory: '<S1>/Memory' incorporates:
 * Gain: '<S2>/Gain'
 * Gain: '<S3>/Gain'
 * Update for Inport: '<Root>/bi_n1'
 * Update for Inport: '<Root>/bi_n2'
 * Sum: '<S1>/Sum1' *)
Memory_PreviousInput := (0.020689655172413793 * bi_n1) + (0.020689655172413793 * bi_n2);
END_CASE;
END_FUNCTION_BLOCK
VAR_GLOBAL CONSTANT
    SS_INITIALIZE: SINT := 0;
    SS_STEP: SINT := 1;
END_VAR
VAR_GLOBAL
END_VAR

```

## В.2 Модуль регулятора

```

(*)
*
* File: Model_SystemKNS_P_Maximum_SensorH_Z.exp
*
* IEC 61131-3 Structured Text (ST) code generated for subsystem
"Model_SystemKNS_P_Maximum_SensorH_Z/Непрерывный регулятор
с максимальной скоростью
откачки при работе
двух насосов"
*
* Model name                : Model_SystemKNS_P_Maximum_SensorH_Z
* Model version              : 1.333
* Model creator              : bukap
* Model last modified by    : bukap
* Model last modified on    : Wed Oct 18 13:01:12 2017
* Model sample time         : 0.1s
* Subsystem name            : Model_SystemKNS_P_Maximum_SensorH_Z/Непрерывный регулятор
с максимальной скоростью
откачки при работе
двух насосов
* Subsystem sample time    : 0.1s
* Simulink PLC Coder version : 1.6 (R2013b) 08-Aug-2013
* ST code generated on     : Wed Oct 18 13:01:54 2017
*

```

```

* Target IDE selection      : 3S CoDeSys 2.3
* Test Bench included      : No
*
*)
FUNCTION_BLOCK sfun_target
VAR_INPUT
    ssMethodType: SINT;
    b_SEL: BOOL;
    b_SHL: BOOL;
    SSL: BOOL;
    SML: BOOL;
    SLL: BOOL;
END_VAR
VAR_OUTPUT
    Pump1: LREAL;
    Pump2: LREAL;
END_VAR
VAR
    Memory1_2_PreviousInput: BOOL;
    Memory1_3_PreviousInput: BOOL;
    Memory1_4_PreviousInput: BOOL;
    Memory_PreviousInput: LREAL;
    Memory3_4_PreviousInput: BOOL;
    Memory3_5_PreviousInput: BOOL;
    Memory2_2_PreviousInput: BOOL;
    Memory2_4_PreviousInput: BOOL;
    hStateUpdate: LREAL;
    hState: LREAL;
    initialize: LREAL;
    b_time: LREAL;
    statePump_j: LREAL;
    statePump: LREAL;
END_VAR
VAR_TEMP
    rtb_Sum3: LREAL;
END_VAR
CASE ssMethodType OF
    SS_INITIALIZE:

        (* InitializeConditions for Memory: '<S1>/Memory1' *)
        Memory1_2_PreviousInput := FALSE;

        (* InitializeConditions for Memory: '<S1>/Memory1' *)
        Memory1_3_PreviousInput := FALSE;

        (* InitializeConditions for Memory: '<S1>/Memory1' *)
        Memory1_4_PreviousInput := FALSE;

        (* InitializeConditions for MATLAB Function: '<S1>/MATLAB Function' *)
        initialize := 0.0;
        b_time := 0.0;
        hState := 0.0;
        hStateUpdate := 0.0;

        (* InitializeConditions for Memory: '<S1>/Memory' *)
        Memory_PreviousInput := 0.0;

        (* InitializeConditions for Memory: '<S1>/Memory3' *)
        Memory3_4_PreviousInput := FALSE;

        (* InitializeConditions for Memory: '<S1>/Memory3' *)
        Memory3_5_PreviousInput := FALSE;

        (* InitializeConditions for MATLAB Function: '<S3>/MATLAB Function' *)
        statePump_j := 0.0;

        (* InitializeConditions for Memory: '<S1>/Memory2' *)
        Memory2_2_PreviousInput := FALSE;

        (* InitializeConditions for Memory: '<S1>/Memory2' *)
        Memory2_4_PreviousInput := FALSE;

        (* InitializeConditions for MATLAB Function: '<S4>/MATLAB Function' *)
        statePump := 0.0;
    SS_STEP:

        (* MATLAB Function: '<S1>/MATLAB Function' incorporates:
        * Constant: '<S1>/Constant'
        * Constant: '<S1>/Constant1'
        * Constant: '<S1>/Constant4'

```

```

* Memory: '<S1>/Memory1' *)
(* MATLAB Function 'Непрерывный регулятор с максимальной скоростью откачки при работе двух
насосов/MATLAB Function': '<S2>:1' *)
(* HSHL - Уровень размещения датчика верхнего уровня *)
(* HSSL - Уровень размещения датчика скорости *)
(* HSML - Уровень размещения датчика среднего уровня *)
(* SHL - Состояние датчика верхнего уровня *)
(* SSL - Состояние датчика скорости *)
(* SML - Состояние датчика нижнего уровня *)
(* Локальные статические переменные *)
(* Инициализация *)
(* Счетчик времени *)
(* Уровень воды *)
(* Приращение уровня воды *)
IF BOOL_TO_DINT(Memory1_3_PreviousInput) = 0 THEN
    (* '<S2>:1:22' *)
    (* Вода ниже датчика скорости *)
    IF BOOL_TO_DINT(Memory1_4_PreviousInput) = 0 THEN
        (* '<S2>:1:23' *)
        (* Вода ниже датчика среднего уровня *)
        (* '<S2>:1:24' *)
        initialize := 0.0;
        (* '<S2>:1:25' *)
        b_time := 0.0;
        (* '<S2>:1:26' *)
        hState := 0.0;
        (* '<S2>:1:27' *)
        hStateUpdate := 0.0;
    ELSIF initialize = 0.0 THEN
        (* '<S2>:1:28' *)
        (* Вода выше датчика среднего уровня *)
        (* '<S2>:1:29' *)
        initialize := 1.0;
        (* Запуск инициализации *)
    ELSE
        (* '<S2>:1:31' *)
        b_time := b_time + 1.0;
        (* Увеличение счетчика времени *)
    END_IF;
ELSE
    (* Вода выше датчика скорости *)
    IF initialize = 1.0 THEN
        (* '<S2>:1:34' *)
        (* Инициализация *)
        (* '<S2>:1:35' *)
        hState := 5.1;
        (* Уровень воды соответствует уровню датчика скорости *)
        (* '<S2>:1:36' *)
        hStateUpdate := 0.099999999999999645 / b_time;
        (* Расчет приращения *)
        (* '<S2>:1:37' *)
        initialize := 0.0;
    ELSIF BOOL_TO_DINT(Memory1_2_PreviousInput) = 1 THEN
        (* '<S2>:1:38' *)
        (* Вода выше датчика верхнего уровня *)
        (* '<S2>:1:39' *)
        hState := 8.0;
        (* Уровень воды соответствует уровню датчика верхнего уровня *)
        (* '<S2>:1:40' *)
        hStateUpdate := 0.0;
    ELSIF hState = 0.0 THEN
        (* '<S2>:1:41' *)
        (* Система включилась когда вода была выше датчика скорости, но ниже датчика
верхнего уровня *)
        (* '<S2>:1:42' *)
        hState := 6.5;
        (* Принять уровень воды по середине между датчиками среднего и верхнего уровня *)
    ELSIF hState <> 0.0 THEN
        (* '<S2>:1:43' *)
        (* Вода выше датчика среднего уровня, но ниже датчика верхнего уровня *)
        (* '<S2>:1:44' *)
        hState := hState + hStateUpdate;
        (* Расчет уровня воды *)
        IF hState > 8.0 THEN
            (* '<S2>:1:45' *)
            (* Уровень воды выше уровня размещения датчика верхнего уровня *)
            (* '<S2>:1:46' *)
            hState := 8.0;
            (* Уровень воды соответствует уровню датчика верхнего уровня *)
        END_IF;
    END_IF;
END_IF;

```

```

ELSE
    (* Вода были выше датчика верхнего уровня, а теперь опустились ниже его *)
    (* '<S2>:1:49' *)
    hState := 0.0;
END_IF;
END_IF;
(* '<S2>:1:53' *)
(* Уровень воды *)

(* MATLAB Function: '<S3>/MATLAB Function' incorporates:
 * Memory: '<S1>/Memory3' *)
(* MATLAB Function 'Непрерывный регулятор с максимальной скоростью откачки при работе двух
насосов/Непрерывное управление насосным агрегатом/MATLAB Function': '<S5>:1' *)
(* SML - Датчик среднего уровня *)
(* SLL - Датчик нижнего уровня *)
(* pump - Сигнал включения/выключения насосного агрегата *)
(* Локальные статические переменные *)
(* Состояние насосного агрегата *)
IF statePump_j = 0.0 THEN
    (* '<S5>:1:13' *)
    (* Насосный агрегат выключен *)
    IF BOOL_TO_DINT(Memory3_4_PreviousInput) = 1 THEN
        (* '<S5>:1:14' *)
        (* Вода достигла среднего уровня *)
        (* '<S5>:1:15' *)
        statePump_j := 1.0;
        (* Включение насосного агрегата *)
    END_IF;
ELSE
    (* Насосный агрегат включен *)
    IF BOOL_TO_DINT(Memory3_5_PreviousInput) = 0 THEN
        (* '<S5>:1:18' *)
        (* Води опустились ниже нижнего уровня *)
        (* '<S5>:1:19' *)
        statePump_j := 0.0;
        (* Выключение насосного агрегата *)
    END_IF;
END_IF;
(* '<S5>:1:23' *)
(* Управление насосным агрегатом *)

(* Switch: '<S3>/Switch' incorporates:
 * MATLAB Function: '<S3>/MATLAB Function' *)
IF statePump_j <> 0.0 THEN

    (* Switch: '<S3>/Switch1' incorporates:
 * Constant: '<S3>/Constant5'
 * Memory: '<S1>/Memory' *)
    IF Memory_PreviousInput <> 0.0 THEN
        rtb_Sum3 := 1450.0;
    ELSE

        (* Gain: '<S1>/Пропорциональный регулятор' incorporates:
 * Constant: '<S1>/Constant2'
 * MATLAB Function: '<S1>/MATLAB Function'
 * Sum: '<S1>/Sum2' *)
        rtb_Sum3 := (5.0 - hState) * -52.0;

        (* Saturate: '<S1>/Ограничение управляющего сигнала' *)
        IF rtb_Sum3 >= 100.0 THEN
            rtb_Sum3 := 100.0;
        ELSIF NOT (rtb_Sum3 > 0.0) THEN
            rtb_Sum3 := 0.0;
        END_IF;
        (* End of Saturate: '<S1>/Ограничение управляющего сигнала' *)

        (* Sum: '<S3>/Sum3' incorporates:
 * Constant: '<S3>/Constant3'
 * Gain: '<S3>/Gain1' *)
        rtb_Sum3 := (5.8 * rtb_Sum3) + 870.0;

        (* Saturate: '<S3>/Ограничение управляющего воздействия' *)
        IF rtb_Sum3 >= 1450.0 THEN
            rtb_Sum3 := 1450.0;
        ELSIF NOT (rtb_Sum3 > 870.0) THEN
            rtb_Sum3 := 870.0;
        END_IF;
        (* End of Saturate: '<S3>/Ограничение управляющего воздействия' *)
    END_IF;
END_IF;
(* End of Switch: '<S3>/Switch1' *)

```

```

    (* Output: '<Root>/Pump1' *)
    Pump1 := rtb_Sum3;
ELSE
    (* Output: '<Root>/Pump1' incorporates:
    * Constant: '<S3>/Constant1' *)
    Pump1 := 0.0;
END_IF;
(* End of Switch: '<S3>/Switch' *)

(* MATLAB Function: '<S4>/MATLAB Function' incorporates:
* Memory: '<S1>/Memory2' *)
(* MATLAB Function 'Непрерывный регулятор с максимальной скоростью откачки при работе двух
насосов/Релейное управление насосным агрегатом/MATLAB Function': '<S6>:1' *)
(* SHL - Датчик верхнего уровня *)
(* SML - Датчик среднего уровня *)
(* pump - Сигнал включения/выключения насосного агрегата *)
(* Локальные статические переменные *)
(* Состояние насосного агрегата *)
IF statePump = 0.0 THEN
    (* '<S6>:1:13' *)
    (* Насосный агрегат выключен *)
    IF BOOL_TO_DINT(Memory2_2_PreviousInput) = 1 THEN
        (* '<S6>:1:14' *)
        (* Воды достигли верхнего уровня *)
        (* '<S6>:1:15' *)
        statePump := 1.0;
        (* Включение насосного агрегата *)
    END_IF;
ELSE
    (* Насосный агрегат включен *)
    IF BOOL_TO_DINT(Memory2_4_PreviousInput) = 0 THEN
        (* '<S6>:1:18' *)
        (* Воды опустились ниже среднего уровня *)
        (* '<S6>:1:19' *)
        statePump := 0.0;
        (* Выключение насосного агрегата *)
    END_IF;
END_IF;
(* '<S6>:1:23' *)
(* Управление насосным агрегатом *)

(* Switch: '<S4>/Switch' incorporates:
* Constant: '<S4>/Constant1'
* Constant: '<S4>/Constant3'
* MATLAB Function: '<S4>/MATLAB Function' *)
IF statePump <> 0.0 THEN
    rtb_Sum3 := 1450.0;
ELSE
    rtb_Sum3 := 0.0;
END_IF;
(* End of Switch: '<S4>/Switch' *)

(* Output: '<Root>/Pump2' *)
Pump2 := rtb_Sum3;

(* Update for Memory: '<S1>/Memory1' incorporates:
* Update for Inport: '<Root>/SHL' *)
Memory1_2_PreviousInput := b_SHL;

(* Update for Memory: '<S1>/Memory1' incorporates:
* Update for Inport: '<Root>/SSL' *)
Memory1_3_PreviousInput := SSL;

(* Update for Memory: '<S1>/Memory1' incorporates:
* Update for Inport: '<Root>/SML' *)
Memory1_4_PreviousInput := SML;

(* Update for Memory: '<S1>/Memory' *)
Memory_PreviousInput := rtb_Sum3;

(* Update for Memory: '<S1>/Memory3' incorporates:
* Update for Inport: '<Root>/SML' *)
Memory3_4_PreviousInput := SML;

(* Update for Memory: '<S1>/Memory3' incorporates:
* Update for Inport: '<Root>/SLL' *)
Memory3_5_PreviousInput := SLL;

(* Update for Memory: '<S1>/Memory2' incorporates:

```



```
      * Update for Inport: '<Root>/SHL' *)
      Memory2_2_PreviousInput := b_SHL;

      (* Update for Memory: '<S1>/Memory2' incorporates:
      * Update for Inport: '<Root>/SML' *)
      Memory2_4_PreviousInput := SML;
END_CASE;
END_FUNCTION_BLOCK
VAR_GLOBAL CONSTANT
  SS_INITIALIZE: SINT := 0;
  SS_STEP: SINT := 1;
END_VAR
VAR_GLOBAL
END_VAR
```

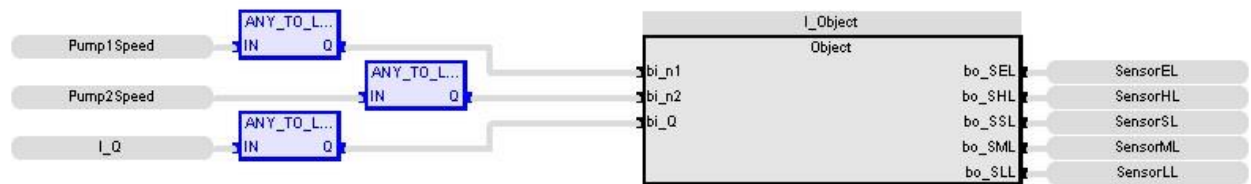
## ДОДАТОК Г – ПРОГРАМНЕ ОБЕСПЕЧЕННЯ СИСТЕМИ УПРАВЛЕННЯ

### Г.1 Програмне забезпечення об'єкта управління

#### Г.1.1 Список програм

[FBD] Main (\* Основная программа \*)  
[ST] Object (\* Объект управления \*)

#### Г.1.2 Основна програма Main



```

I_Object (Object)
Pump1Speed (* Скорость насосного агрегата 1 *)
Pump2Speed (* Скорость насосного агрегата 2 *)
I_Q (* Расход воды *)
SensorEL (* Датчик аварийного уровня *) (* Датчик аварийного уровня *)
SensorHL (* Датчик верхнего уровня *) (* Датчик верхнего уровня *)
SensorSL (* Датчик скорости *) (* Датчик скорости *)
SensorML (* Датчик среднего уровня *) (* Датчик среднего уровня *)
SensorLL (* Датчик нижнего уровня *) (* Датчик нижнего уровня *)

```

#### Г.1.3 Користувачький функціональний блок Object

```

IF I_State >= LREAL#10.0 THEN
    I_State := LREAL#10.0;
ELSIF I_State <= LREAL#0.0 THEN
    I_State := LREAL#0.0;
END_IF;

bo_SEL := I_State >= 9.0;
bo_SHL := I_State >= 8.0;
bo_SSL := I_State >= 5.1;
bo_SML := I_State >= 5.0;
bo_SLL := I_State >= 2.0;

I_State := (((bi_Q - I_Memory) / LREAL#1000.0) / LREAL#7.0685834705770345) * LREAL#0.1) + I_State;
IF I_State >= 10.0 THEN
    I_State := 10.0;
ELSIF I_State <= 0.0 THEN
    I_State := 0.0;
END_IF;

I_Memory := (LREAL#0.020689655172413793 * bi_n1) + (LREAL#0.020689655172413793 * bi_n2);
I_State (* Состояние интегратора *)
bo_SEL (* Датчик аварийного уровня *)
bo_SHL (* Датчик верхнего уровня *)
bo_SSL (* Датчик скорости *)
bo_SML (* Датчик среднего уровня *)
bo_SLL (* Датчик нижнего уровня *)
bi_Q (* Расход воды *)
I_Memory (* Значение в предыдущем цикле *)
bi_n1 (* Скорость насосного агрегата 1 *)
bi_n2 (* Скорость насосного агрегата 2 *)

```

#### Г.1.4 Список змінних

Global variables

```

Pump1Speed (* Скорость насосного агрегата 1 *)
REAL
Embedded: <Symbol>
Pump2Speed (* Скорость насосного агрегата 2 *)
REAL
Embedded: <Symbol>
SensorEL (* Датчик аварийного уровня *) (* Датчик аварийного уровня *)
BOOL
Embedded: <Symbol>
SensorHL (* Датчик верхнего уровня *) (* Датчик верхнего уровня *)
BOOL
Embedded: <Symbol>
SensorSL (* Датчик скорости *) (* Датчик скорости *)
BOOL
Embedded: <Symbol>
SensorML (* Датчик среднего уровня *) (* Датчик среднего уровня *)
BOOL
Embedded: <Symbol>
SensorLL (* Датчик нижнего уровня *) (* Датчик нижнего уровня *)
BOOL
Embedded: <Symbol>
Main
l_Object
Object
l_Q (* Расход воды *)
LREAL
:= LREAL#35.0
Object
bi_n1 (* Скорость насосного агрегата 1 *)
LREAL
bi_n2 (* Скорость насосного агрегата 2 *)
LREAL
bi_Q (* Расход воды *)
LREAL
bo_SEL (* Датчик аварийного уровня *)
BOOL
bo_SHL (* Датчик верхнего уровня *)
BOOL
bo_SSL (* Датчик скорости *)
BOOL
bo_SML (* Датчик среднего уровня *)
BOOL
bo_SLL (* Датчик нижнего уровня *)
BOOL
l_State (* Состояние интегратора *)
LREAL
:= LREAL#1.0
l_Memory (* Значение в предыдущем цикле *)
LREAL
:= LREAL#0.0

```

## Г.2 Программе забезпечення регулятора

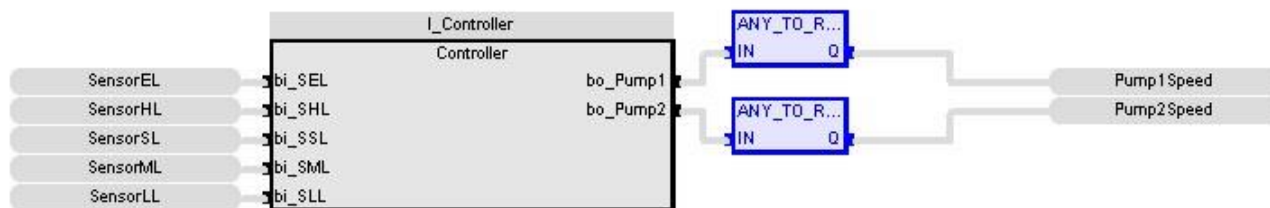
### Г.2.1 Список програм

```

[FBD] Main (* Основная программа *)
[ST] Controller (* Регулятор *)

```

### Г.2.2 Основна програма Main



```

l_Controller (Controller)
SensorEL (* Датчик аварийного уровня *)
SensorHL (* Датчик верхнего уровня *)
SensorSL (* Датчик скорости *)

```

```

SensorML (* Датчик среднего уровня *)
SensorLL (* Датчик нижнего уровня *)
Pump1Speed (* Скорость насосного агрегата 2 *)
Pump2Speed (* Скорость насосного агрегата 1 *)

```

## Г.2.3 Користувачький функціональний блок Controller

```

IF ANY_TO_DINT(1_SSL) = 0 THEN
  IF ANY_TO_DINT(1_SML) = 0 THEN
    initialize := 0.0;
    b_time := 0.0;
    hState := 0.0;
    hStateUpdate := 0.0;
  ELSIF initialize = 0.0 THEN
    initialize := 1.0;
  ELSE
    b_time := b_time + 1.0;
  END_IF;
ELSE
  IF initialize = 1.0 THEN
    hState := 5.1;
    hStateUpdate := LREAL#0.099999999999999645 / b_time;
    initialize := 0.0;
  ELSIF ANY_TO_DINT(1_SHL) = 1 THEN
    hState := 8.0;
    hStateUpdate := 0.0;
  ELSIF hState = 0.0 THEN
    hState := 6.5;
  ELSIF hState <> 0.0 THEN
    hState := hState + hStateUpdate;
    IF hState > 8.0 THEN
      hState := 8.0;
    END_IF;
  ELSE
    hState := 0.0;
  END_IF;
END_IF;

IF statePump_j = 0.0 THEN
  IF ANY_TO_DINT(1_SML) = 1 THEN
    statePump_j := 1.0;
  END_IF;
ELSE
  IF ANY_TO_DINT(1_SLL) = 0 THEN
    statePump_j := 0.0;
  END_IF;
END_IF;

IF statePump_j <> 0.0 THEN
  IF 1_Pump2 <> 0.0 THEN
    rtb_Sum3 := 1450.0;
  ELSE
    rtb_Sum3 := (LREAL#5.0 - hState) * LREAL#-52.0;

    IF rtb_Sum3 >= 100.0 THEN
      rtb_Sum3 := 100.0;
    ELSIF NOT (rtb_Sum3 > 0.0) THEN
      rtb_Sum3 := 0.0;
    END_IF;

    rtb_Sum3 := (LREAL#5.8 * rtb_Sum3) + LREAL#870.0;

    IF rtb_Sum3 >= 1450.0 THEN
      rtb_Sum3 := 1450.0;
    ELSIF NOT (rtb_Sum3 > 870.0) THEN
      rtb_Sum3 := 870.0;
    END_IF;
  END_IF;

  bo_Pump1 := rtb_Sum3;
ELSE
  bo_Pump1 := 0.0;
END_IF;

IF statePump = 0.0 THEN

```

```

        IF ANY_TO_DINT(l_SHL) = 1 THEN
            statePump := 1.0;
        END_IF;
    ELSE
        IF ANY_TO_DINT(l_SML) = 0 THEN
            statePump := 0.0;
        END_IF;
    END_IF;

    IF statePump <> 0.0 THEN
        rtb_Sum3 := 1450.0;
    ELSE
        rtb_Sum3 := 0.0;
    END_IF;

    bo_Pump2 := rtb_Sum3;

    l_SHL := bi_SHL;
    l_SSL := bi_SSL;
    l_SML := bi_SML;
    l_SLL := bi_SLL;

    l_Pump2 := rtb_Sum3;
    l_SSL
    l_SML
    initialize
    b_time
    hState
    hStateUpdate
    l_SHL
    statePump_j
    l_SLL
    l_Pump2
    rtb_Sum3
    bo_Pump1 (* Скорость насосного агрегата 1 *)
    statePump
    bo_Pump2 (* Скорость насосного агрегата 2 *)
    bi_SHL (* Датчик верхнего уровня *)
    bi_SSL (* Датчик скорости *)
    bi_SML (* Датчик среднего уровня *)
    bi_SLL (* Датчик нижнего уровня *)

```

## Г2.4 Список змінних

```

Global variables
SensorEL (* Датчик аварийного уровня *)
    BOOL
    Embedded: <Symbol>
    Profile = STRATON
SensorHL (* Датчик верхнего уровня *)
    BOOL
    Embedded: <Symbol>
    Profile = STRATON
SensorSL (* Датчик скорости *)
    BOOL
    Embedded: <Symbol>
    Profile = STRATON
SensorML (* Датчик среднего уровня *)
    BOOL
    Embedded: <Symbol>
    Profile = STRATON
SensorLL (* Датчик нижнего уровня *)
    BOOL
    Embedded: <Symbol>
    Profile = STRATON
Pump1Speed (* Скорость насосного агрегата 2 *)
    REAL
    Embedded: <Symbol>
    Profile = STRATON
Pump2Speed (* Скорость насосного агрегата 1 *)
    REAL
    Embedded: <Symbol>
    Profile = STRATON
Main
l_Controller
    Controller
Controller
bi_SEL (* Датчик аварийного уровня *)
    BOOL

```

```
bi_SHL (* Датчик верхнего уровня *)
  BOOL
bi_SSL (* Датчик скорости *)
  BOOL
bi_SML (* Датчик среднего уровня *)
  BOOL
bi_SLL (* Датчик нижнего уровня *)
  BOOL
bo_Pump1 (* Скорость насосного агрегата 1 *)
  LREAL
bo_Pump2 (* Скорость насосного агрегата 2 *)
  LREAL
l_SHL
  BOOL
  := FALSE
l_SSL
  BOOL
  := FALSE
l_SML
  BOOL
  := FALSE
l_Pump2
  LREAL
  := LREAL#0.0
l_SLL
  BOOL
  := FALSE
hStateUpdate
  LREAL
  := LREAL#0.0
hState
  LREAL
  := LREAL#0.0
initialize
  LREAL
  := LREAL#0.0
b_time
  LREAL
  := LREAL#0.0
statePump_j
  LREAL
  := LREAL#0.0
statePump
  LREAL
  := LREAL#0.0
rtb_Sum3
  LREAL
  := LREAL#0.0
```

## ДОДАТОК Д – ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ЛЮДИНО- МАШИННОГО ІНТЕРФЕЙСУ

### Д.1 Драйвера

State	Идентификатор	Description	File name
Filter text	Filter text	Filter text	Filter text
	Driver for internal variables		Intern
	Driver for mathematics variables		MATHDR32
	Driver for system variables		SYSDRV
	VIPA_200		S7TCP32
	zenon Logic: Controller		STRATONNG

### Д.2 Список змінних

State	Name	Идентификатор	Едини...	Net address	Data block	Offset	Bit num...	Alignm...	Symbol...	Driver	Data type	Decimals	Start offset
Filter text	Filter text	Filter text	Filter...	Filter text	Filter text	Filter text	Filter...	Filter...	Filter...	Filter text	Filter text	Filter...	Filter text
	Border			0	0	0	0	0		Intern - Driver for int...	BOOL	0	0
	Controller/Global/Pump15p...	Скорость насосног...		0	0	0	0	0		STRATONNG - zenon ...	REAL	0	0
	Controller/Global/Pump25p...	Скорость насосног...		0	0	0	0	0		STRATONNG - zenon ...	REAL	0	0
	Controller/Global/SensorEL	Датчик аварийног...		0	0	0	0	0		STRATONNG - zenon ...	BOOL	0	0
	Controller/Global/SensorHL	Датчик верхнего ...		0	0	0	0	0		STRATONNG - zenon ...	BOOL	0	0
	Controller/Global/SensorLL	Датчик нижнего у...		0	0	0	0	0		STRATONNG - zenon ...	BOOL	0	0
	Controller/Global/SensorML	Датчик среднего ...		0	0	0	0	0		STRATONNG - zenon ...	BOOL	0	0
	Controller/Global/SensorSL	Датчик скорости		0	0	0	0	0		STRATONNG - zenon ...	BOOL	0	0
	Level	Уровень стоков		0	0	0	0	0		MATHDR32 - Driver fo...	LREAL	0	0
	Pump1Speed	Скорость насосног...		0	0	0	0	0		S7TCP32 - VIPA_200	REAL	0	0
	Pump2Speed	Скорость насосног...		0	0	0	0	0		S7TCP32 - VIPA_200	REAL	0	0
	SensorEL	Датчик аварийног...		0	0	0	0	0		S7TCP32 - VIPA_200	BOOL	0	0
	SensorHL	Датчик верхнего ...		0	0	0	0	0		S7TCP32 - VIPA_200	BOOL	0	0
	SensorLL	Датчик нижнего у...		0	0	0	0	0		S7TCP32 - VIPA_200	BOOL	0	0
	SensorML	Датчик среднего ...		0	0	0	0	0		S7TCP32 - VIPA_200	BOOL	0	0
	SensorSL	Датчик скорости		0	0	0	0	0		S7TCP32 - VIPA_200	BOOL	0	0

### Д.3 Шаблони зображень

State	Display...	Name	Background c...	Freel...	L...	To...	Rig...	Bo...
Filter text	Filter text	Filter text	Filter text	Filter...	Filter...	Filter...	Filter...	Filter...
	<input checked="" type="checkbox"/>	ALARM STATUS LINE	■ #FF0000	<input checked="" type="checkbox"/>	0	0	2560	18
	<input checked="" type="checkbox"/>	MainScreenFrame	■ #AAAAAA	<input type="checkbox"/>	0	0	1366	700
	<input checked="" type="checkbox"/>	ButtonScreenFrame	■ #AAAAAA	<input type="checkbox"/>	0	700	1366	768



## Д.4 Функції

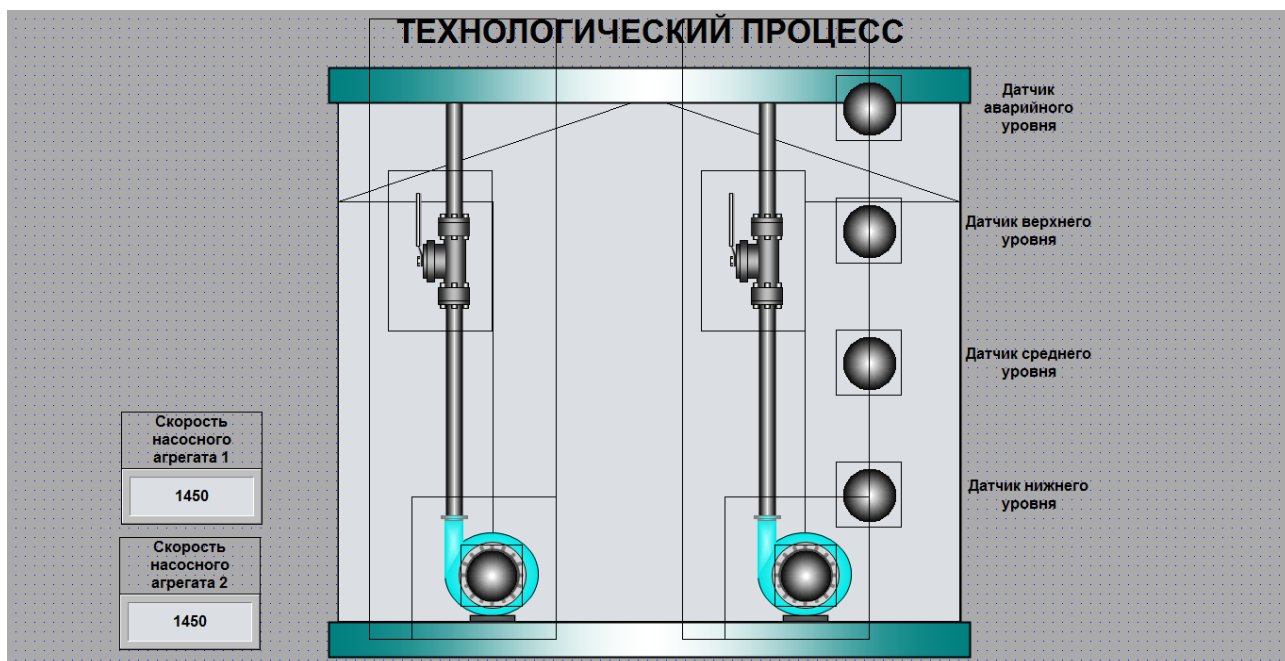
State	Name	Type	Parameter
Filter text	Filter text	Filter text	Filter text
	LanguageSwitch	Language switch	ZENONSTR.TXT - Список шрифтов 1
	SwitchMainScreen	Screen switch	MainScreen (Standard)
	SwitchTrendScreen	Screen switch	TrendScreen (Extended Trend) - SPEED[HD]
	SwitchCELScreen	Screen switch	CELScreen (Chronological Event List) - [*]...
	SwitchAMLScreen	Screen switch	AMLScreen (Alarm Message List) - [*][*]-[...]
	SwitchButtonScreen	Screen switch	ButtonScreen (Standard)
	ExitRuntime	Exit Runtime	
	ReloadProject	Reload project online	changed objects

## Д.5 Зображення

State	Name	Screen type	Frame	Backgro...	Start function
Filter text	Filter text	Filter text	Filter text	Filter t...	Filter text
	AMLScreen	Alarm Messa...	MainScreenFrame	#A#	< no function linked >
	ButtonScreen	Standard	ButtonScreenFrame	#A#	< no function linked >
	CELScreen	Chronological...	MainScreenFrame	#A#	< no function linked >
	MainScreen	Standard	MainScreenFrame	#A#	SwitchButtonScreen
	TrendScreen	Extended Trend	MainScreenFrame	#A#	< no function linked >

ТЕХНОЛОГИЯ	ТРЕНД	СПИСОК СОБЫТИЙ	СПИСОК ТРЕВОГ	ПЕРЕЗАГРУЗИТЬ	ВЫХОД
------------	-------	----------------	---------------	---------------	-------





**ВІДГУКИ КОНСУЛЬТАНТІВ КВАЛІФІКАЦІЙНОЇ РОБОТИ**



**ВІДГУК**

на кваліфікаційну роботу магістра на тему:

**«Обґрунтування параметрів комп'ютерної системи наземної перекачної станції ім. Сташкова»**

студента гр. 123М-19-1 **Саврицький Олександр Ігорович**

Кваліфікаційна робота представлена пояснювальною запискою об'ємом \_\_\_\_ стор. формату А4 і графічною частиною \_\_\_\_ листів формату А2.

Метою є обґрунтування параметрів комп'ютерної системи наземної перекачної станції ім. Сташкова.

Мета кваліфікаційної роботи: Удосконалення існуючої комп'ютерної мережі підприємства шляхом розробки математичної моделі мережі і проведення обчислювального експерименту.

Актуальність теми у повному обсязі викладена у вступі роботи. Проведений аналіз характеристик і особливостей комп'ютерної мережі підприємства. Проаналізовані методи і способи моделювання комп'ютерних мереж. Зроблені відповідні висновки що до необхідності аналізу роботи комп'ютерної мережі, її стійкості до впливу шкідливого програмного забезпечення. Відповідно до мети роботи обрані сучасні методи розробки математичної моделі. Для виконання завдань дослідження використовуються методи теорії масового обслуговування, розроблена математична модель комп'ютерної мережі як мережі масового обслуговування. Проведено дослідження властивостей мережі з різними параметрами інформаційного середовища та апаратних засобів. Науковий підхід до вирішення поставлених у роботі завдань дозволив зробити висновки що до доцільності використання математичних моделей комп'ютерної мережі для формування практичних рекомендацій поліпшення роботи мережі за умови незмінності її структури.

Розроблені моделі приймального резервуару і насосного агрегату, розроблена модель насосної станції. Сформульовані критерії якості

функціонування об'єкту управління, основним критерієм якості є мінімум витрат електроенергії. На базі безперервної моделі системи управління розроблена цифрова модель системи об'єкта управління. На базі програмних модулів розроблено ПЗ системи управління для програмного ПЛК zenon Logic що входить до складу SCADA системи zenon, та виконана перевірка функціонування програмного забезпечення. Програмне забезпечення функціонує відповідно до заданих алгоритмів і забезпечує виконання поставлених завдань.

Відповідно до вимог кваліфікаційна робота забезпечена графічними матеріалами.

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності фахівця спеціальності 123 «Комп'ютерна інженерія».

Виконані в кваліфікаційній роботі завдання відповідають вимогам до професійної діяльності фахівця освітньо-кваліфікаційного рівня магістра.

Оформлення пояснювальної записки та графічного матеріалу до неї виконано згідно з вимогами. Роботу виконано самостійно, відповідно до завдання та у повному обсязі.

У роботі відзначено такі недоліки:

1. Зустрічаються помилки та неточності у оформленні тексту пояснювальної записки;
2. Не показано результати, які б дозволили повністю розкрити можливості використання математичної моделі комп'ютерної мережі.

Оцінки по розділах - «\_\_\_\_\_».

Кваліфікаційна робота цілому заслуговує оцінку «\_\_\_\_\_95\_\_\_\_\_».

**Керівник роботи**

**проф. Коротенко Г.М.**

09.12.2020 р.

## РЕЦЕНЗІЯ

на кваліфікаційну роботу магістра по темі:

### **«Обґрунтування параметрів комп'ютерної системи наземної перекачної станції ім. Сташкова»**

студента гр. 123м-19-1 **Саврицький Олександр Ігорович**

Метою є обґрунтування параметрів комп'ютерної системи наземної перекачної станції ім. Сташкова.

Висновок про відповідність кваліфікаційної роботи завданню та освітньо-професійній програмі позитивний. У роботі використаний науковий підхід до вирішення поставленої задачі. Відповідно до задач, що вирішувалися, обґрунтовано обрані методи вирішення

Кваліфікаційна робота виконана відповідно до вимог, у повному обсязі. Зміст розділів і підрозділів роботи показують, що здобувач чітко розуміє значення окремих етапів роботи, доцільність глибини розгляду окремих вирішених завдань і необхідності практичного використання результатів. Науковий підхід, використаний у роботі, показує що здобувач володіє сучасними методами математичного моделювання мереж і розуміє доцільність їх використання з практичної точки зору.

Позитивні сторони атестаційної роботи:

1. використано результати аналізу самих сучасних підходів до моделювання комп'ютерних мереж;
2. визначені основні характеристики комп'ютерної мережі за її математичною моделлю;
3. розроблені моделі приймального резервуару і насосного агрегату, розроблена модель насосної станції, сформульовані критерії якості функціонування об'єкту управління, основним критерієм якості є мінімум витрат електроенергії;

4. на базі безперервної моделі системи управління розроблена цифрова модель об'єкта системи управління;
5. на базі програмних модулів розроблено ПЗ системи управління для програмного ПЛК zenon Logic що входить до складу SCADA системи zenon, та виконана перевірка функціонування програмного забезпечення. Програмне забезпечення функціонує відповідно до заданих алгоритмів і забезпечує виконання поставлених завдань.

Основні недоліки атестаційної роботи:

1. не використані усі можливості розробленої математичної моделі комп'ютерної мережі;
2. значно завищену об'єм пояснювальної записки.

Кваліфікаційна робота в цілому заслуговує оцінку «\_\_\_\_\_».

**Рецензент** \_\_\_\_\_

\_\_\_\_.12.2020 г.